Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 856761, 17 pages
doi:10.1155/2010/856761

Research Article

A Video Browsing Tool for Content Management in Postproduction

Werner Bailer, Wolfgang Weiss, Gert Kienast, Georg Thallinger, and Werner Haas

JOANNEUM RESEARCH Forschungsgesellschaft mbH, Institute of Information Systems, Steyrergasse 17, 8010 Graz, Austria

Correspondence should be addressed to Werner Bailer, werner.bailer@joanneum.at

Received 31 August 2009; Revised 24 November 2009; Accepted 17 December 2009

Academic Editor: Jungong Han

We propose an interactive video browsing tool for supporting content management and selection in postproduction. The approach is based on a process model for multimedia content abstraction. A software framework based on this process model and desktop and Web-based client applications are presented. For evaluation, we apply two TRECVID style fact finding approaches (retrieval and question answering tasks) and a user survey to the evaluation of the video browsing tool. We analyze the correlation between the results of the different methods, whether different aspects can be evaluated independently with the survey, and if a learning effect can be measured with the different methods, and we also compare the full-featured desktop and the limited Web-based user interface. The results show that the retrieval task correlates better with the user experience according to the survey. The survey rather measures the general user experience while different aspects of the usability cannot be analyzed independently.

1. Introduction

With the increasing amount of multimedia data being produced, there is growing demand for more efficient ways of supporting exploration and navigation of multimedia data. Viewing complete multimedia items in order to locate relevant segments is prohibitive even for relatively small content sets due the required user time for viewing and the amount of data that needs to be transferred. Classical search and retrieval approaches require sufficient metadata to index the content and the feasibility to formulate a query in terms of the available metadata. Manual annotation of the content yields semantically meaningful metadata that can be effectively searched by human users, but at high annotation costs. Automatic metadata extraction approaches are in many cases not able to fully capture the semantics of the content, which makes the metadata difficult to query.

Multimedia content abstraction methods are complementary to search and retrieval approaches, as they allow for exploration of an unknown content set, without the requirement to specify a query in advance. This is relevant in cases where only few metadata are available for the content set, and where the user does not know what to expect in the content set, so that she is not able to formulate a query. In order to enable the user to deal with large content sets, it has to be presented in a form which facilitates its comprehension and allows to quickly judge the relevance of segments of the content set. Media content abstraction methods will (i) support the user in quickly gaining an overview of a known or unknown content set, (ii) organize content by similarity in terms of any feature or group of features, and (iii) select representative content for subsets of the content set that can be used for visualization.

The term *video abstract* is defined in [1] as "a sequence of still or moving images presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content while the essential message of the original is preserved". The authors of [2] use the term *video abstraction* to denote all approaches for the extraction and presentation of representative frames and for the generation of video skims. In this paper we use the term *content abstraction* to refer to all approaches that aim at providing condensed representations of segments of a single media item or a collection of items, that are relevant or salient, independent of the purpose, context, form, creation method and presentation style of the abstract. Despite their differences in all those aspects, the existing approaches for creating multimedia content abstractions share a number of

similar steps which allow to define a common process model. Based on the definition of a process model for multimedia content abstraction, we aim at defining a software framework supporting video browsing.

The application scenario is that of content management in the post-production phase of film and TV production. In post-production environments, users typically deal with large amounts of audiovisual material, such as newly shot scenes, archive material and computer generated sequences. A large portion of the material is unedited and often very redundant, for example, containing several takes of the same scene shot by a number of different cameras. Typically only few metadata annotations are available (e.g., which production, which camera, which date). The goal is to support the user in navigating and organizing these audiovisual collections, so that unusable material can be discarded, yielding a reduced set of material from one scene or location available for selection in the post-production steps.

Media production workflows are becoming increasingly flexible and distributed, involving many contributors located at different sites. This poses new challenges for managing audiovisual assets, as efficient access to content needs to be possible remotely and because people who could be consulted when looking for content are not in reach. Based on a previous desktop application for browsing audiovisual content [3] we propose a Web application that provides the same functionality, but allows to access content repositories remotely.

The contributions of this work are the following. We present a process model for multimedia content abstraction and implement a framework for video browsing based on this model. The framework can be easily extended to support browsing by any low-, mid-, or high-level feature. We present a desktop and a Web-based user interface for the framework. We evaluate both user interfaces using different evaluation approaches and compare the results between the two user interfaces and the different evaluation methods.

The rest of this paper is organized as follows. Section 2 discusses aspects of multimedia content abstraction and presents a general process model. In Section 3 we present an implementation of this process for interactive video browsing as well as a desktop and Web-based user interface in Section 5. The evaluation of the desktop and Web-based video browsing tool and its results are discussed in Section 6, and Section 7 concludes the paper.

2. Multimedia Content Abstraction

2.1. Aspects of the Problem. Multimedia content abstraction encompasses different ways of summarizing, condensing and skimming multimedia content. There are a number of aspects that discriminate these approaches proposed in literature. In the following we discuss some of them, focusing on those that influence the design of our software framework for video browsing. We do not attempt to provide a complete survey of related work here, for a comprehensive overview and comparison of video abstraction methods see for example [2].

Content abstraction can be done manually, automatically or semi-automatically (e.g., using user input to define examples of relevant content segments [4]). A basic aspect when creating the abstract is its **purpose**, which can be to objectively summarize the content conveying all of the original message or to deliberately bias the viewer (e.g., when creating a movie trailer, cf. [5]). In our case the purpose is to maximize the amount of information contained in the abstract.

Somewhat related to the purpose is the **context** of the abstract, which may be undefined and independent of the initial input of the user (e.g., when a user starts browsing), it can be defined by user input, or it can be predefined, for example, when abstracts are used for representing search results and the user's query is known [6]. Domain knowledge also contributes to the definition of the context, as it helps defining the relevance of content segments. Most video abstraction approaches for sports broadcasts exploit this knowledge (e.g., goal scenes in soccer games are relevant). The context in film and TV postproduction is given by the current production a user is working on. However, while this context is possibly defined in scripts and storyboards, it is not formalized in a way that is directly usable by content management tools.

A number of aspects are related to the media type of the content to be extracted. The **dimension** of the content may be a single media item (e.g., one video) or a collection of items (an example for the visualization of a content set is presented in [7]). In the latter case all items may be of the same or of different types (e.g., a mixed collection of still images and videos). The media type also determines whether the content set has a defined **order**. For example, a video or audio stream has an intrinsic temporal order, which is often kept in the abstract. In our case the dimension is given by the set of content related to a certain production, which is often 30 times or more of the duration of the final content.

One of the most important aspects is the **content structure**. In [8] the authors discriminate *scripted* (such as movies) from *unscripted* content (such as sports video, surveillance video, home videos). Of course, the boundaries between the two are very fuzzy. Another dimension of structure is *edited* versus *unedited* content. While some content is not intended to be edited (e.g., surveillance video), there exist rushes for both scripted and unscripted content. For edited scripted content the abstraction algorithm can attempt to detect and use the structure of the content (such as dialogs [5]), while for unscripted (and especially also unedited) content other approaches are required (e.g., [9]). Content structure does not only exist on the level of the single media item, but also on the level of the collection in the case of multiitem abstracts. In some cases the collection has a "macrostructure", such as a set of rushes produced according to a script. The content encountered in our use case is typically unedited, but depending on the production it can be structured or unstructured.

There is a big variety of approaches for the **presentation** of abstracts. It can be interactive or non-interactive,

sequential or hierarchical, and different media types and visualizations can be used. Typical ways of presentation are static visualizations of representative frames (using different visualizations such as story boards or comic book style [10]), hierarchical static or navigatable visualizations of representative frames (e.g., [11]) and video skims [12]. One aspect related to presentation is whether the abstraction system is distributed. Web-based browsing of video content has already been considered in early work on video retrieval (e.g., [13]). However, most of this work deals with browsing abstract of single video items as collections of video content were rarely accessible on the Web. In [14] authors propose search and browsing interfaces for the Open Video archive. Due to the fact that flexibility and interactivity of Web applications has been quite limited, most of these approaches are limited to static or animated key frames. Only recently it has become possible to provide the functionality available in many desktop applications for video browsing also on the Web.

Unified frameworks for multimedia content abstraction have been proposed, mostly to integrate content abstraction and retrieval (e.g., [8, 15]), but they are often limited to some types of media (e.g., only video), to only scripted or only unscripted content, or they only support certain presentation forms (such as skims [16]).

2.2. Process Model. In [3] we have proposed a multimedia content abstraction process that supports the creation of content abstractions independent of many of the aspects discussed in Section 2.1, that is, media type, context, presentation (interactive and noninteractive) and visualization, extending the generic five step process for video skimming and four step process in clustering-based approaches described in [2]. In the following, we briefly review the definition of the process and relate it to our video browsing use case.

Design. The first stage deals with the conceptualization of the content abstraction and makes basic decisions about its purpose and form. If the work is done manually, this involves a creative intervention by the user. If it is done automatically, many of these decisions may have been already taken by the developer of the application, and they are hard-wired or depend on the application's state and context. In the case of interactive video browsing for post-production, many design decisions of the abstract are taken when developing the application. The user just has control over some presentation aspects.

Clustering. In this stage, similarities within the content set are found and content segments that are related in terms of some feature are grouped. If selection has been performed before, the selected subset of content segments is used as an input, otherwise clustering is performed on the whole content set. Clustering is a key step in content abstraction, as it is crucial for the reduction of redundancy. The clustering step in the interactive video browsing tool is specific to the feature the user has selected for clustering, and each clustering step uses one feature. Content clustering and selection steps are repeated iteratively in this case.

Selection. This step selects relevant segments or groups of segments according to a defined set of criteria. If these criteria have been specified by the user or are known from the application context, the selection step can be performed before clustering (e.g., sports highlights extraction). In other cases, the selection step is performed after clustering, selecting relevant clusters instead of segments. In many automatic content abstraction processes the selection criteria are a result of clustering, for example, outliers such as unusual events in the content are found relevant to be included in the abstract (e.g., when summarizing surveillance video). The selection of content for interactive video browsing has two aspects: The decisions which kind of content to keep and which to discard are made by the user's interactions. The selection of representative content to visualize subsets of the content depend on automatic content analysis (e.g., key frame extraction) and feature specific algorithms.

Presentation. In order to visualize and/or auralize the selected groups of content segments, either new media items are created (e.g., mosaics of a shot [17], plots of a data space, time lines) or representative segments are used (e.g., representative frames for a set of video segments, a short clip). The media items representing groups of content segments are organized according to the layout of the presentation, forming a new multimedia document. The presentation in our interactive video browsing tool is a light table representation of selected key frames. The key frames can be kept in their original order or ordered by a feature value.

Consumption. If the result of the content abstraction is noninteractive (e.g., video skim, movie trailer), the consumption step only consists of viewing the document (and possibly navigation using the player controls). In the interactive case, such as in video browsing, the user selects a subset of the content segments and maybe also changes further parameters, thus altering the input for selection and clustering. The result of re-running the creation process is an updated presentation that better suits the user's needs and interests.

The basic workflow in the browsing tool, shown in Figure 1, is as follows: the user starts from the complete content set. By selecting one of the available features the content will be clustered according to this feature. Depending on the current size of the content set, a fraction of the segments (mostly a few percent or even less) is selected to represent a cluster. The user can then decide to select a subset of clusters that seems to be relevant and discard the others, or repeat clustering on the current content set using another feature. In the first case, the reduced content set is the input to the clustering step in the next iteration. The user can select relevant items at any time and drag them into the result list.

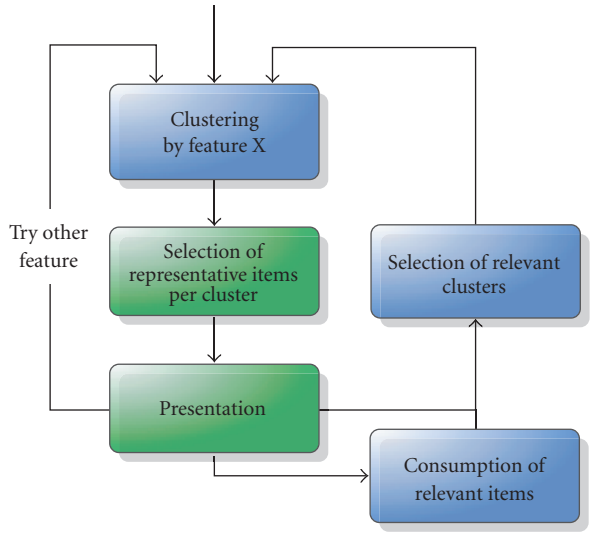

Figure 1: The basic workflow of the video browsing tool.

3. Implementing the Abstraction Process for Interactive Video Browsing

From the user's point of view, the basic difference between content browsing and search and retrieval is the limited need to know how to formulate a query and about what to expect from the content set. Thus, the content browsing tool must support the user in building a query step by step, by trying to add new restrictions and reducing the content set when applying the chain of restrictions built up so far (cf. the ostensive model of developing information needs [18]).

3.1. Process. The three core steps of the abstraction process, that is, **selection**, **clustering** and **presentation**, can be mapped to components in the software framework in a straight forward fashion. This is more difficult for the first (**design**) and last (**consumption**) step, as they are much more dependent on the specific application. In addition to the conceptual phases of the process described above there is the technical need for a **preprocessing** phase in the software implementation of the abstraction process. This step ingests material into the system, performs the required content analysis and annotation and prepares the data structures (e.g., indices) that are required by the following selection and clustering operations. The preprocessing phase is directly influenced by the design step of the process, as the decisions taken there determine the features and annotations needed and thus the content analysis operations that have to be performed. In the case of interactive video browsing, the **selection** and **clustering** steps are executed iteratively. In the **presentation** step the selection of representative media items or the creation of a visualization of a segment depends on the feature and is also tied to the clustering and selection approaches. The rest of the presentation step, as well as the consumption step, are implemented in the user interface components.

3.2. Components. This section describes the manifestation of the functionalities of the framework in software components. The framework defines interfaces for all these components. The feature specific components are implemented as plug-ins, allowing to easily add new features or change the implementation for a certain feature. Figure 2 shows the components of the framework.

The metadata repository is a transversal component for storing metadata and links to data throughout all steps of the process. The indexing service ingests content descriptions and builds additional efficient index structures. The summarizer, together with its plugins, implements the **selection** and **clustering** steps of the process for the different features.

The content analysis tools performing the actual feature extraction and producing the MPEG-7 descriptions that are imported by the indexing service as well as the features which are extracted are described in Section 4. The user interface components implementing the **presentation** step of the process are described in detail in Section 5.

3.2.1. Metadata Repository. The metadata repository is a basic infrastructure component for managing the media items under control of the video browsing application. Essence and derived essence created by automatic content analysis (such as for example representative frames) are stored in the file system. The complete metadata descriptions are stored as MPEG-7 documents in the file system as well. In addition, more efficiently searchable index structures are kept for those metadata items that are needed for clustering and selection. They are kept in a relational database. We currently use SQLite (http://www.sqlite.org), but if necessary, a more powerful database system could be integrated instead.

3.2.2. Indexing Service. The indexing service is responsible for ingest of content into the abstraction system. It does not perform content analysis itself, as legacy metadata or manual annotations might be available. The input to the indexing service are MPEG-7 descriptions conforming to the Detailed Audiovisual Profile [19]. The service watches a directory for new MPEG-7 descriptions or is triggered by a web service call. The indexing service processes the metadata descriptions and fills the index data structures. The core implementation of the service just performs feature-independent tasks such as registering new content, while plug-ins are invoked for all other tasks.

Feature specific indexing is performed by a set of indexing plug-ins. The plug-ins extract the features related information from the MPEG-7 documents and create the necessary database and/or index structure entries. A plug-in will also create additional tables in the database or index structures if they are not yet there (i.e., if the plug-in for the feature has not been used before). This increases the flexibility of the framework, as new indexing plug-ins can be easily registered with the indexing service and will be used for all further incoming documents. In order to speed up clustering in the summarizer, the indexer plug-ins for some features also create and update additional information

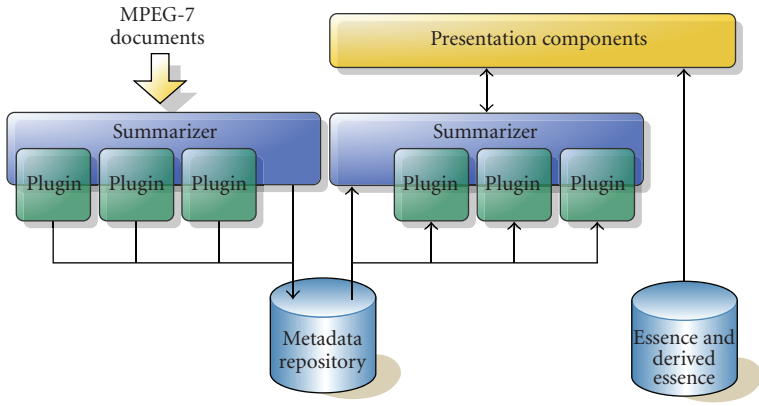

Figure 2: The components of the framework for video browsing.

such as a table of mutual similarities of descriptors in the documents indexed so far.

3.2.3. Summarizer. The summarizer is the component handling clustering, filtering and selection of representative media items. It accesses the data structures created and filled by the indexing service and has a generic interface towards the presentation layer in order to allow the use of different visualization and interaction paradigms. The functionality of the core implementation of the summarizer is mainly that of a broker, as—like in the indexing service—all feature-specific tasks are delegated to a set of plug-ins. The summarizer has a state that is defined by the current data set and its cluster structure. It also keeps a history of the clustering and selection operations carried out so far, as well as their parameters. This allows implementing undo and redo functionality in interactive applications, as well as storing the users' browsing trails in order to improve the clustering and selection algorithms.

Each plug-in provides the following functionality for one feature: the clustering algorithm and optionally algorithms for selecting a subset of the current data and for selecting/creating representative media items for a data set. The framework defines interfaces for all three types of algorithms. For the two latter ones, simple feature-independent default implementations are provided by the framework, but a plug-in can override them. It is possible to provide multiple plug-ins for one feature, for example, to experiment with different clustering algorithms.

3.2.4. Presentation Components. There are two implementations of the presentation components: a desktop application and a Web application. In the desktop application, the user interface (described in Section 5) is directly linked to the summarizer. Figure 3 illustrates the architecture of the Web application. The summarizer libraries offer their functionality to the Web-based version as Web services using gSoap (http://gsoap2.sourceforge.net). The Web-based video browsing tool is a Java Web application built with the Google Web Toolkit (http://code.google.com/webtoolkit/) and deployed in the servlet container Apache Tomcat

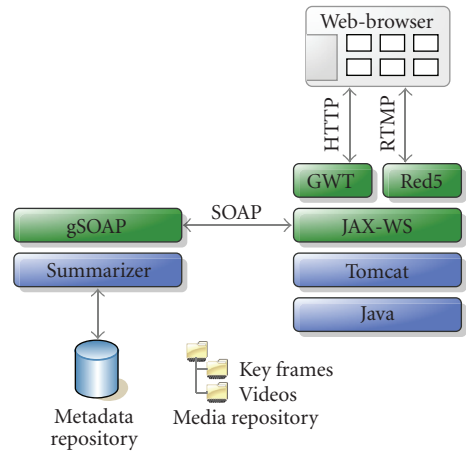

Figure 3: Architecture of the Web-based video browsing tool.

(http://tomcat.apache.org). To ensure high scalability we use default capabilities of the servlet container where each request is handled in a separate thread which can run on a own core of a processor. The Web service client is implemented with the Java API for XML Web Services (JAX-WS) (https://jax-ws.dev.java.net). Over the Web services cluster information and datasets are retrieved. The key frame images and videos are loaded directly from the media repository. The videos are streamed on demand with the Red5 (http://red5.org) Flash server to the client.

4. Feature Extraction

Feature extraction is performed by a content analysis framework using a dataflow graph approach [20]. The feature extraction is performed before ingest by the indexing service and produces one metadata description per video conforming to the MPEG-7 Detailed Audiovisual Profile [19].

First shot boundary detection and representative frame extraction are performed. The results of this step are used

as prerequisites for the extraction of other features discussed below and for visualization. As shot boundaries are natural limits of the occurrence of most visual features (such as camera movements, object occurrences), they are an important prerequisite for further visual feature extraction algorithms. For each shot, a number of representative frames is selected. The selection of representative frame positions is based on the visual activity in the material, that is, the more object and/or camera motion, the shorter the time interval between two representative frame positions.

Based on the shot structure the features described below are extracted. This are the features that are available for clustering in the browsing tool. Each of the subsections also describes the clustering algorithms that are used for the feature.

The content analysis tools support distributing content analysis tasks across different cores or machines in order to increase the throughput of the system. For the features listed below, the total processing takes about six times longer than realtime.

4.1. Camera Motion. The use of camera motion as a browsing feature is twofold: it is often used to guide the user's attention and express relevance of certain parts of the scene, for example, zooming on an object or person is an indicator of relevance, and in field sports, pans indicate the direction of the game. Secondly, it is an important selection criterion during editing, as visual grammar imposes constraints on the camera motion of sequences to be combined. The extraction algorithm (described in detail in [21]) is based on feature tracking, which is a compromise between spatially detailed motion description and runtime performance. The feature trajectories are then clustered by similarity in terms of a motion model and the cluster representing the global motion is selected. Camera motion is described on a sub-shot level. A new camera motion segment is created, when there is a significant change of the camera motion pattern (e.g., a pan stops, a zoom starts in addition to a tilt). For each of these segments, the types of motion present and a roughly quantized amount of motion are described.

We have implemented two clustering algorithms for camera motion. The first creates a fixed number of clusters, one for each type of camera motion (pan left, pan right, tilt up, tilt down, zoom in, zoom out, static). A camera motion segment is assigned to a cluster, if that type of camera motion is present in the segment, for example, if a segment contains a pan left and a zoom in, it is assigned to both clusters. The second clustering method tries to better model the actual data. Using the amounts of each type of motion, each camera motion segment is described by a vector in a three-dimensional feature space. The feature vectors are then clustered using the Mean Shift algorithm [22]. The algorithm determines the number of clusters and assigns each camera motion segment to one of them. Depending on the data, the clusters contain single camera motions or combinations and a textual label for the cluster is created (e.g., "moderate pan and strong zoom").

4.2. Visual Activity. Visual activity is a measure of the dynamics in a scene. Together with camera motion information, it is a measure for the local motion in a scene and can thus be used to discriminate quiet scenes from those with object motion. In this application we just measure the amplitude of visual change. The list of amplitude values is then median filtered to be robust against short term distortions and split into homogeneous segments. Each of these sub-shot segments is described by its average activity value. Clustering is performed using the k-means algorithm.

4.3. Audio Volume. Audio volume can for example be used to discriminate shots without any sound, calm shots of inanimate objects, interviews with a constant volume level and loud outdoor shots in city streets. As no content-based audio segmentation is available in the system, we use segments of a fixed length of 30 seconds. A list of audio volume samples is extracted for each of these segments by calculating the average volume of a 0.5 seconds time window. The list is then median filtered to be robust against short term distortions and split into homogeneous segments. Each of these sub-segments is described by its average volume value. Clustering is performed using the k-means algorithm.

4.4. Face Occurrence. The occurrence of faces is a salient feature in video content, as it allows inferring the presence of humans in the scene. The size of a face is also a hint for the role of the person, that is, a large face indicates that this person is in the center of attention. Our extractor is based on the face detection algorithm from OpenCV (http://opencvlibrary.sourceforge.net). In order to make the description more reliable and to eliminate false positives, which mostly occur for a single or a few frames, we only accept face occurrences that are stable over a longer time (we use a time window of about a second to check this). As a result we get a continuous segmentation into sub-shot segments with and without faces. There is no need for a specific clustering algorithm, as there are only two groups of segments (face and nonface).

4.5. Global Color Similarity. Global color similarity allows to group shots that depict visually similar content, for example, several takes of the same scene or different shots taken at the same location (if the foreground objects are not too dominant). To describe the color properties of a shot, the MPEG-7 ColorLayout descriptor [23] is extracted from each representative frame. The ColorLayout descriptor has the advantage of also taking the spatial color distribution of the image into account. In order to reduce the number of color descriptors to be processed, similar descriptors extracted from representative frames of the same shot are eliminated. Then the pair-wise similarities between all remaining descriptors of the content set are calculated and stored in a matrix. The similarity matrix is used as input for hierarchical clustering using the single linkage algorithm [24]. The cutoff value for the resulting tree is determined from the desired number of clusters.

4.6. Repeated Takes. In film and video production usually large amounts of raw material are shot and only a small fraction of this material is used in the final edited content. The reason for shooting that amount of material is that the same scene is often taken from different camera positions and several alternative takes for each of them are recorded, partly because of mistakes of the actors or technical failures, partly to experiment with different artistic options. The action performed in each of these takes is similar, but not identical, for example, has omissions and insertions, or object and actor positions and trajectories are slightly different. Identifying the takes belonging to the same scene and grouping them can thus significantly increase the efficiency of the work.

We use the approach proposed in [25] to identify repeated takes of the same scene. This algorithm uses a variant of the Longest Common Subsequence (LCSS) measure on a sequence of visual activity samples and color and texture features of regularly samples key frames to identify takes of the same scene. The detection results are described in the MPEG-7 document.

4.7. Multiple Views. Recently the production of multi-view video content is of growing importance, mainly driven by stereoscopic cinema. 3D television is also an emerging application area. For multi-view content the relation of clips between views is stored in the metadata description. In addition, key frames need to be extracted synchronously from all views. If necessary, the clips from different views can be automatically temporally aligned, using the method for repeated take detection [25] with a different parameterization.

The component that implements most of the support for multiview content is the indexing service. In addition plug-ins for handling multiview specific metadata have been added to the indexing service and the summarizer. The indexing service adds information about the relation of the streams to the database. Stream-specific metadata can be supported by the same indexing service plug-ins that handle single view content, while a new plug-in has been developed handling cross-stream metadata.

5. User Interfaces

The user interfaces of the desktop (Figure 4) and Web-based (Figure 5) version of the video browsing tool have been designed to be as similar as possible. As illustrated in these figures, the central component of the browsing tool's user interface is a light table (5). The light table shows the current content set and cluster structure using a number of representative frames for each of the clusters. The clusters are visualized by colored areas around the images. By clicking on an image in the light table view, a video player (a Flash video player in the case of the Web-based version) is opened and plays the segment of the video that is represented by that image. The workflow in the browsing tool is as follows: the user starts with selecting a dataset (1). By selecting one of the available features (2) the content will be clustered according to this feature (e.g., camera motion, visual activity, faces, or global color similarity). Depending on the current size of the content set, a fraction of the segments (mostly a few percent or even less) is selected to represent a cluster. The user can then decide to select a subset of clusters that seems to be relevant and discard the others (3), or repeat clustering on the current content set using another feature (2). In the first case, the reduced content set is the input to the clustering step in the next iteration. The cluster selection and the size adjustment of key frames are visualized differently in the desktop and Web-based versions. The clustered segments can be ordered by original temporal order or by the feature value.

On the left side of the application window the history (6) and the result list (7) are displayed. The history window automatically records all clustering and selection actions done by the user. By clicking on one of the entries in the history, the user can set the state of the summarizer (i.e., the content set) back to this point. The user can then choose to discard the subsequent steps and use other cluster/selection operations, or—in the desktop version—to branch the browsing path and explore the content using alternative cluster features. The result list (7) can be used to memorize video segments and to extract segments of videos for further video editing, for example, as edit decision list (EDL). The user can drag relevant key frames into the result list at any time, thus adding the corresponding segment of the content to it. The size of the images in the light table view can be changed dynamically (4) so that the user can choose between the level of detail and the number of visible images without scrolling.

In the desktop application, the temporal context of a key frame is shown by a time line of temporally adjacent key frames that is shown when the user moves the mouse over a frame. Figure 6 shows such as example in a view that is clustered by repeated takes of a scene.

In case of multiview content, the browsing tool allows clustering content by the view it originates from or by a scene shot from multiple views. The same features are also available for contextual similarity search on any of the results shown in the tool. Figure 7 shows an example of clustering multi-view content by the camera from which it has been shot, using synchronously extracted key frames from the views.

6. Evaluation

The evaluation of video browsing tools is still an open issue, with different evaluation approaches proposed in literature. In [26], we have reviewed approaches in the literature and compared different evaluation methods. Here we apply different of these methods to the desktop and Web-based version of the video browsing tool and compare their results for our video browsing tool.

6.1. Research Questions. Given the fact that there is no established method for the evaluation of multimedia browsing we have chosen to apply both TRECVID [27] style approaches as well as a survey taking the user experience into account and intend to compare their results. The *retrieval tasks* contain

Figure 4: Screenshot of the video browsing tool desktop application.

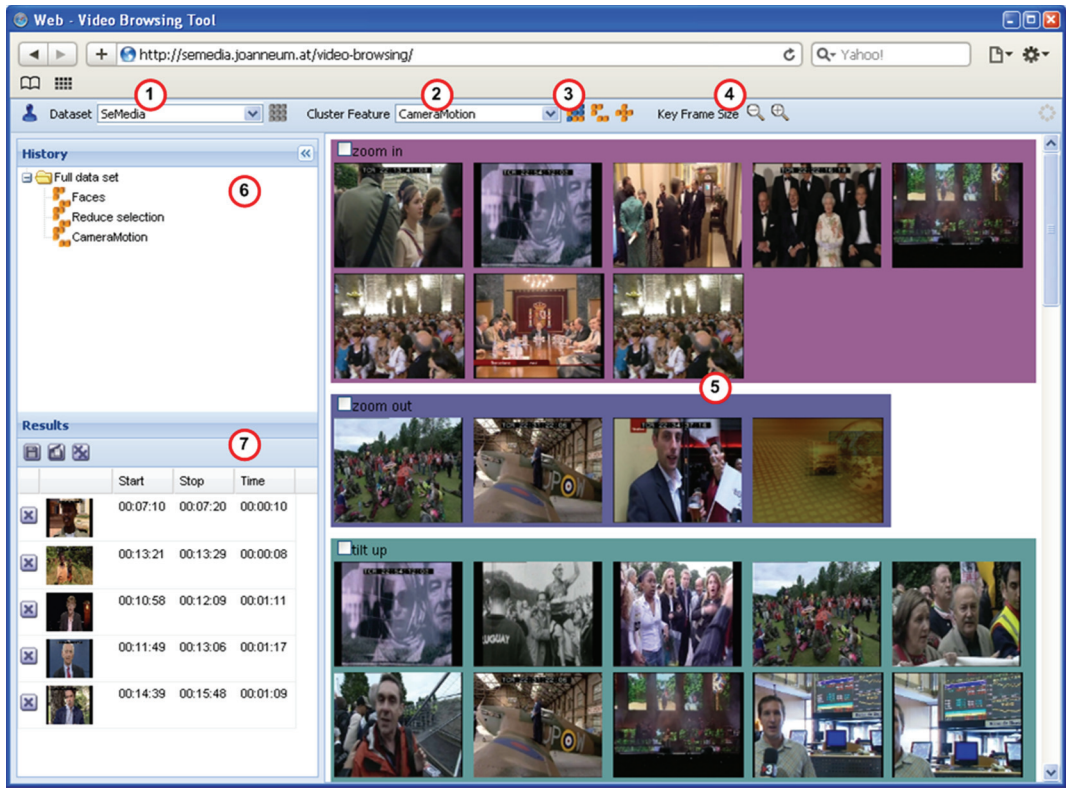

Figure 5: Screenshot of the Web-based video browsing tool.

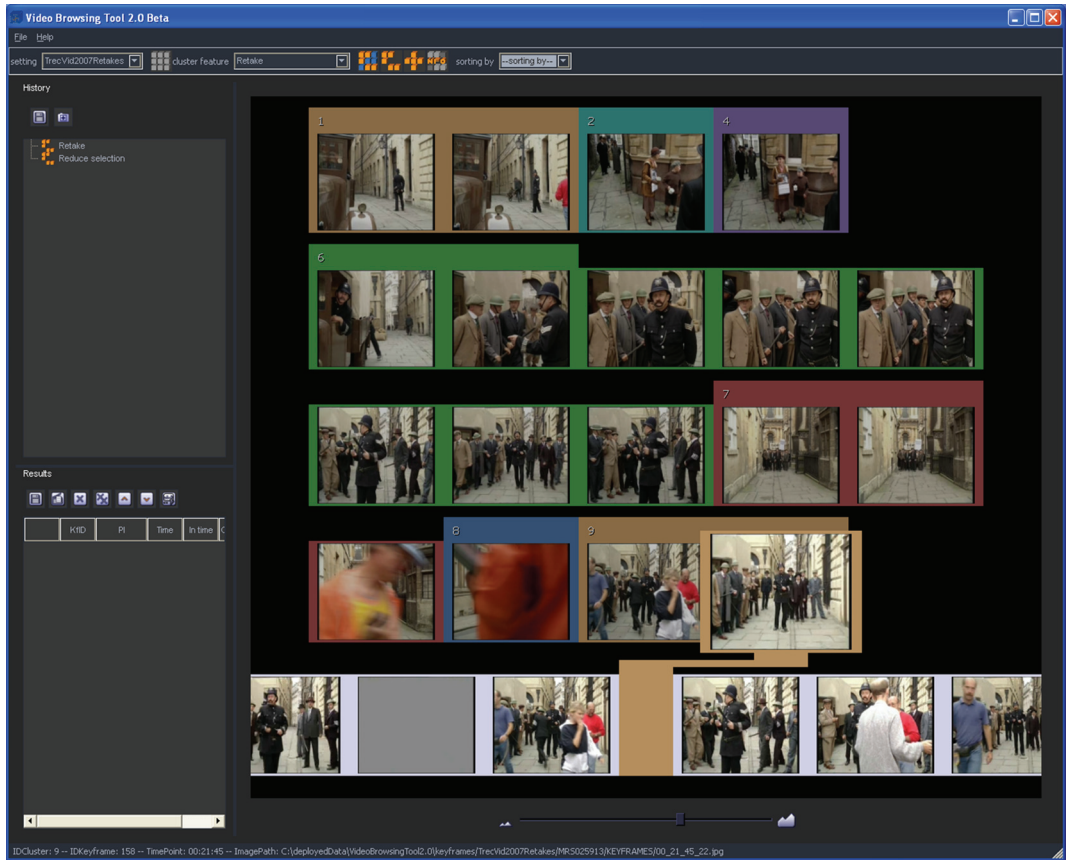

FIGURE 6: Screenshot of the video browsing tool: repeated takes. The images in the bottom row show the temporal context of the key frame under the mouse cursor.

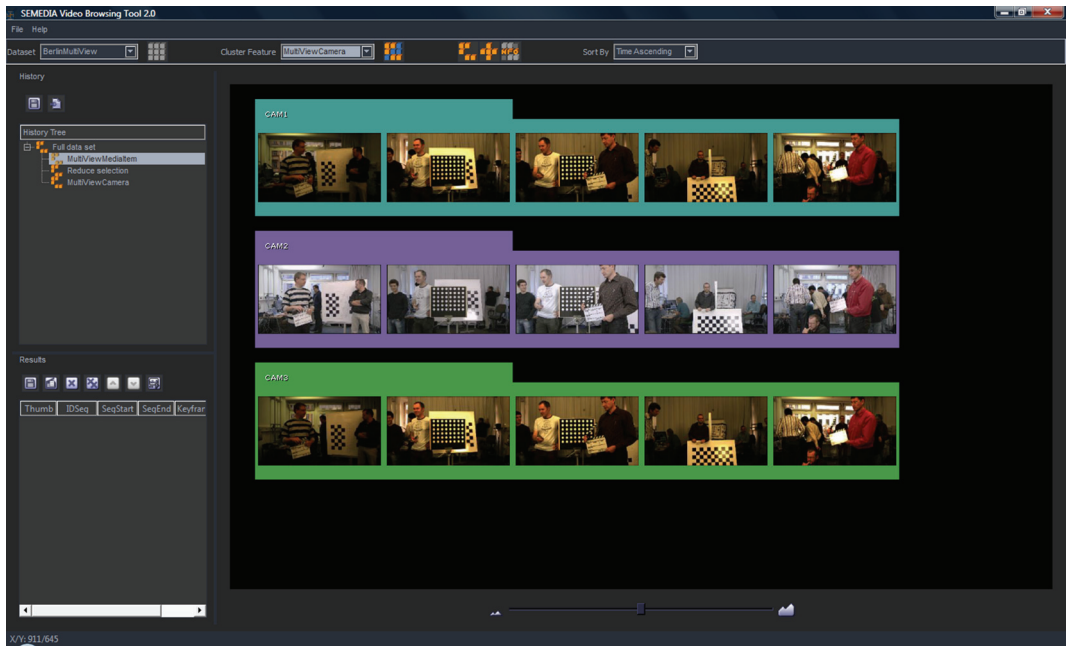

FIGURE 7: Screenshot of the video browsing tool: multi-view content.

a well defined need for video clips (motivated by scenarios in film and TV post-production), the *question answering tasks* are more goal oriented, leaving the description of the needed video clips more fuzzy. We designed corresponding pairs of retrieval and question answering tasks that target the same content sets. The *survey* asks about the experiences of the users when completing the two types of tasks. The retrieval and question answering tasks as well as the questionnaires are available at http://semedia.joanneum.at/.

In particular, we want to answer the following questions.

(i) Do different user groups achieve different results?

(ii) Is there a correlation between the results of the retrieval and question answering tasks and the assessment of the users in the post-task questions of the survey? We want to know whether the results from the different approaches yield similar or complementary results.

(iii) Can the post-task questions of the survey be answered independently? Surveys aim at giving a more holistic picture of the user experience. We are thus interested whether the questions asking about different aspects of the tool can be treated independently.

(iv) Is there a learning effect? Is it evident in the task results that users achieve better results when using the tool for some time, and does that correspond to the user's experience as expressed in the survey?

(v) Do users achieve a higher precision score when viewing the video before selecting a segment?

(vi) Is there a difference in the results achieved when using the desktop and Web-based version of the browsing tool?

The experiments to answer the first five questions are conducted with the desktop version of the browsing tool and the last question was evaluated with the Web version of the browsing tool.

6.2. Materials and Procedure. The survey is independent of the data set used. For the retrieval and question answering tasks two data sets are used. The TRECVID BBC Rushes 2006 data set is the one used in the TRECVID 2006 rushes exploitation task and consists of about 25 hours of rushes of travel documentaries (in French). The SEMEDIA data set is a part of the data collected by BBC (http://www.bbc.co.uk) and CCMA (http://www.ccma.cat) in the context of the SEMEDIA project (http://www.semedia.org) and consists of about 10 hours of edited news stories and complete news, sports and talk show programs (in English and Catalan).

There are slight differences in the evaluation procedure of the desktop application and the Web-based application. Therefore, we refer subsequently to *desktop evaluation* and *Web evaluation* to describe the differences.

6.2.1. Retrieval Tasks. Each of the retrieval tasks consists of a one line description of the synopsis of a video segment. The task is to use the browsing tool to locate all segments that match the given textual description. The results are collected in the result list of the browsing tool and saved at the end of the task. The result lists are then matched against a list of ground truth segments created before. The ground truth has been created based on the agreement of annotations from two annotators.

The retrieval tasks for the TRECVID BBC Rushes 2006 data set are the same as in the evaluation for the TRECVID 2006 rushes exploitation task described in [28], results can thus be compared.

6.2.2. Question Answering Tasks. The question answering tasks are only done in the evaluation of the desktop application. Each question answering task is a multiple choice question with six statements of which one or more are true. The question is a description of a scene, where each of the options is a statement about the scene. The questions are chosen so that they share the set of relevant video segments with a corresponding retrieval task.

For example, retrieval task 5 is *Find segments showing a football player scoring a goal*. The corresponding question is

Question 5: A football player scoring a goal . . .

(a) wears a green shirt,

(b) does so from a penalty,

(c) is shown in close up,

(d) is shown cheering at the sideline,

(e) wears a white shirt,

(f) is shown from the camera behind the goal.

6.2.3. Survey. The survey consists of three questionnaires: The pre-test questionnaire is completed once for each individual user who takes part in the evaluation after they are trained in the use of the browsing tool. The post-task questionnaire is completed after each task that each user finishes during the experiment. The questions of the post-task questionnaire are listed in Table 1. The post-test questionnaire is completed once for each participant after completing the last task. The questionnaire is largely based on the one used for the TRECVID 2004 interactive search task [29]. Some questions that were too specific to retrieval systems have been discarded and two questions specific to video browsing have been added to the post-test questionnaire.

6.2.4. Procedure. The evaluation session starts with an introduction of the browsing tool and an explanation of the evaluation procedure. Then the users have 10 minutes time for getting accustomed to using the browsing tool. Before starting to work on the tasks the users complete the pre-test part of the survey.

One evaluation session consists of a sequence of 4 retrieval tasks or a sequence of 4 question-answering tasks. The participants are evenly divided into 4 groups with varying assignment of task types and data sets in order to avoid an effect of the order on the results. In the evaluation

Table 1: Questions of the post-task questionnaire. Possible answers for each of the questions are: not at all, a little, fairly, quite a bit, very much.

TVB1	I was familiar with the topic of the query.
TVB3	I found that it was easy to find clips that are relevant.
TVB4	For this topic I had enough time to find enough clips.
TVB5	For this particular topic the tool interface allowed me to do browsing efficiently.
TVB6	For this particular topic I was satisfied with the results of the browsing.

of the Web-based version only retrieval tasks have been used. The working time for one task is 10 minutes including the time to complete the post-task questionnaire for each task. The users are allowed to ask staff for technical support about the use of the tool during the evaluation.

After the 4 tasks the users complete the posttest part of the survey. The total time for the session is thus about 60 minutes. Users can choose to do one or two sessions. In the latter case they work on a different type of task and a different data set in each of the sessions and complete only one pre-test survey in the first and one post-test survey in the second session.

6.3. Subjects. Desktop Evaluation: The tests have been performed with 19 users. In the pre-test part of the survey we have collected information about the subjects. According to the frequency the users use digital video retrieval systems, we introduced two groups: The first group consists of 11 subjects who never or rarely use digital video retrieval systems. The second group of 8 subjects represents more experienced users, who use digital video retrieval systems at least once a day.

Two thirds of the users search the Web or information systems more than once per day. More than half of the users were unfamiliar with the tool to be evaluated, only 10% were fairly or more familiar with it. Two thirds had no or little knowledge of the data sets used, only 17% were fairly or more familiar with all of the data.

Web evaluation: The evaluation of the Web-based video browsing tool has been performed with ten participants of our institute who are not involved in video browsing and have not participated in the evaluation of the desktop application. Only one subject is a little familiar with the TRECVID 2006 dataset, all others stated that they are not familiar with any of the datasets used in the evaluation. Eight of the subjects of this evaluation do not use any digital video retrieval system. Also eight subjects are not familiar with the video browsing tool (Web and desktop). Therefore it was not possible to create two user groups based on the experience of the users as in the desktop evaluation. Two people stated that they are a little familiar with the video browsing tool. Four search the Web very frequently, the others less frequently.

6.4. Results

6.4.1. Different User Groups. As we have two different user groups (experienced and inexperienced users) we want to determine whether the results for the groups are different. We try to reject the null hypothesis that the F1 measures, defined as $F1 = (2 * precision * recall)/(precision + recall)$, achieved by the two groups for the retrieval and question answering tasks have the same mean. For the retrieval tasks we have 24 samples from the inexperienced users (mean F1 0.26) and 12 from the experienced users (mean F1 0.23). For the question answering tasks we have 25 samples from the inexperienced group (mean F1 0.37) and 16 samples from the experienced user group (mean F1 0.45). We apply a two-tailed independent two-sample t-test which yields a P-value of .75 for the retrieval tasks and .48 for the question answering tasks, both at a significance level of 95%. The lower value for the question answering task seems to be mainly due to the lower number of samples. We can thus not reject the null hypothesis, that is, there are no significant differences between the two user groups.

6.4.2. Correlation between Methods. In order to compare the different evaluation approaches we analyze the correlation between the results. The assumption is that the F1 scores of a retrieval task and the corresponding question answering task are correlated, as well as the F1 measures with the answers to the questions TVB3-6 in the post-task questionnaire of the respective task (cf. Table 1).

The correlation coefficients between the F1 measures of the retrieval and question answering tasks are $r = -0.45$ and $\rho = -0.33$ (P-value .41). There is a slightly negative correlation between the results but no significant one. A t-test also shows at a significance level of 0.0001 that the two distributions have different means. We can conclude that retrieval and question answering tasks are not directly comparable, even if the users need a very similar result set to answer each of them.

A possible reason for the differences in results could be the imbalance in precision and recall. In the retrieval tasks, recall is typically lower than precision. Although users do not collect a result set in the question answering task, it can theoretically be seen as consisting of a retrieval task (collecting all necessary material) and an analysis of the collected material to answer the question. Low recall rates would of course decrease the ability to answer the question correctly. Another reason for the difference could be that users approach retrieval tasks ("collecting data") and question answering tasks ("fact finding") in a very different way.

Table 2 shows the correlation between the task results (again F1 measures) and the answers to the post-task questions of the respective task. The retrieval results are only correlated with question TVB6 at a significance level of 0.10,

that is, the user's satisfaction with the browsing result is positively correlated with the actual retrieval performance.

There is also only one strong correlation for the question answering task. The F1 measure is correlated with question TVB4 at a significance level of 0.10. But this is a negative correlation, that is, the users scored worse on the question answering tasks for which they felt to have more time. A possible explanation is that users feel stressed in cases where they encounter many video segments that match the query but think they have more time than when they only encounter few relevant ones.

6.4.3. Independence of Post-Task Questions. After each retrieval or question answering task the users answer the questions listed in Table 1. The correlation among the questions is shown in Table 3. There is a strong correlation among the questions TVB3, TVB4, TVB5 and TVB6, that can be accepted at a significance level of 0.10, in two cases even at a level of 0.01. These results show that it is difficult for the user to judge certain aspects separately (e.g., whether the tool was helpful in this case). Instead a general impression of the browsing experience is rated, including the satisfaction with the tool and with the results and the impression to have sufficient time.

The familiarity with the topic is not or only very weakly correlated with the other aspects. There is only a correlation with the perceived easiness of the task ($\rho = 0.67$ at significance level 0.10), that is, the task seems easier for users who are familiar with the topic. However, they do not feel that they have more time or achieve more satisfying results than others.

6.4.4. Learning Effect. We analyze this by looking for trends in the results achieved for the 4 tasks done in one session. As we have four different sequences of tasks, two different data sets and the tasks are done in different order by different users, the difficulty of individual tasks does not influence the trend. Figure 8 shows the scores of the post-task questions for the first through fourth task done by the participants. As expected, some of the measures (such as the familiarity with the search topic) do not show a clear pattern, but some seem to have a trend. If we fit a linear trend function to the data we get a clear trend for two of the questions: TVB4 (sufficient time, slope 0.22) and TVB6 (satisfaction with results, slope 0.19). The longer users work with the tool the higher is their satisfaction with the results and they perceive the working time as more sufficient.

The question is whether this trend can also be measured in the task results. When fitting a trend function to the results of the question answering tasks, we get slopes of -0.06 for precision and -0.04 for recall, that is, no clear trend can be seen, especially not a positive trend as in the survey answers. For the retrieval tasks the trend function for precision has a slope of 0.12 and that for recall of 0.01. The user's perception is supported by the precision values of the retrieval task, although the increase is not as strong as in the survey answers. The fact that the satisfaction is more related to precision than to recall can be explained as follows. The users know only about the video segments they have found, that is, not about the correct ones not found that would be measured by recall. Thus the perceived quality of the results depends on how well the segments in the result set match the query, which correlates with precision.

6.4.5. Higher Precision Score When Viewing the Video. Users have the option to view a video segment in the player before or after selecting it or to just drag the corresponding key frame to the result list without viewing the video. We can expect that viewing the video serves as a validation and thus the precision should be higher in cases where the video player has been used. Thus we try to reject the null hypothesis that the distribution of precision achieved without using the video player has a higher mean than that with using the video player. We have 46 samples, for 20 (44%) the video player has been used, the means are $\mu_{\text{withplayer}} = 0.58$ and $\mu_{\text{withoutplayer}} = 0.49$. We apply a independent two-sample t-test which yields a P-value of .25 for the one-tailed test, that is, we cannot reject the null hypothesis. It seems that users use the video player in cases where they are unsure while they add segments for which they are sure without using the player. Thus the precision for the segments added after viewing them is not significantly better.

6.4.6. Web Tool Evaluation. To determine whether the results of the retrieval tasks of the Web-based application and the desktop application are different, we have evaluated the series of measurements with two-tailed two-sample t-tests assuming different variances. We try to reject the null hypothesis in each test that the means of precision and recall of the Web and the desktop version are identical.

The first test (see also Table 4) consists of 38 samples of the Web evaluation and 34 samples of the desktop evaluation (users from our institute's staff). We get a P-value for the precision of .036 and a P-value for the recall of .222, both at a significance level of 0.05. Thus we can reject the null hypothesis for the precision, which means the results of the Web-based application are worse in contrast to the desktop application for this user group. On the other hand we cannot reject the null hypothesis for the recall values, which means there is no overall significant difference in this test setup.

The second test (see also Table 5) consists of 20 samples of the Web evaluation and 20 samples of the desktop evaluation (users from the SEMEDIA project). We get a P-value for the precision of .526 and a P-value for the recall of .387, both at a significance level of 0.05. This means we cannot reject the null hypothesis for both tests, that is, there are no significant differences of the Web-based application and the desktop application with this user group.

Figure 9 shows the scores of the post-task questions for the first through fourth task done by the participants at the Web evaluation. Only the satisfaction (TVB6) shows a slope of -0.1. This is in contrast to the desktop version where the users were more satisfied over the tasks (slope 0.19). All other questions (TVB1–TVB5) of this test do not show a clear pattern.

Table 2: Correlation between F1 measures of retrieval (R) and question answering (Q) task results and the post-task questionnaire (r denotes Pearson's product-moment correlation coefficient, ρ denotes Spearman's rank correlation coefficient and P the associated P-value).

		TVB1	TVB3	TVB4	TVB5	TVB6
R (F1)	r	−0.06	0.27	0.33	0.42	0.62
	ρ	−0.05	0.21	0.36	0.40	0.71
	P	.91	.61	.38	.33	.05
Q (F1)	r	0.23	−0.46	−0.80	−0.40	−0.46
	ρ	0.33	−0.41	−0.68	−0.16	−0.15
	P	.42	.31	.06	.70	.73

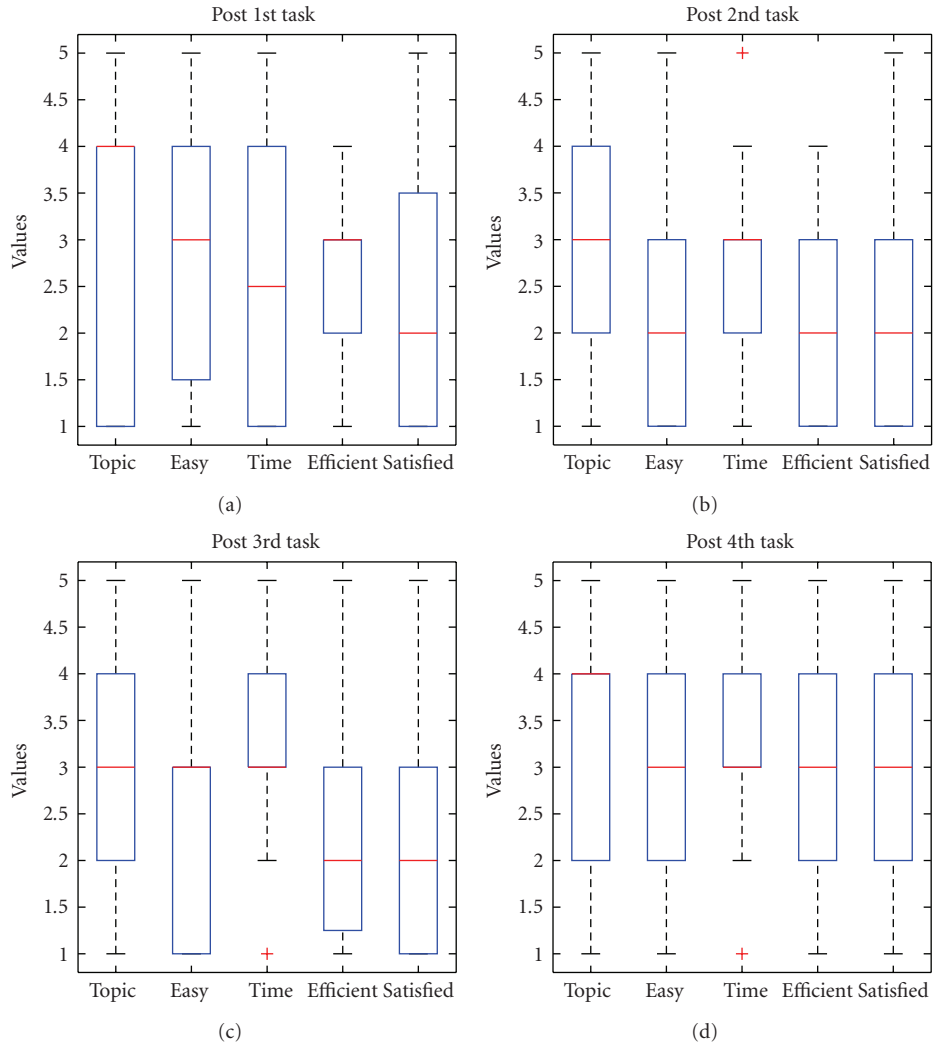

Figure 8: Results of the post-task question after each task done by the user using the desktop version.

Figure 10 illustrates the cumulated and normalized number of found items in the result list of all users during the working time. Remarkable are the tasks 1 and 6 which have both an approximating concave curve. At one third of the working time the users had about the half of the items in the result list and at the half of the working time about 70% of the items were in the result list. Furthermore, the users achieved the best results on these tasks (task 1: precision, 0.56 recall 0.30; task 6: precision, 0.65 recall 0.20).

6.4.7. Post-Test Questionnaire and User Feedback. In the post-test questionnaire we have collected information about the video browsing tool and general free text feedback of the users.

Desktop Evaluation. The half of the users stated that the response time was "fairly fast" and one third of the users are the opinion that it is "quite fast". The responses to "Learning how to use the system was easy" are as follows: not at all 4%, a little 17%, fairly 42%, quite a bit 21% and very much 13%.

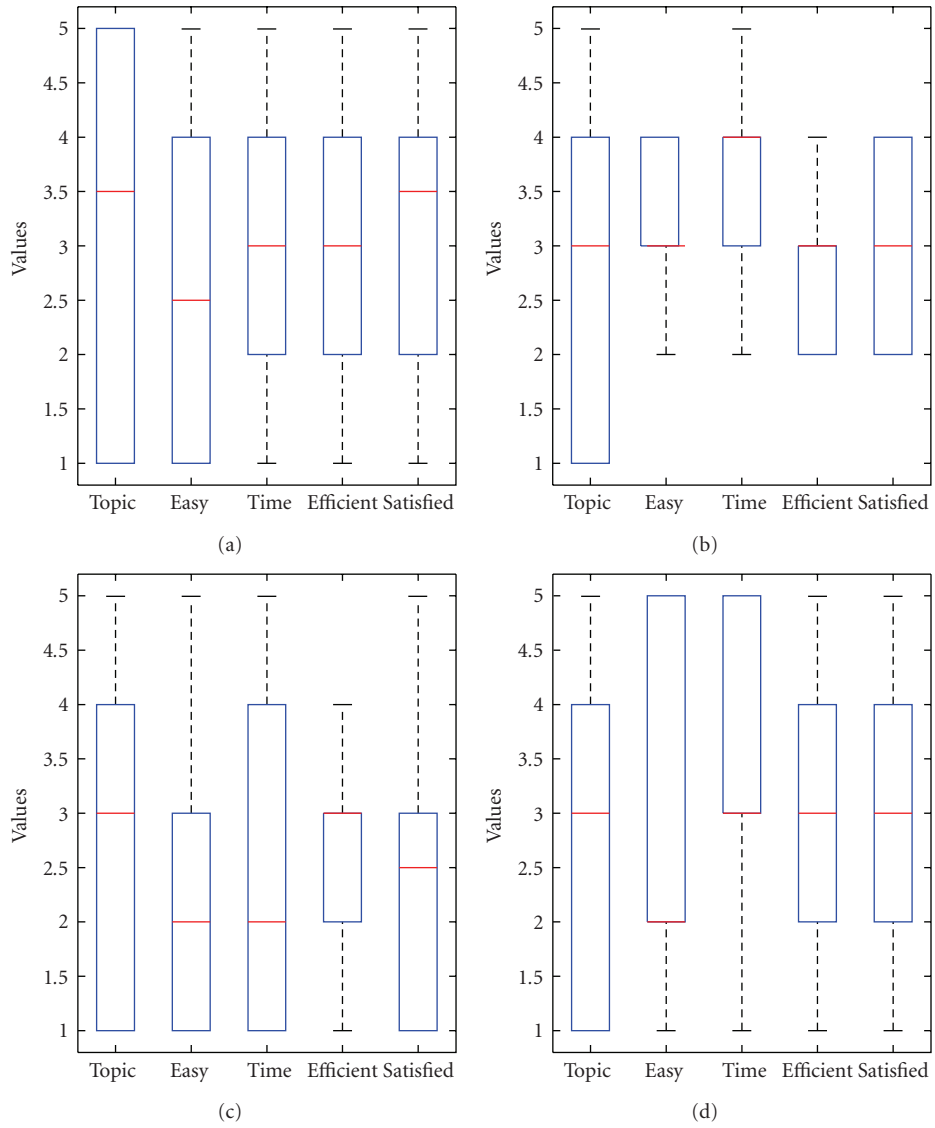

Figure 9: Results of the post-task question after each task done by the user using the Web-based version.

50% of the users answered that the system interface allowed to do the tasks efficiently is a little help, 42% answered that it is a fairly good help.

Web Evaluation. 40% of the Web users stated that the systems response time is "little fast", one third of the users choose "not fast at all" and the last third are the opinion that the system responds "fairly". The half of the users answered that it is quite easy to learn how to use the system. In contrast to the desktop version, 50% of the Web users are the opinion that the system interface "quite a bit" allows to do the retrieveal task, the other answers are: fairly 20%, a little 20% and not at all 10%.

According to the free text feedback of the users, the most annoying things of both versions are performance issues and that the cluster features sometimes produce not correct results (each 11 out of 46 answers), but 7 out of 43 answers about what users liked best of the system mention the clustering features. In 4 out of 43 answers the users wished additional features for clustering. Furthermore, 11 out of 43 answered that the interface is easy to use. Also 4 answers were about to display more metadata and information about videos, clusters and key frames.

7. Conclusion

Multimedia content abstraction approaches such as browsing applications and video summaries are of growing importance for dealing with multimedia collections. They are complementary to search and retrieval approaches and focus on problems where the formulation of a query is difficult due to the available metadata and/or the user's knowledge of the content set.

We have proposed a software implementation of a process model for multimedia content abstraction for a video browsing tool targeted at application in post-production.

Table 3: Correlation among questions of the post-task questionnaire (r denotes Pearson's product-moment correlation coefficient, ρ denotes Spearman's rank correlation coefficient and P the associated P-value). The questions TVB1 through TVB6 are listed in Table 1.

		TVB3	TVB4	TVB5	TVB6
TVB1	r	0.33	0.27	0.43	0.36
	ρ	0.67	0.50	0.54	0.33
	P	.07	.20	.17	0.43
TVB3	r		0.90	0.86	0.86
	ρ		0.86	0.86	0.64
	P		.01	.01	.08
TVB4	r			0.67	0.81
	ρ			0.64	0.74
	P			.09	.03
TVB5	r				0.84
	ρ				0.67
	P				.07

Table 4: Results of the comparison of precision (p) and recall (r) of the Web-based application and the desktop application using a two-tailed two-sample t-test assuming different variances, significance level 0.05.

	Web Evaluation		Desktop Evaluation (institute members)	
	p	r	p	r
Samples	38	38	34	34
Mean	0.342	0.156	0.525	0.206
Variance	0.11	0.026	0.148	0.032
P-value	.036	.222		

Table 5: Results of the comparison of precision (p) and recall (r) of the Web-based application and the desktop application using a two-tailed two-sample t-test assuming different variances, significance level 0.05.

	Web Evaluation		Desktop Evaluation (SEMEDIA partners)	
	p	r	p	r
Samples	20	20	20	20
Mean	0.349	0.2	0.423	0.275
Variance	0.114	0.031	0.146	0.115
P-value	.526	.387		

The browsing tool is an interactive application that allows to perform iterative clustering and selection in order to filter the content down to a manageable set of relevant items. Clustering can be performed using the features camera motion, visual activity, audio volume, face occurrences, global color similarity, repeated takes and relations in multi-view content. A number of representative frames are used to visualize a cluster. A desktop and a Web-based implementation of the client application have been presented. Concerning scalability, the tool is designed for content sets in the production workflow, which are expected to be around

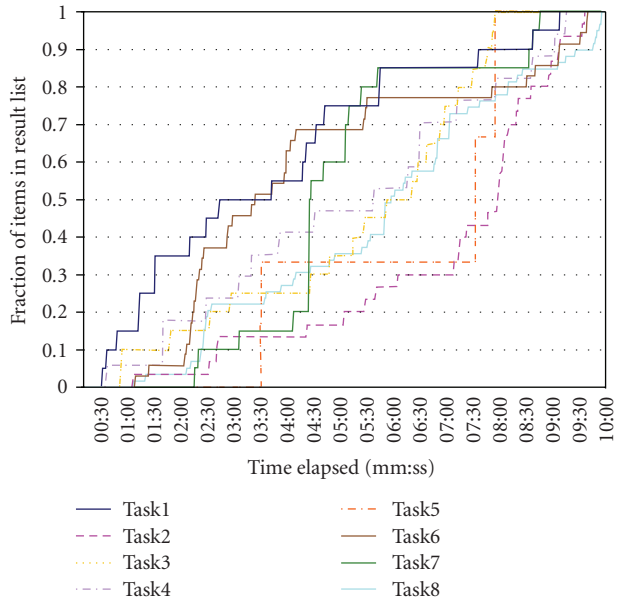

Figure 10: Cumulated and normalized number of items put into the result list of all users during the working time for the retrieval task.

100 hours per production. The response times for clustering on the complete content is not more than a few seconds for most of the features. Furthermore the increase in runtime with growing data sets is sublinear, that is, for a data set of eightfold size the time for clustering increases only by a factor between 1.3 and 3.9.

We have applied two TRECVID style fact-finding approaches and a user survey to the evaluation of a video browsing tool. We have analyzed the correlation between the results of the different methods, whether different aspects can be evaluated independently with the survey, if a learning effect can be measured with the different methods, and have compared the desktop and Web-based client applications.

In general, the results show (not unexpectedly) that especially the recall scores are rather low in such an application. This is definitely an issue that needs to be addressed in future work in video browsing.

We are also interested in comparing the different evaluation approaches for video browsing tools. We can conclude that the retrieval task correlates better with the user experience according to the survey than the question answering tasks. As retrieving relevant content is also closer to the real-world application of the tool than finding facts about the content, it seems to be the more appropriate evaluation method in this case, although it is a costly method due to the efforts for data set and ground truth preparation. Thus only retrieval tasks and a survey have been used for comparing the desktop and then Web-based client applications.

It turns out that the survey rather measures the general user experience while different aspects of the usability cannot be analyzed independently. This means that surveys are rather suitable for comparing the general usability of tools

for certain applications than for getting information about strengths and weaknesses of a certain tool.

Acknowledgments

The authors would like to thank their colleagues who contributed to the implementation of the components described here, especially Christian Schober, Harald Stiegler, and András Horti, as well as all people who took part in the evaluation sessions. The research leading to this paper has been partially supported by the European Commission under the contracts IST-2-511316-IP, "IP-RACINE—Integrated Project Research Area Cinema" (http://www.ipracine.org), FP6-045032, "SEMEDIA" (http://www.semedia.org), and FP7-215475, "2020 3D Media—Spatial Sound and Vision" (http://www.20203dmedia.eu/). BBC 2006 Rushes video is copyrighted. The BBC 2006 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

References

[1] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345–353, 1996.

[2] B. T. Truong and S. Venkatesh, "Video abstraction: a systematic review and classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, no. 1, article 3, 2007.

[3] W. Bailer and G. Thallinger, "A framework for multimedia content abstraction and its application to rushes exploration," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR '07)*, pp. 146–153, Amsterdam, The Netherlands, July 2007.

[4] J. Oh and K. A. Hua, "An efficient technique for summarizing videos using visual contents," in *Proceedings of IEEE International Conference on Multi-Media and Expo (ICME '00)*, pp. 1167–1170, New York, NY, USA, 2000.

[5] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Video abstracting," *Communications of the ACM*, vol. 40, no. 12, pp. 55–62, 1997.

[6] M. G. Christel, A. G. Hauptmann, A. S. Warmack, and S. A. Crosby, "Adjustable filmstrips and skims as abstractions for a digital video library," in *Proceedings of the Forum on Research and Technology Advances in Digital Libraries (ADL '99)*, pp. 98–104, Baltimore, Md, USA, 1999.

[7] D. Ponceleon, "Hierachical brushing in a collection of video data," in *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS '01)*, vol. 4, p. 116, IEEE Computer Society, Washington, DC, USA, 2001.

[8] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang, *A Unified Framework for Video Summarization, Browsing and Retrieval: With Applications to Consumer and Surveillance Video*, Academic Press, New York, NY, USA, 2005.

[9] P. Chiu, A. Girgensohn, W. Polak, E. Rieffel, and L. Wilcox, "A genetic algorithm for video segmentation and summarization," in *Proceedings of IEEE International Conference on Multi-Media and Expo (ICME '00)*, vol. 3, pp. 1329–1332, New York, NY, USA, 2000.

[10] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: generating semantically meaningful video summaries," in *Proceedings of the ACM International Multimedia Conference & Exhibition (ACMMM '99)*, pp. 383–392, Orlando, Fla, USA, November 1999.

[11] M. M. Yeung and B.-L. Leo, "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, 1997.

[12] M. A. Smith and T. Kanade, "Video skimming for quick browsing based on audio and image characterization," Tech. Rep. CMU-CS-95-186, Carnegie Mellon University, Pittsburgh, Pa, USA, July 1995.

[13] J. R. Smith and S.-F. Chang, "Image and video search engine for the World Wide Web," in *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Sience and Technology (IE '97)*, vol. 3022 of *Proceedings of SPIE*, San Jose, Calif, USA, February 1997.

[14] G. Geisler, G. Marchionini, B. M. Wildemuth, et al., "Video browsing interfaces for the open video project," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI '02)*, pp. 514–515, ACM, New York, NY, USA, 2002.

[15] Y. Rui and T. S. Huang, "A unified framework for video browsing and retrieval," in *Image and Video Processing Handbook*, A. C. Bovik, Ed., pp. 705–715, Academic Press, New York, NY, USA, 2000.

[16] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *Proceedings of the ACM International Multimedia Conference and Exhibition (MULTIMEDIA '02)*, pp. 189–198, New York, NY, USA, 2002.

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

[18] I. Campbell and C. J. van Rijsbergen, "The ostensive model of developing information needs," in *Proceedings of the 2nd International Conference on Conceptions of Library Science (COLIS '96)*, pp. 251–268, Copenhagen, Denmark, 1996.

[19] W. Bailer and P. Schallauer, "The detailed audiovisual profile: enabling interoperability between MPEG-7 based systems," in *Proceedings of the 12th International Multi-Media Modelling Conference (MMM '06)*, H. Feng, S. Yang, and Y. Zhuang, Eds., pp. 217–224, Beijing, China, January 2006.

[20] H. Stiegler, "Module developers guide," Tech. Rep., JOANNEUM RESEARCH, Institute of Information Systems & Information Management, Graz, Austria, 2007.

[21] W. Bailer, P. Schallauer, and G. Thallinger, "JOANNEUM RESEARCH at TRECVID 2005—camera motion detection," in *Proceedings of the TRECVID Workshop*, pp. 182–189, Gaithersburg, Md, USA, November 2005.

[22] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[23] "Information technology-multimedia content description interface—part 3: visual," ISO/IEC 15938-3, 2001.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.

[25] W. Bailer, F. Lee, and G. Thallinger, "A distance measure for repeated takes of one scene," *Visual Computer*, vol. 25, no. 1, pp. 53–68, 2009.

[26] W. Bailer and H. Rehatschek, "Comparing fact finding tasks and user survey for evaluating a video browsing tool," in *Proceedings of the ACM Multimedia Conference, with Co-located Workshops and Symposiums (MM '09)*, pp. 741–744, Beijing, China, October 2009.

[27] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 321–330, Santa Barbara, Calif, USA, 2006.

[28] W. Bailer, C. Schober, and G. Thallinger, "Video content browsing based on iterative feature clustering for rushes exploitation," in *Proceedings of the TRECVID Workshop*, pp. 230–239, Gaithersburg, Md, USA, November 2006.

[29] A. Smeaton and P. Wilkins, "TRECVID 2004: Interactive search questionnaires," http://www-nlpir.nist.gov/projects/tv2004/questionnaires.html.