A Graph-Based Approach and Analysis Framework for Hierarchical Content Browsing

Markus Rickert
Technische Universität Chemnitz
Straße der Nationen 62
D-09111 Chemnitz, Germany
markus.rickert@cs.tu-chemitz.de

Benedikt Etzold
Technische Universität Chemnitz
Straße der Nationen 62
D-09111 Chemnitz, Germany
benedikt.etzold@cs.tu-chemitz.de

Maximilian Eibl
Technische Universität Chemnitz
Straße der Nationen 62
D-09111 Chemnitz, Germany
maximilian.eibl@cs.tu-chemitz.de

ABSTRACT
Systems for multimedia retrieval have been object of scientific research for many years. When it comes to present results to the user many solutions disregard the set of problems connected to content delivery. Especially time-constrained results of video retrieval systems need a different visualization. In this paper we present our solution for hierarchical content browsing of video files. Our workflow covers the phases of ingest, transcoding, automatic analysis, intellectual annotation and data aggregation. We describe an algorithm for the graph-based analysis of the content structure in videos. By identifying the requirements of professional users we developed a user interface enabling to access retrieval results in different hierarchical abstraction levels.

Author Keywords
Content-browsing, video analysis, video retrieval, graph-based analysis, visualization, algorithm, user interface

ACM Classification Keywords
E.1 [**Data Structures**]: *Graphs and networks*, H.5.1 [**Information Interfaces and Presentation**]: Multimedia, I.2.10 [**Vision and Scene Understanding**]: *Video analysis*,

INTRODUCTION
Compared to other areas of information retrieval, the content-browsing of audiovisual media bears special challenges. Videos are time-dependent. Usually the user's intention is to find an element inside a video, depicting a certain semantic concept like a person, topic, location or event. By querying a video database, the returned result is either a complete video item or a single element inside a video item determined by its time position. Professional users are not mainly interested in finding only a single occurrence of the queued semantic concept. They want to gather the whole sequence related to their search query, e.g. to reuse it in a news report or for historic research. The user usually sees the retrieval result as a starting point for a further manual searching process inside the video item which is operated by using the playback and seek functions of the player software.

"3rd International Workshop on Interactive Content Consumption at TVX'15, June 3rd, 2015, Brussels, Belgium.

In this paper we present our approach to provide a hierarchical presentation of video items to support professional users while browsing and consuming the content of a media retrieval system. Based on the primary focus of video content from television programs, this solution works best on video material edited in a post-production workflow. It is not supposed to be used on e.g. surveillance videos. Our framework has been developed to provide automatic and intellectual annotation to historical television recorded on video tapes. The digitized master copies and their metadata can be searched and displayed in a web-based user interface (UI). Video shots and sequences can be explored as a hierarchical structure in the UI. The system is in use in a pilot project by the "media state authority of Saxony" (Sächsische Landesmedienanstalt) in Germany.

USER REQUIREMENTS & EXISTING WORKFLOWS
Our use case focuses on user groups in professions that rely heavily on reviewing large amounts of video data on a daily basis like journalists, editors or historians.

In a set of interviews we asked a group of experts to describe their daily work. Thereby, we especially focused on those areas that deal with the examination of the results of archive queries. Other fields of interest were the process of querying, preferred software solutions and the planning of new reports or videos. Our findings were subsequently merged into an extensive workflow that was used for identifying different problem areas.

Altogether, we spoke to three experts from three different German TV stations, who all work in the field of TV journalism. Their similar statements and their reports on the workflows of other professionals and institutions give reason to believe that our workflow is representative for a significant part in this field of work. Conducting surveys and interviews [17, 18], we identified some of the main problems they face as a part of their working routine:

- *Metadata* is often either fragmentary, or missing completely. While standards or recommendations exist in most professions, they are usually ignored due to bottlenecks in time and personnel.
- *Video data* is normally stored in its final state, e.g. a film that has already been edited in post-production. In the case of search queries returning more than one result, users often receive a single file containing a queue of all relevant video files.

- In *TV production*, time pressure is always high because of narrow schedules and the need for instant coverage of current events.

Specific software solutions addressing these issues do not yet exist in professional scenarios. This leads to a highly inefficient workflow: Precision rates are usually low because of the described storing modalities and the lack of precise metadata. Therefore, numerous files of comparatively large size have to be inspected in a short period of time.

Classical User Interfaces

The software that is used is normally designed to handle the simple consumption of video content (e.g. VLC Media Player or Apple QuickTime) or the tasks of professional post-production (e.g. Avid MediaComposer or Adobe Premiere). Both approaches are based on a perspective that emphasizes the linear structure of the completed video whilst or after the process of editing. By showing an ordered sequence of single shots, they present the content in consideration of the editor's intention but not of the needs of an expert using a retrieval system.

Requirements

Based on these findings, we compiled a list of requirements that have to be met by a user interface to improve the user experience significantly:

- Metadata is usable for both video processing and visualization.
- Information can be displayed based on the video's structure.
- Richness of detail can be increased for single segments of the video.
- The video itself can be accessed through any bit of information displayed in the UI.
- Relevant segments of the video can be used in later steps of the user's workflow, e.g. editing.

FRAMEWORK

Our framework provides functionalities for audio and video analysis, manual annotation, data warehousing, retrieval and visualization. It uses specialized components for each aspect. The core "dispatcher" is controlling the analysis process, allocation of work units and data aggregation. As deduced from [8], the requirements for a scalable analysis system based on heterogeneous scientific algorithms on the field of audio and video analysis are complex. The framework is presented here in its complete workflow for the first time. Earlier publications covered only aspects of distinct components. A predecessor partial framework was presented in [3].

Our framework needs to support individual solutions, programmed in varying languages, based on different operating system environments and requesting various quantities of resources. Therefor it runs in an environment of virtual machines on a cluster of five Intel Xeon dual-quad-core host servers. The main components were written in C# .Net source code and make use of service-orientated-architecture and web services. This provides a redundant and hardware independent service, while supporting a variety of separate execution environments for each component. It also allows for a possible scale-out with additional hardware if needed.

The execution workflow for an individual video tape or file consists of five phases, as depicted in Figure 1. On the level of each stage it is intended to reach a maximum of concurrency.

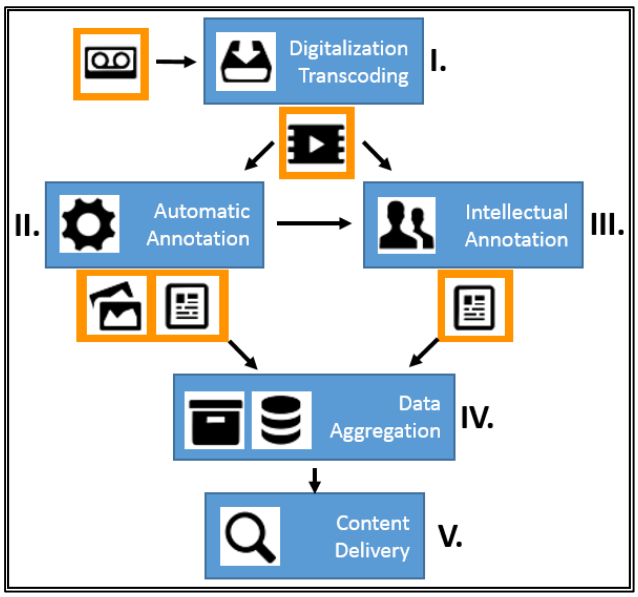

Figure 1: Framework Workflow in its five Phases.

I. Digitization and Transcoding

The very first step is an incoming control of each video tape and the generation of a unique identifier. Our id system consists of a 12-byte block and can be represented and displayed for human reading as a combination of 12 hexadecimal digits (e.g. 0000-0074-0000-0026-Z) in four segments plus a calculated check character. After the initial logging, the video tape is digitized with an automatic robot ingest system as described in [12]. It is running batch jobs in parallel on up to six tape players.

The resulting digital master file is encoded as a broadband IMX50 video codec captured in an mxf-container for archiving and data exchange. As defined by [8] we create proxy versions of the archive file by transcoding it. For automatic annotation, analysis and as a preview video for the web UI, we use an h.264 codec at level 4.1 wrapped in an mp4-container.

II. Automatic Analysis and Annotation

The created analysis proxy video is transferred to the analysis cluster. The dispatcher schedules the analysis of each video file as a sequence of consecutive analysis steps. For performance reasons, each component can be instantiated multiple times. In the common configuration, the system runs with up to 12 individual virtual machines. The analysis components are controlled by the dispatcher via a web-service interfaces.

Shot detection component

The shot detection is the first component in the workflow. It provides a segmentation of the continuous video stream into parts of uninterrupted camera recordings (shots). The algorithms developed by [9] and [10] are based on calculating the cumulated error rate of individual motion vectors for each image block between two successive frames. The component's output is a list of metadata for every detected shot. Key frames of the shot are extracted for use in the UI and successive components.

Face detection component

The face detection component uses the key frames from the shot detection to mark bounding boxes around each detected face. The used algorithm is optimized for high precision and developed by [10]. It is specialized on data corpora from local television broadcasts. Its result data is a set of metadata of the bounding box around detected faces and a sample image for each detected face.

Text extraction component

The text extraction component detects areas of overlay text boxes within the video steam. The algorithm by [11] uses a weighted discrete cosine transform (DCT) to detect macroblocks by finding regions located in the medium frequency spectrum. By normalizing the eigenvalues, a mask is calculated which is used to separate the textbox from the rest of the image. For text to character transformation the software tesseract-ocr is used (https://code.google.com/p/tesseract-ocr/). The component creates key frame samples of the detected textboxes, metadata about the locations of the textboxes and the extracted text from the OCR (optical character recognition).

Speech Recognition

The Speech Recognition component makes use of the speaker change recognition method described by [6] and extended by [3]. It provides data for the differentiation of individual voices and pauses. By applying Gaussian Mixture Models individual speaker can be trained and recognized. The detected utterances of individual speakers are transferred to an automatic speech recognition (ASR) software. The resulting data provides not only the recognized words. It adds metadata about the time position and duration of the utterance and an id code for identification and re-recognition of the speaker.

III. Intellectual Annotation

This framework is not only used for demonstration of our solutions. It is in productive use for archiving historical tape based material. This constitutes the need for additional intellectual annotation, since today's automatic annotation can provide support, but it cannot substitute the intellectual work of a human entirely. Secondly, the manual annotated metadata is used as training sets and test sets for the development of new algorithms. Therefor we collect metadata for each video tape in form of classical intellectual annotation as it is already implemented in media archives.

Scene & Topic Annotation

We developed a web-based annotation tool for the intellectual annotation of the analyzed video files. To support the professional user, the tool makes use of the detected video shots. The video is presented in slices of camera shots. The video player repeats the current shot in a loop. This makes it easier for the user to fill out all input fields, without constantly dealing with the player controls. When the user is finished with the shot, he can jump to the next. The user marks the boundaries of storyline sequences as collections of multiple shots and adds a variety of bibliographical metadata like title and type of video content, topics and subjects in terms of individuals, locations, institutions, things, creations (like art) and other metadata useful for either information retrieval or as development test data.

IV. Data Aggregation

In the past we were only analyzing video assets for isolated scientific experiments. To process large quantities of videos now, the integration of results from different analysis algorithms becomes a key challenge. For our environment of use cases, a data-warehouse solution is needed to aggregate more than only the results of video analysis. On the one hand it needs to incorporate the metadata supplied from its sources, like production information and data from TV broadcasters. On the other hand it has to provide its data as an export artifact witch is compatible with formats and conventions used by achieving facilities and institutes.

A special challenge was to find a scheme, which complies with the way video content producers and archives structure their data, and includes technical data, like feature-vectors and audiovisual classifications. Our selected database-scheme is adapted from a common standard for video documentation[1] developed for the German public television. We combined metadata fields from the point of mandatory and optional meta-data classes with the goal to maintain a maximum of compatibility.

V. Content Delivery and Visualization

For data exchange and archiving, the digital master file, the proxy files and the metadata are exported to a LTO tape library. Search and content access for the user are provided by a webserver.

The user interface is used for web-based intellectual annotation, controlling the analysis process, information retrieval and for content browsing. The UI is able to handle multiple tenants and has as scalable interface for different display resolutions or devices. Each function runs in its own web-app.

[1] (REM http://rmd.dra.de/remid.php?id=REM_APR_3)

GRAPH-BASED VIDEO CLUSTERING

During the analysis and automatic annotation we extract segments of camera shots from the video stream. This shot-segmentation is helpful for content browsing, but it suffers from over-segmentation. The structure is too detailed for the visualization of the actions inside a video. The user needs to be able to search for scenes or sequences as basic units.

Procedure Sequence-Graph
input: List of detected shot-boundaries and transitions Sh, list of sequences Sq.
output: Sequence-graph $G_1(E_a, V_a)$
1. **for each** *detected shot and transition* sh_i **from** Sh **do**
2. add new vertex Vs_i to G_1
3. add new edge Es_i to G_1 connecting Vs_i and Vs_{i+1}
4. **end for each**
5. **for each** *sequence* sq_j **from** Sq **do**
6. create new Va_j in G_1
7. **end for each**
8. **for each** Vs_i **from** G_1 **do**
9. **if** Vs_i *belongs to sequence* sq_j **then**
10. remove Vs_i and it's out-edges and in-edge from G_1
11. add Vs_i and it's edges as sub-elements to the Va_j
12. **end if**
13. **end for each**
14. **for each** Es_i **removed from** G_1
15. add new Ea_i edge in G_1 connecting Va_j with the predecessors resp. successors of Vs_i
16. **end for each**
17. **for each** Ea_k **from** G_1 **do**
18. **if** *more than one* Ea *exists with the same source-vertex and the same target-vertex as* Ea_k **than**
19. remove all duplicates and increment the weight of Ea_k
20. **end if**
21. **end for each**

Figure 2: Procedure to create a Sequence-Graph.

Different approaches for clustering or grouping of related shots were published. A detailed survey on the field of video segmentation is given by [13].

A common strategy in many clustering approaches is to find structures and similarities in the given video. The similarity measurement can be based on classification of e.g. motion vectors, dominant color, edge histogram and editing tempo. By calculating the similarity of consecutive shots, groups can be identified. "Overlapping-links" introduced by [16] was one of the early strategies to find structures inside of videos. It was extended by [19, 20]. The algorithm can cluster similar shots and the shots laying in between as a Logical shot units (LSUs) [21].

Our solution was inspired by overlapping-links, the concept of a Scene-Transition-Graph (STG) [14, 22] and the Scene Detection solution published by [15]. These approaches are still subject to actual publication and optimizations like [23, 24, 25]. Shots are represented as nodes, transitions as edges. Shots with a high similarity are clustered into group-nodes. This process leads to a digraph with cycles.

		Description
Es	→	*Singular Edges* – Directed edge between two Singular Nodes (Ns) representing the transition from a camera shot to its successor in the sequence of the video.
Vs	◯	*Singular Nodes* – A single continuous camera shot.
Ea	⇢	*Aggregated Edge* – Directed edge between two Aggregated Nodes (Na) or between a Singular Node and an Aggregated Node. It represents a set of interrelated Singular Nodes, respectively a sub-graph containing a scene in the video.
Va	□	*Aggregated Node* – A group of Singular or Aggregated Nodes as a sequence or sub-graph.
C	●	*Color-Similarity-Group* (C) — A list of shots, grouped by its visual similarity. The similarity is measured by a combination of the MPEG-descriptors Edge-Histogram (EHD) and Color-Layout (CLD). [10 pp.169]
Sq	⊥	*Sequence-List* (Sq) – A List of shots, grouped by their affiliation to a sequence, found by intellectual annotation. A sequence represents a segment of continuous action or location in a video.

Table 1: Data structures.

	Metadata & Parameter
Es	• Duration of the transition. • Type of transition (cut, wipe, dissolve, fade). As described in the taxonomy by [7]
Vs	• Number of the shot. • Times of start, duration and end of the shot • Extracted keyframes of the first and last frame. • Extracted keyframes from face detection • Data from text extraction
Ea	• Weight.
Va	• A representative keyframe. • Start-time of the earliest sub-element. • End-time of the latest sub-element. • Metadata of the speech recognition. • Annotation: topic, location, subjects, individuals etc.

Table 2: Metadata available in the data structures.

Data Structure

Our proposed solution is derived from the concept of shot-transition-graphs. We use a weighted directed graph for the representation of hierarchical sequence structures in a video. Edges represent transitions between distinct shots or sequences. Nodes represent single shots or sub-segments with a new graph of shots inside. See Table 1.

Sequence-Graph-Algorithm

In order to access the video content in a graph based hierarchical structure, we create a directed graph to represent the video's shots and sequences. The vertices belonging to a

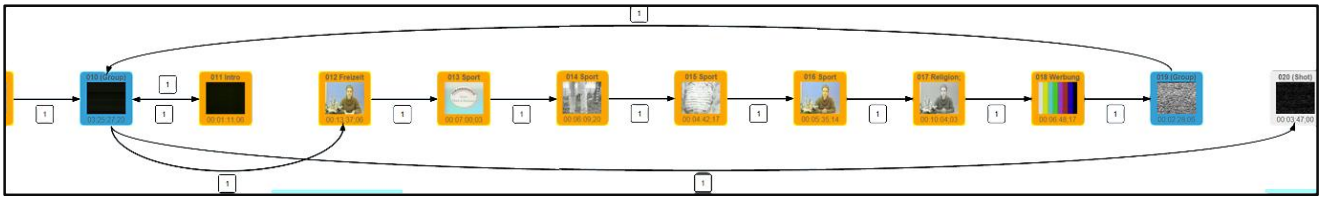

Figure 3: Visualization of vertices and edges of the Sequence-Graph

sequence are aggregated to build a second level in the hierarchy. Metadata created during the intellectual annotation performs the aggregation.

Procedure Similarity-Graph
input: List of Color-Similarity-Groups C, graph G_1
output: Sequence-Graph with Similarity-Subgraphs G_2

1. **for each** Va_i **from** G_1 **do**
2. create new temporal graph Gt_i
3. **for each** *similarity group* sq_j **from** Sq **do**
4. **if** one or more *sub-vertex* Vs_i *of* Va_i *is* $\in$ sq_i **then**
5. add new group-vertex Va_j to Gt_j
6. **end if**
7. **end for each**
8. **for each** *sub-vertex* Vs_i *of* Va_i **not from** sq_i **do**
9. add new non-group-vertex Vs_j to Gt_j
10. **end for each**
11. **for each** *sub-edge* $Es \cup Ea$ **from** Va_i **do**
12. add new edge Ea_j to Gt_j connecting the corresponding vertex of its sources group-vertex Va and its targets group-vertex Va, respectively the non-group-vertex Vs if source or target is not part of an similarity group sq_i.
13. **end for each**
14. Calculate the strongly connected components of Gt_j
15. **for each** strongly connected component scc_k **from** Gt_j **do**
16. create new similarity-vertex Va_k as sub-element in Va_i
17. **for each** shot-vertex Vs_l **from** Va_i **do**
18. **if** Vs_l *is member of* scc_k **then**
19. remove Vs_l and it's out-edges and in-edge from Va_i
20. add Va_l and it's edges as sub-elements to the Va_k
21. **end if**
22. **end for each**
23. **for each** Es_m *removed* **from** Va_i **do**
24. add new Ea_m edge in Va_k connecting Va_k with it's predecessors resp. successors
25. **end for each**
26. **end for each**
27. **for each** edge Ea_m **from** Va_i **do**
28. **if** *more than one* Ea *exists with the same source-vertex and the same target-vertex as* Ea_k **than**
29. remove all duplicates and increment the weight of Ea_m
30. **end if**
31. **end for each**
32. **end for each**

Figure 4: Similarity-Graph Procedure

The resulting Sequence-Graph (Algorithm in Figure 2) is representing all content sequences as aggregated nodes and the remaining singular nodes not belonging to a sequence on the first level. Inside each aggregated node a sub-graph was created on the second level, representing the chain of shots forming a sequence.

Similarity-Graph-Algorithm

One important feature of videos from film and television is the presence of recurring images. This happens especially when interviews or dialogs are recorded where the same individuals are shown several times. In terms of film grammar this is called the shot-/ reverse-shot method. See Figure 4

Resulting Graph Structure

The final resulting graph represents the video in a hierarchical structure. On the first level all sequences and all standalone shots can be accessed. By selecting a sequence all shots and similarity groups inside the selected sequence can be accessed. If a shot shows a similar image multiple times, each instance of this image is aggregated to a group. Recurring shots are recognizable by cyclic structures of the edges. On selecting a similarity group the individual instances of the similar shots can be accessed. The results of the two clustering-steps and the final 3-layer graph are shown in Figure 5. Figure 6 shows die visualization of a single layer as used in the UI.

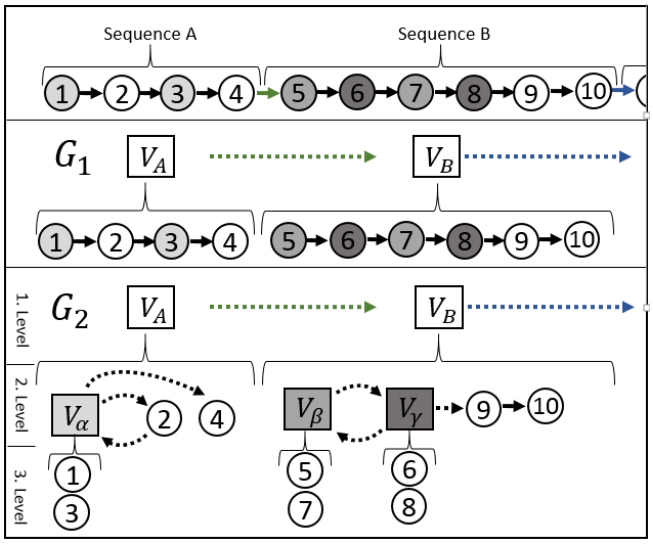

Figure 5: G_1(Sequence-Graph), G_2 (Similarity-Graph and Sequence-Graph)

GRAPH-BASED USER INTERFACE

UI approaches with the purpose of addressing structures in video content have been developed mainly in in the fields of film studies and in human-computer-interaction (HCI). They

normally focus on certain key aspects like analysis, description [4] or summarization [1] of content. From their perspective, the temporal order of a video's single sequences is an important bit of information and therefore one of the fundamental principles of their modus operandi.

By shifting the main focus to the video's structure, we managed to design a user interface that makes it possible to quickly overlook a whole file without losing any detail.

Graph-based User Interface

In order to avoid the issues reported by our user groups, as described above, we decided to organize all available information in a way that emphasizes the video's structure. Richness of detail is increased from top (overview) to bottom (all details and metadata). The presented metadata-types are summarized in Table 2. The following interface description is connected to the layers presented in the Figures 6 and 7.

I. *Video player* – The player can be used to examine the single segments in any intended way. In order to provide permanent availability, it remains at the top of the screen when scrolling to the lower parts of the UI**.**
II. *Current graph* – Its nodes represent either a single shot group or a cluster of related groups. By using a simple directed graph for the top level, we were able to display all nodes in a familiar left-to-right-order. Every node contains a representative image sample and some basic information on its content. The existence of child graphs is color coded (blue) on this level of detail.
III. *Collapsible container* that is used to display a more granular child graph belonging to a certain top-level node.
IV. *Queue* – Nodes can be transferred in a drag-and-drop operated queue of cards that offer a more detailed view of their content. Furthermore, they can be used to manage a collection of shots or shot groups that can be watched directly or exported for further use e.g. in editing software.
V. *Details-view* – shows all data that is available for one of the cards. It consists of several lines displaying key frames, detected faces, off text and text overlays.

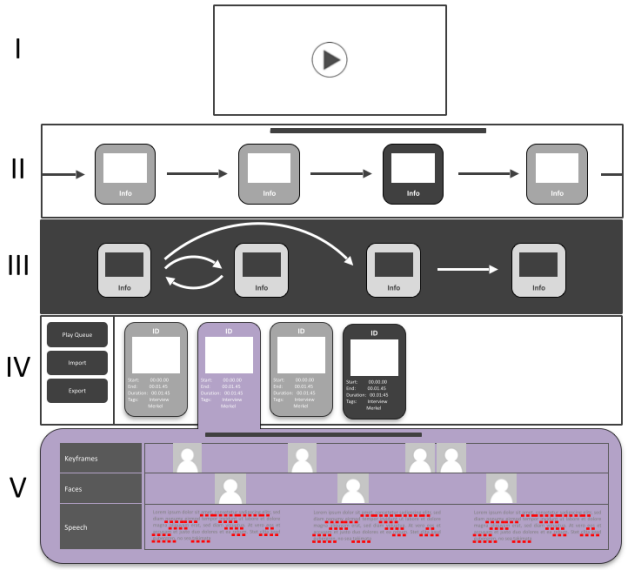

Figure 7: Schematic View of the UI.

EVALUATION

We performed a first evaluation of our approach by using a combination of baseline tests and questionnaires. Therefore, we designed a set of tasks comparable to those described by our group of experts. A screenshot of the graph-based UI is depicted in Figure 7. The content used for evaluation consists of real television news programs, produced during the early to mid-1990s. It was archived on VHS video tapes. The actual test-set was composed by randomly selecting 1377 minutes of this video material.

Four expert users were asked to perform searching tasks. They were given short descriptions of 27 randomly picked video sequences with durations between 5 seconds and 10 minutes. The task was to find the described sequences in the corresponding video file and to write down the time codes of the sequence boundaries. Searching tasks like these are quite comparable to the real live work of video editors, because video content in tape based archives is only marginally documented. Manual content browsing in a video player and non-linear editing software (NLE) is used to find sequences of video content reusable in new video clips.

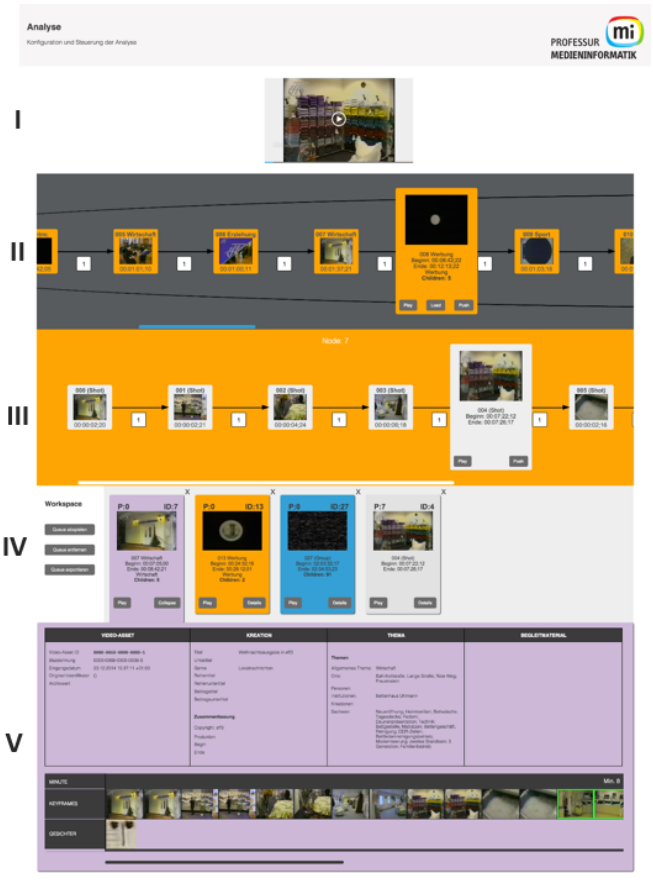

Figure 6: Multilayer-View of a Graph-based UI.

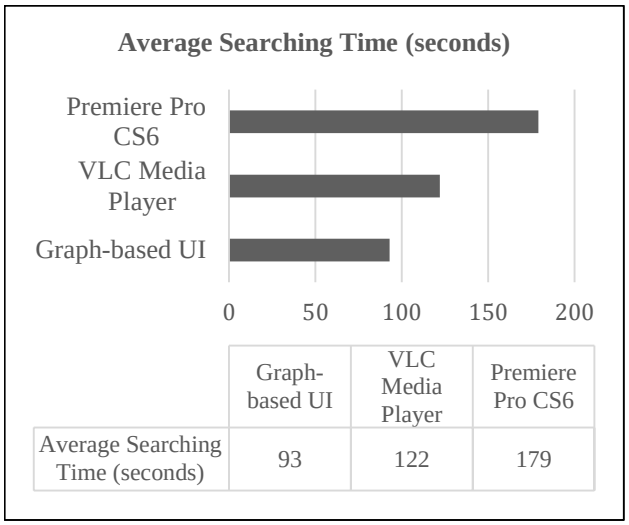

Figure 8: Evaluation results.

For comparison, the searching tasks were performed by using our graph-based user interface, VLC Media Player and Adobe Premiere Pro (CS 6). For each task the time needed for completion was recorded. Overall, 108 different search operations were performed. Furthermore, differences in the accuracy of the time-codes were taken into account. With the graph-based UI, the average duration per searching task was 93 seconds. When searching with VLC (average: 122 s) and Premiere Pro (average: 179s) significantly more time was needed (Figure 8). As a result our graph-based solution outperformed VLC and Premiere Pro. In VLC 27.8% more time was needed. Searching in Premiere Pro needed 48.5% more time. One reason for the weak performance of Premiere Pro could be the zoom function. It was heavily used by the testers, but leaded to longer searching times.

One disadvantage of the graph-based UI turned out to be the fact that entities and events inside a video shot cannot be isolated. They are bound to the boundaries of the surrounding shot and cannot be exported independently. In terms of perceiving the actual structure of the video, all users reported gaining a deeper understanding when using our approach than when using VLC or Premiere Pro.

FUTURE WORK

The next step for the analysis and graph-based clustering will be the substitution of the manual annotation of video sequences by an automatic sequence segmentation algorithm. Surveys on the state-of-the-art in video-segmentation indicate that a multimodal fusion of the analysis results can be used to cluster successive shots into video sequences. Most approaches use visual similarity features. But as discussed in [7], concepts and rules from the production of video content can be useful to find sequences or scenes inside video content.

The graph-based user interface will be evaluated in additional user tests, exploring if its use is beneficial for non-professional users as well. A second study will evaluate, which text-based metadata should be presented at the different elements to comply the need of the users. Currently, extensions of the UI are under development to enable a sync-function. It will allow adapting the presented graph elements when the current position in the video shifts to the next sequence. This will give the UI a two-sided interaction between the video player and the graph structure.

CONCLUSION

In this paper we presented our concept of a hierarchical presentation of video items in a graph-based structure. We described our framework which incorporates video and audio analysis, intellectual annotation and graph analysis to construct a multi-layer structure for content-consumption. Our web-based UI shows how classical sequential content browsing in videos can be extended to incorporate the inner structures and relations of the video's sub-elements.

ACKNOWLEDGMENTS

Parts of this work were accomplished in the research project validAX funded by the German Federal Ministry of Education and Research.

REFERENCES

[1] Adami, N., Benini s., Leonardi R. An overview of video shot clustering and summarization techniques for mobile applications. *In Proc MobiMedia '06,* ACM (2006), No. 27

[3] Knauf, R., Kürsten, J., Kurze, A., Ritter M., Berger A., Heinich S., Eibl M. Produce. annotate. archive. repurpose --: accelerating the composition and metadata accumulation of tv content. *In Proc. AIEMPro'11,* ACM (2011), 30-36

[4] Korte, H. Einführung in die systematische Filmanalyse. *Schmidt (1999)*, 40.

[6] Lu, L. and Zhang, H.-J. Speaker change detection and tracking in real-time news broadcasting analysis. *In Proc MULTIMEDIA'02*, ACM (2002), 602-610.

[7] Rickert, M. and Eibl, M. A proposal for a taxonomy of semantic editing devices to support semantic classification. *In Proc. RACS 2014*. ACM (2014), 34–39.

[8] Rickert, M. and Eibl, M. Evaluation of media analysis and information retrieval solutions for audio-visual content through their integration in realistic workflows of the broadcast industry. *In Proc. RACS 2013*. ACM Press (2013), 118–121.

[9] Ritter, M. and Eibl, M. 2011. An Extensible Tool for the Annotation of Videos Using Segmentation and Tracking. Lecture Notes in Computer Science. Springer Berlin Heidelberg. 295–304.

[10] Ritter, Marc. Optimierung von Algorithmen zur Videoanalyse. Chemnitz (2013), 1–336. ISBN 978-3-944640-09-9, 119-144, 187-213

[11] Heinich, S. Textdetektion und -extraktion mit gewichteter DCT und mehrwertiger Bildzerlegung, *In Proc. WAM 2009*, TU-Chemnitz (2009), 151-162, ISBN 978-3-000278-58-7

[12] Manthey, R., Herms, R., Ritter, M., Storz, M., Eibl, M. A Support Framework for Automated Video and Multimedia Workflows for Production and Archive. *Proc. HCI International 2013*. Springer (2013), 336-341.

[13] Del Fabro, M., Böszörmenyi, L. State-of-the-art and future challenges in video scene detection: a survey. *Journal Multimedia systems Vol. 19. Issue 5*. Springer (2013), 427–454

[14] Yeung, M.M., Boon-Lock Y. Time-constrained clustering for segmentation of video into story units, *In Proc. 13th IC on Pattern Recognition*, IEEE (1996), 375-380 vol.3, doi: 10.1109/ICPR.1996.546973.

[15] Ngo, C.-W., Ma, Y.-F., Zhang, H.-J. Video summarization and scene detection by graph modeling. *In Circuits and Systems for Video Technology vol. 15, issue 2*, IEEE (2005), 296–305.

[16] Hanjalic, A., Lagendijk, R. L., Biemond, J. Automated high-level movie segmentation for advanced video-retrieval systems. *In Circuits and Systems for Video Technology vol. 9, issue 4*, IEEE (1999), 580-5

[17] Berger, A.; Kürsten, J.; Eibl, M.: Visual String of Reformulation. *In Proc HCI International*, Springer (2009) - LNCS 5618

[18] Berger, A: Design Thinking for Search User Interface Design. *In Proc EuroHCIR2011*, Newcastle, (2011), 38-41

[19] Kwon, Y.-M., Song, C.-J. and Kim, I.-J. A new approach for high level video structuring. *In Proc. Multimedia and Expo*, ICME 2000, 773–776.

[20] Wang, W. and Gao, W. Automatic segmentation of news items based on video and audio features. In Proc. Advances in Multimedia Information Processing, PCM 2001, LNCS 2195, 498–505

[21] Vendrig, J. and Worring, M. Systematic evaluation of logical story unit segmentation. Multimedia. *In IEEE Transactions on. 4, 4* IEEE (2002), 492–499.

[22] Shi, J. and Malik, J. Normalized Cuts and Image Segmentation. *In IEEE Transactions on Pattern Analysis and Machine Intelligence. 22, 8* IEEE (2000), 888–905.

[23] Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H. and Trancoso, I. Multi-modal scene segmentation using scene transition graphs. *In Proc. MM '09 of the 17th ACM international conference on Multimedia.*ACM(2009), 665-668.

[24] Xu, S., Feng, B., Ding, P. and Xu, B. 2012. Graph-based multi-modal scene detection for movie and teleplay. *In Proc. Acoustics, Speech and Signal Processing (ICASSP) 2012.* IEEE (2012), 1413–1416.

[25] Porteous, J., Benini, S., Canini, L., Charles, F., Cavazza, M. and Leonardi, R. 2010. Interactive storytelling via video content recombination. *In Proc. MM '10 of the 17th ACM international conference on Multimedia.* ACM (2010), 1715–1718.