International Journal of Multimedia Information Retrieval manuscript No.
(will be inserted by the editor)

The Video Browser Showdown: A Live Evaluation of Interactive Video Search Tools

Klaus Schoeffmann · David Ahlström · Werner Bailer · Claudiu Cobârzan · Frank Hopfgartner · Kevin McGuinness · Cathal Gurrin · Christian Frisson · Duy-Dinh Le · Manfred Del Fabro · Hongliang Bai · Wolfgang Weiss

Authors version of manuscript published by Springer: `http://link.springer.com/article/10.1007/s13735-013-0050-8`

Abstract The Video Browser Showdown evaluates the performance of exploratory video search tools on a common data set in a common environment and in presence of the audience. The main goal of this competition is to enable researchers in the field of interactive video search to directly compare their tools at work. In this paper we present results from the second Video Browser Showdown (VBS2013) and describe and evaluate the tools of all participating teams in detail. The evaluation results give insights on how exploratory video search tools are used and how they perform in direct comparison. Moreover, we compare the achieved performance to results from another user study where 16 participants employed a standard video player to complete the same tasks as performed in VBS2013. This comparison shows that the sophisticated tools enable better performance in general but for some tasks common video players provide similar performance and could even outperform the expert tools. Our results highlight the need for further improvement of professional tools for interactive search in videos.

Keywords Video Browsing · Video Search · Video Retrieval · Exploratory Search

1 Introduction

Video browsing is the interactive process of exploring video content in order to find particular segments or to learn about the content structure. A typical video browsing tool – although a very simple one – is a common video player that provides navigation facilities for changing the playback position in a video (e.g., with a seeker-bar). Many browsing tools, with better interaction means than provided by a typical video player, have been presented in the literature (for a detailed review see [29]). While many of them are advanced navigation methods (e.g., [7,8]), extended video players (e.g., [10,14,18]) or enhanced video content visualizations [6], some are highly sophisticated browsing tools (e.g., [1,25,30]). These sophisticated tools provide very specific interfaces and advanced interaction methods, such as combined mouse/keyboard interaction for 3D navigation, table-of-content navigation in videos, navigation trees and spatial interaction (e.g., [12,13,22,23,26,28]). Interaction design for multimedia information retrieval is still a challenge, as raised in [21].

Klaus Schoeffmann
Alpen-Adria-Universität Klagenfurt, Austria

David Ahlström
Alpen-Adria-Universität Klagenfurt, Austria

Werner Bailer
JOANNEUM RESEARCH, Graz, Austria

Claudiu Cobârzan
Alpen-Adria-Universität Klagenfurt, Austria

Frank Hopfgartner
Technische Universität Berlin, Germany

Kevin McGuinness
Dublin City University, Ireland

Cathal Gurrin
Dublin City University, Ireland

Christian Frisson
Université de Mons, Belgium

Duy-Dinh Le
National Institute of Informatics, Japan

Manfred Del Fabro
Alpen-Adria-Universität Klagenfurt, Austria

Hongliang Bai
Orange Labs International Centers, China

Wolfgang Weiss
JOANNEUM RESEARCH, Graz, Austria

Although many tools have been proposed for interactive video search (see [29]), it is not obvious how well these tools perform for specific search tasks and for specific user groups such as expert and novice users [32,37]. It is also non-trivial to directly compare the tools without a common user study. While for video retrieval a fair comparison can be achieved by providing a common data set and common queries, this is hardly possible for video browsing tools due to their highly interactive nature. In order to directly compare them, user studies with exactly the same setup (i.e., same queries, data set, users, time limit, environment etc.) have to be performed. One step in this direction is the Interactive Known-Item-Search (KIS) task of TRECVID [33, 24], which ran over the last three years. The KIS scenario simulates the situation where someone knows of a video scene, and knows it is contained in the data set, but does not know where and how to find it. However, TRECVID KIS is rather focused on video retrieval than on interactive video browsing. Thus, the Interactive KIS tools can serve their purpose well relying on less interactive features, such as text query input.

The Video Browser Showdown (VBS) is a live competition (i.e., a live and highly active demo session at a conference) to evaluate interactive video browsing tools that target exploratory search scenarios rather than the typical automatic video retrieval approach. The aim is to evaluate video browsing tools for their efficiency at KIS tasks with a well-defined data set and setup in direct comparison to other tools. It is important to note that the VBS shares some similarities with the VideOlympics [34] (such as running known-item queries in front of a live audience) but differs in terms of rules and targeted methods/tools. While the VideOlympics mainly focuses on automatic video retrieval in video collections, the VBS targets highly interactive search tools that support users in exploiting their knowledge about the content for faster visual search. Moreover, VBS' main focus is content search in single video files rather than searching content in larger video collections. Also, participants are not allowed to perform any speech recognition or optical character recognition (OCR) that can be used with a textual query. It acts as a small and live 'user study' with both domain experts and novice users. Although participants know the data set before the competition and can use it for content analysis, they do not know the queries beforehand. Instead of using textual queries only (as e.g. in the TRECVID KIS task), target segments are presented as short video clips on a shared screen and the participants see them for the first time in the competition.

The VBS simulates search problems that are similar to practical situations where users can rely on prior knowledge to retrieve desired segments in videos. For example, if the query video shows a weather forecast scene of a news video, the user may immediately conclude that the segment is located at the end of a news video and focus her search on that part of the video file. In order to allow such search behavior, it is important that the underlying video search tool provides appropriate interaction means that can effectively support the searcher in using her knowledge to find the needed content as quickly as possible.

In this paper we first describe the tools that participated in the last Video Browser Showdown competition (VBS2013) and report on the achieved performance. We also compare the efficiency of these tools to the efficiency that can be achieved with a common video player. For that purpose, in addition to the Video Browser Showdown we have performed a baseline study with 16 participants, where exactly the same search tasks (same video queries and same data set) had to be performed with an HTML5 video player. Our results show that for most of the search tasks the professional search tools of VBS2013 were much faster. However, for a few tasks the participants of the baseline study could achieve a similar result as the VBS2013 tools and even outperform them. We discuss why a simple standard HTML5 video player, with only a small seeker-bar for content navigation, could outperform expert tools that use complex content analysis techniques like concept detection or face detection and give some ideas of how the expert tools could possibly achieve a better performance.

Accordingly, the contribution of this paper is twofold. It compares state-of-the-art video browsing tools with each other and with a simple video player. It provides detailed information on how fast users can interactively find content in videos and how far ahead state-of-the-art research tools are, when compared to classic content navigation, provided by the seeker-bar of a video player. The presented results show that novice users with very simple navigation tools, provided by a common video player, can easily keep up with the state-of-the-art tools in the VBS in terms of visual search time. Our findings highlight the need for more sophisticated interactive video search tools, which focus on user-based search instead of automatic video retrieval. This need is also stated in another recent article [38]. In order to enable other researchers to compare the performance of their exploratory video search tools to the one reported in this paper, we provide the ground-truth data used with VBS2013 on the website of the Video Browser Showdown[1].

[1] http://www.videobrowsershowdown.org/

The rest of the paper is structured as follows. In Section 2 we give more details about the VBS competition, including its setup, rules, and design decisions. Section 3 describes the video browsing tools that participated in VBS2013. The achieved performance of these tools – with both expert users and novice users – is analyzed in Section 4. The additionally performed baseline study, where users employed a simple video player for the same tasks, is described in Section 5 and the results of this study are compared to the results of VBS2013 in Section 6. Finally, Section 7 concludes this work.

2 Video Browser Showdown 2013

The first Video Browser Showdown was organized as a special session at the 18th International Conference on MultiMedia Modeling (MMM'12) in Klagenfurt, Austria. The first instalment of VBS can be seen as an entertainment event in the spirit of the VideOlympics. For VBS2013, which was organized a year later at MMM'13 in Huangshan, China, we included an activity logging module to perform a thorough scientific analysis of the performance of the different systems.

Six different video browsing tools participated in the competition that had to be used for 16 different search tasks. These tasks were separated into ten expert tasks and six novice tasks, because the VBS wants to foster video search tools that can be operated by non-experts – after a short introduction – as well. While in the former case the developers themselves performed the search, in the latter case six volunteers from the audience, one for each video search tool, were randomly selected to perform the search. The six volunteers were PhD students or researchers in the field of multimedia. Accordingly, all volunteers had technical and theoretical multimedia-related knowledge but were not necessarily video search experts. Before starting the novice competition all volunteers got a 10 minutes long introduction about the assigned search tool from the corresponding developer. A task started with a 20 seconds query video presented on a shared screen. This 20 seconds excerpt does not necessarily start and stop at shot or scene boundaries. The reason for using a rather long and non-aligned query segment is due to the fact that the VBS wants to simulate real-life situations, where users remember several related content for a target segment. Participants were given a maximum time limit of three minutes to find the target sequence in the corresponding video file. This time limit should motivate smart search behavior instead of simple fast-forward approaches. Before presenting the query video, the moderator mentioned the name of the video file to search in.

A common data set of ten video files was used for VBS2013, which was provided to the participants one month before the competition. This was required to enable the participants to perform content analysis on the video files. During the competition the organizers of the VBS2013 randomly selected the 20 second segments that formed the queries for the search task. VBS2013 started with the Experts Run, where ten search tasks were performed. The Experts Run was followed by the Novice Run, were another six randomly selected target segments were used as search tasks. Figure 1 shows the duration of the videos used for the experts run as well as the location of the target segments. Figure 2 shows example content uniformly sampled from the videos used for the ten expert tasks. As can be seen in the figure, the content was quite diverse because all videos were roughly one hour long and contained Flemish (Dutch language) news content. The reason for using only about ten hours of video content for the VBS is simply the fact that interactive search in larger archives would take too long for a live competition.

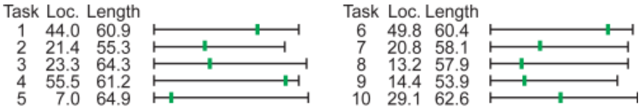


Fig. 1 Location of the center of the target segment (green) in each of the ten videos used for the expert tasks (location and video length in minutes).

We note that participants could have learned the content of the video files in order to benefit from that knowledge at the competition. However, from the performance reported in Section 4 we conclude that nobody actually did that. The participating teams were free to use and implement content-based analysis in their tools, but were not allowed to use automatic speech recognition, OCR, or manual annotation techniques because the VBS focuses on interactive video search rather than pure text search. However, text search and filtering based on automatically extracted metadata from content analysis without automatic speech recognition and OCR (e.g., semantic concepts) were allowed.

Figures 3 and 4 show the setup of the VBS session at MMM2013: the systems of all participating teams were organized in a U-shape arrangement in front of the moderator and the shared screen, which was used for presenting the query videos and the current state of all teams via the VBS server. This HTTP-like communication server was connected to all systems over a dedicated Ethernet switch and computed the performance scores for each tool and each task accordingly (see Figure 4). Each tool provided a submission feature

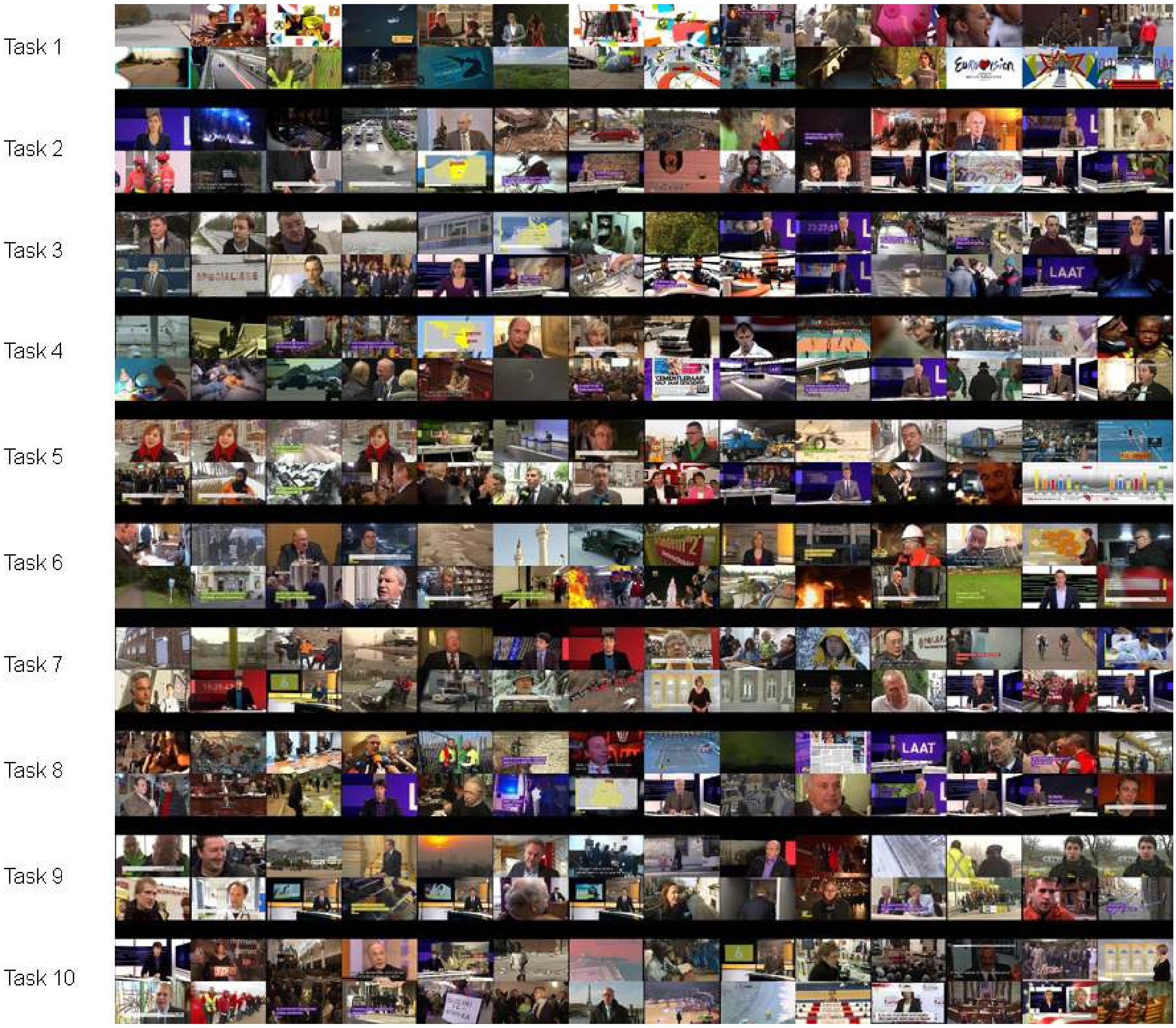


Fig. 2 Example frames of the videos used for the ten expert taks (uniform sampling with a distance of 112 seconds).

that could be used by the participant to send the current position in the video (i.e., the frame number or segment) to the server. The server checked the submitted frame number for correctness and computed a score for the corresponding tool and task based on the submission time and the number of false submissions. The following formulas were used to compute the score s_i^k for tool k and task i, where m_i^k is the number of submissions by tool k for task i and p_i^k is the penalty due to wrong submissions. The overall score S^k for tool k is simply the sum of the scores of the ten expert tasks and the six novice tasks (see Equation 3).

Equations 1 and 2 were designed such that participants submitting several wrong results get significantly less points than participants submitting just one correct result. This should avoid trial-and-error approaches. Additionally, the linear decrease of the score over time should motivate the teams to find the target sequence as fast as possible.

$$s_i^k = \frac{100 - 50\frac{t}{T_{max}}}{p_i^k} \tag{1}$$

$$p_i^k = \begin{cases} 1, & \text{if } m_i^k \leq 1 \\ m_i^k - 1, & \text{otherwise} \end{cases} \tag{2}$$

$$S^k = \sum_{i=1}^{16} s_i^k \tag{3}$$

The hardware for the competition was not normalized; all participating teams were free to use the equipment best supporting the requirements and efficiency of

Fig. 3 A picture of the VBS2013 participants in action.

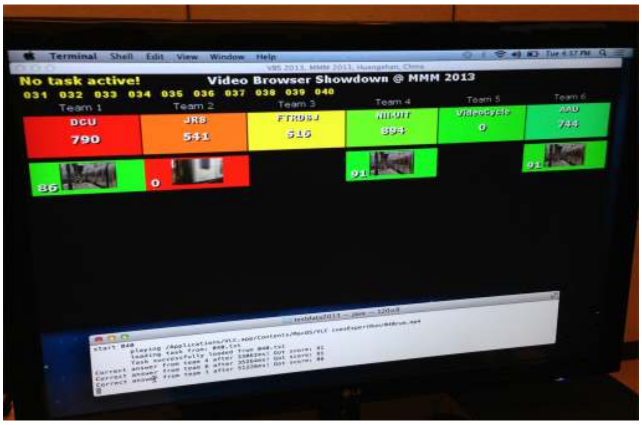


Fig. 4 In addition to presenting the target scene at the beginning of a search task, the shared screen is used to show the live output of the VBS-Server, which presents the current score for each tool. One column is dedicated to one specific tool and shows the overall score at the top and all submissions below in descending order (last submission above). Correct submissions are presented in green color and wrong submissions are presented in red.

their video browsers. All teams used high-end notebook computers that were connected to external 21-inch displays provided by the conference organizers. It was even allowed to use specialized hardware if desired (only one team, the VideoCycle team, used specialized hardware: a jog wheel as shown in Figure 9).

We note that the design of the Video Browser Showdown, as described in this section, has several limitations (e.g., a very limited number of tasks, too few users per video browsing tool, the used data set was limited to news content, and a non-controlled task environment within which various unknown factors could influence participants search performance). These limitations prevent us to draw any exact and definitive conclusions regarding the performance of the participating video browsing tools. Instead, we regard the presented results as strong guiding performance indications. To achieve more profound and representative performance results – which was not the primarily objective of the Video Browser Showdown – we would need to conduct a dedicated and controlled user study with more participants, tasks, and sample videos.

3 Participating Video Browsing Tools

The following subsections briefly describe each of the six video browsing tools that participated in VBS2013.

3.1 DCU

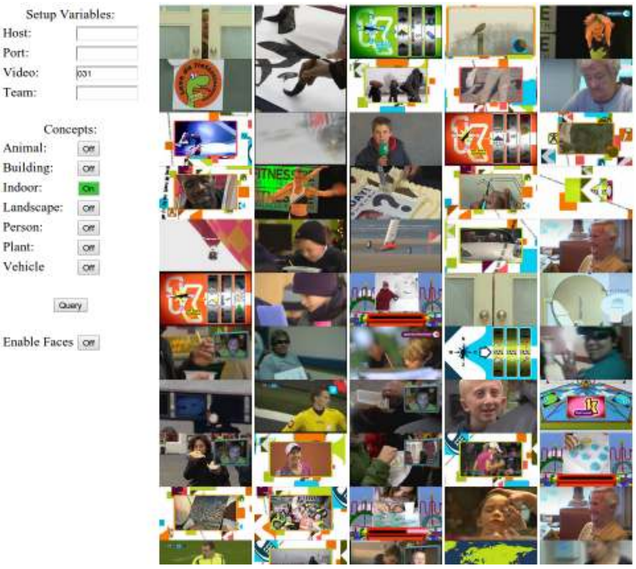
The DCU tool [31] won the novice user run and ended up as runner-up of the whole competition. As can be seen in the screenshot shown in Figure 5, the tool provides a quick overview over different scenes within a video by displaying key frames. Each key frame represents a group of visually similar shots, thus reducing the cognitive work load associated with scanning through a large number of heterogeneous key frames. This grouping of these frames is achieved by performing agglomerative clustering on the frames' VLAD descriptors [17] that are computed by performing k-means clustering (with k = 64) on the entire set of sparse SIFT [20] descriptors extracted for each video. PCA is used to reduce the dimension of the VLAD descriptors to 128 dimensions.

On the left hand side of the interface, the user can apply different concept filters, such as *animals, buildings, faces,* or *plants*, to narrow down the search. These semantic concepts are detected by training Support Vector Machines on the Bag-of-Visual-Word feature representation. In addition, faces are detected using the approach of Viola and Jones [36]. The filter that was most commonly used during the VBS was the face detector filter.

After clicking on one of the key frames in the result list, a new page opens where the video can be viewed and similar shots are displayed. Since focusing on VLAD descriptors allowed for the representation of all key frames of a video in less than 50MB of memory, similarity search is performed by determining the query's VLAD vector and exhaustively computing the distance to all other key frames of the video rather than using approximate nearest neighbors [16].

3.2 JRS

The JRS tool used a video browsing application targeted at content management in post-production of film



Setup Variables:
Host:
Port:
Video: 031
Team:

Concepts:
Animal: Off
Building: Off
Indoor: On
Landscape: Off
Person: Off
Plant: Off
Vehicle Off

Enable Faces Off

Fig. 5 Screenshot of the DCU video browser. The left panel allows the user to select various concept detectors. Ranked shots from fusing the chosen concept detectors are displayed on the right. The face filter suppresses all shots not containing faces.

and TV [4]. It works based on automatic content analysis that performs camera motion estimation, visual activity estimation, extraction of global color features and estimation of object trajectories. The central component of the tool's user interface is a light table that shows the current content set and cluster structure using a number of key frames for each of the clusters (see Figure 6). The user applies an iterative content selection process which allows for the clustering of content or searching for similar key frames based on the extracted features. After that, the relevant key frames can be selected in order to reduce the content set. A history function records all user search and cluster actions and allows the user to jump back to a previous point.

3.3 FTRDBJ

The FTRDBJ tool allows video browsing by a combination of Semantic Indexing (SIN) and Instance Search (INS). The method of interaction allows users to index the target clip via their knowledge of the video content. The system offers users a set of concepts and the SIN module returns candidate key frames based on users' selection of concepts. Users can choose key frames which contain the interest items, and the INS module recommends some similar key frames related to the target clip. Finally, the precise time stamps of the clip are given by the temporal refinement.

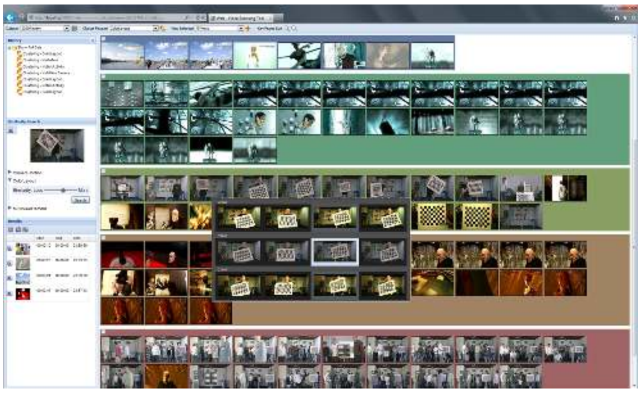


Fig. 6 Screenshot of the JRS video browser. The main component is the light table view on the right, indicating clusters by the background color. The controls on the left are the browsing history, similarity search and the result list.

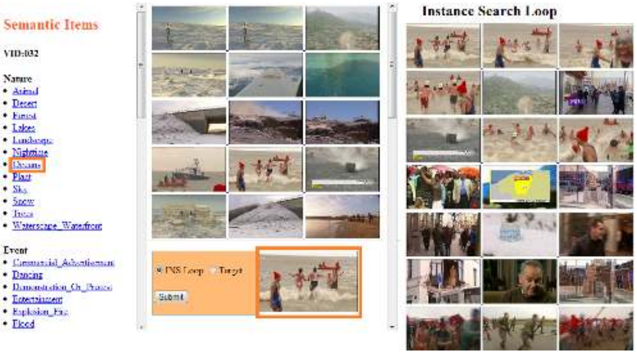


Fig. 7 Screenshot of FTRDBJ video browser. The left column shows the selected concepts related to the target video. The SIN recognition results are listed in the middle column.

Details of SIN algorithm can be found in the corresponding proceedings of the TRECVID Semantic Index task [35]. In the SIN module, 59-concept models are pre-learned through a composite-kernel Support Vector Machine (SVM) and training data of TRECVID SIN 2011 and 2012. The 59 concepts are shown on the left column of Figure 7.

3.4 NII-UIT

The NII-UIT-VBS tool [19] shown in Figure 8 won the first prize of the whole competition. To quickly find the target clip, the key idea is to smartly select a small number of candidate segments for further investigation by the user. To this end, a concept-based filter and color distribution-based filter are used.

The concept-based filter uses classifiers trained in advance with specific concepts such as *Face*, *Indoor*, *Outdoor*, *Daytime*, *Nighttime*, *Animation*, *Entertainment*, and *Sports* to annotate representative key frames of candidate segments. The concepts are selected with high abstraction level so that the user can easily recognize it from the target clip. For example, if the target clip is about people talking in an interview, picking candidate segments annotated with face-concept can help to focus on potential clips. For training classifiers, global features such as color moments and local binary patterns with RBF-kernel and LibSVM are used. The annotations from TRECVID Semantic Indexing Benchmark [24] are used. The Viola and Jones face detector implemented in OpenCV is used for detecting faces and to annotate the *Face* concept.

The color distribution-based filter is used to select candidate segments having same color distribution with the target clip at some location. For example, if the target clip is related to news studio, some parts located at the bottom of key frames will likely have the same color. One key frame is selected for each short segment. A 4×4 grid is used to divide each key frame into 16 regions. For each region, a HSV color histogram is built. Given a target color in the predefined palette, a similarity measure (Euclidean distance) is calculated between each of histogram and the histogram of the target color. Only top 20 regions with the highest similarity corresponding with different key frames are returned.

Since it is not easy for general users to select suitable concepts or colors in the filters by watching the target clip in a short time, another simple but efficient approach to select candidate segments is used. It is based on coarse-to-fine strategy. At the coarsest level, one key frame is used to represent the content of each one minute-segment (called super segment). At the finest level, five key frames are used to represent content of each 10-second segments. Since the number of super segments in one hour video is 60, skimming these super segments helps to identify potential locations. When the user clicks on the representative key frame of each super segment, up to four 10-second segments can be added to the list of candidate segments. When the user wants to look at more details of one candidate segment, five representative key frames are shown to help the user confirm whether the target clip contains that segment. The two segments adjacent with the previewed segment are also shown in order to help the user more information before going to the final decision.

The user interface shown in Figure 8 is optimally designed so as to reduce unnecessary navigations. On the top screen are the two filters and the video player. The super-segment panel shows key frames representing for super-segments. The candidate-segment panel shows key frames representing for candidate segments (i.e., 10-second segment) that are selected by either using filters or skimming super segments. The preview-segment panel shows five key frames of each candidate segment and two adjacent segments. The leftmost panel shows all 10-second segments randomly. It is used to add more candidate segments into the list.

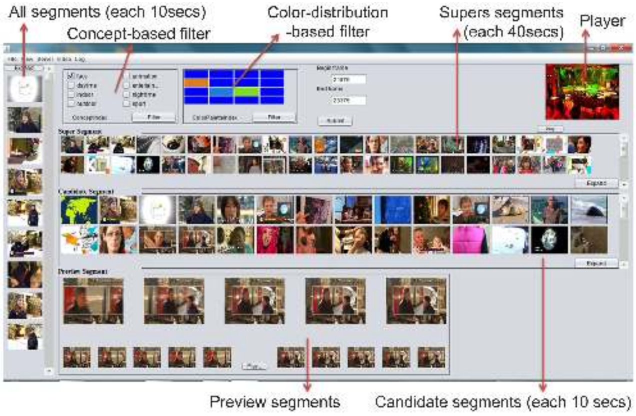

All segments (each 10secs)
Concept-based filter
Color-distribution -based filter
Supers segments (each 40secs)
Player
Preview segments
Candidate segments (each 10 secs)

Fig. 8 Screenshot of the NII-UIT-VBS Tool.

From the log files of the tool it is obvious that the two filters help a lot in quickly finding candidate segments. Especially the *Face* concept is useful when the number of target clips related to people is high. In addition, by using a coarse-to-fine approach and organizing candidate segments in temporal order, it helps the user easier to confirm whether a candidate segment is overlapped to the target clip. For example, when the target clip has people appearance, the user can use the *face* concept filter to select candidate segments. If only one key frame is used, it takes more time to confirm if using the video player to play the segment associated with that key frame. By using five key frames per candidate segments, and adjacent segments in the preview-segment panel, the time is reduced.

3.5 VideoCycle

The version of VideoCycle that took part in the VBS challenge is a subset of the whole framework described in [11]. As illustrated in Figure 9, it had been restricted to a video timeline featuring, from bottom to top:

– Bottom: a summary row of key frames evenly-distributed over the length of the video along the bottom border of the application window.
– In between: a selection row with key frames corresponding to the duration range determined by the span of the selection centered around the playback cursor over the aforementioned summary row.
– Top: a large playback view of the video.

In addition to the usual control offered by a mouse or trackpad, the span of the selection could be modified by the inner wheel of a jog wheel and skipping frames by its outer wheel, while the segment boundaries could be submitted to the VBS server by pressing a button of the

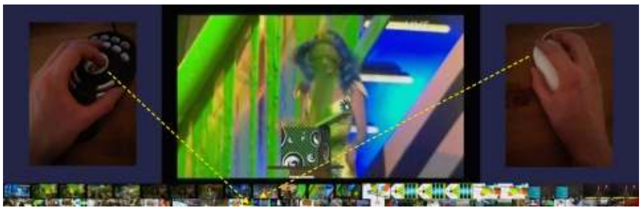


Fig. 9 Screenshot of the VideoCycle timeline with a large video playback (top), summary key frames (bottom), selection key frames (in between), and pictures of controllers: jog wheel for scrolling and playback speed adjustment, and mouse for frame skipping (left/right inserts).

jog wheel (see left insert in Figure 9). Key frames from the selection row could be clicked to reposition the playback cursor to the corresponding frame. By assigning the jog wheel to the left hand and the mouse to the right hand (or their permutation depending on the preferred hand), the user can keep a better motor memory of the controllers (as opposed to keyboard shortcuts), thus reducing the head movements that would have been required to look periodically at the controllers during the search, and focus the attention on the screen.

No user command logging was recorded during the tasks. But the participant used a repeatable browsing technique for each task, comparable to exhaustive search: linearly skimming through the video using the cursor while looking at the selection key frames row, increasing its span when a matching scene was discovered, if not skimming the video again. Provided the score calculation formula explained in Section 2, the participant did not try to set the time boundaries of the segment carefully before submitting it to the server, but rather made sure that the current playback time would be in between as the system would submit this value for both boundaries.

Without the browser of segments clustered by content-based similarity normally positioned above the timeline in the full version of VideoCycle, this setup sets itself as an intermediary baseline: it provides a more advanced visualization and improved interaction techniques over the standard video player used as baseline for the additional tests described in Section 5.

3.6 AAU

The AAU video browsing tool [9] applies content-based analysis to automatically detect repeating segments within videos. Repeating or similar segments are clustered based on their color layouts and motion patterns. Each cluster is annotated with a different color on the navigation bars of the video player.

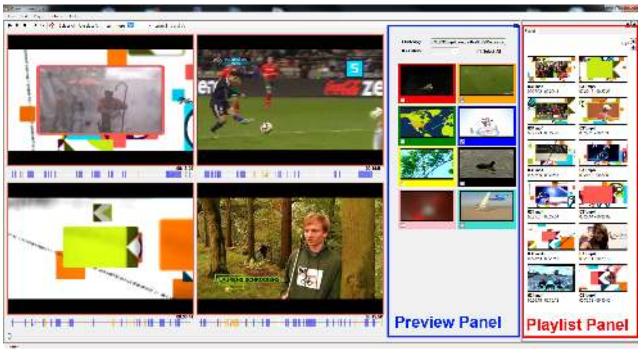


Fig. 10 Screenshot of the AAU video browser. The four windows at the left side can be used to browse four parts of a video in parallel. Next to them the preview panel is displayed. At the right side the playlist view is showing segments that belong to one cluster.

Figure 10 shows an example where a video is divided into four parts of equal length, which can be watched in parallel. The augmented navigation bars below the video windows have to be used to search for a certain segment. A meta-slider, which is placed at the bottom of the screen, can be used to browse all displayed parts in parallel. Users experience videos in the same way as usual, but the amount of time needed for certain search tasks can be reduced by taking advantage of the additional information provided by augmented navigation bars.

No predefined classes are used for the clustering. A latent indexing of the content is performed. The classification of the emerging clusters is the part where the user comes into the loop. A preview panel shows representative frames of the clusters. Each of these frames is surrounded by a border, colored with the color that is used for marking all segments of the corresponding cluster on the navigation bars. The preview panel can be used to quickly make a basic discrimination of the content of a video. If the frames shown in the preview panel are not diverse enough, all segments that belong to a cluster can be displayed in a playlist by clicking on the representative frame. The elements of a playlist are ordered chronologically, thus scanning the items of a playlist from the beginning to the end is also an option to search for a certain video segment.

An analysis of the log files shows that the expert made heavy use of the advanced features, while the novice user mainly relied on searching several parts of the videos in parallel using the navigation bar. The expert relied for four tasks only on the playlist to search the given segment, three times parallel scrolling without augmented navigation bars was used and in another three cases the augmented navigation bars were used. The novice only relied once on the augmented navigation bars and used the parallel scrolling (without augmented navigation bars) for the other five tasks. The playlist was never used. Comparing the mere time the navigation bars were used, it can be seen that the expert used the navigation bars for 16.7 seconds on average, while the novice only used them for 13.1 seconds on average. The expert found 9 out of 10 items, making four wrong submissions. The novice found all of the six searched segments, making only one wrong submission.

4 VBS2013 Evaluation

4.1 Team Performance

When looking at the overall score of all teams that participated in the VBS2013 (Figure 11), we can see that it was a very tight race between the NII-UIT team, the DCU team, and the AAU team. Based on that observation it is quite interesting that the first two teams were the only ones that used face detectors for content filtering (see Section 3.1 and 3.4), which was obviously advantageous as the common data set contained news content. Another interesting fact is the quite high performance of the AAU team, which used a rather simple but highly interactive video browsing tool.

While NII-UIT achieved a better scoring in the Expert Run and finally won the competition, DCU was slightly better in the Novice Run. However, from Figure 12, which shows the box plot for the submission time of correct submissions per team, we can see the reason for the good performance of NII-UIT. This team was much more steady in terms of search time and submitted all found segments within 54 seconds in the Expert Run (DCU 98 seconds, AAU 61 seconds) and within 63 seconds in the Novice Run (DCU 58 seconds, AAU 59 seconds). The box plot also shows that for the Novice Run the IQR of NII-UIT is much smaller than the IQR of DCU, which means the search times of NII-UIT were consistently low.

Figure 11 shows that JRS and FTRDBJ achieved almost the same score in the Expert Run but JRS was significantly better in the Novice Run. Figures 12 and 13, which show the average number of correct and wrong submissions per team, reveal why JRS was finally better in scoring than FTRDBJ.We can see that JRS submitted more wrong submissions in the Expert Run – and significantly less wrong submissions in the Novice Run – but was faster at submitting the correct segments for both runs (Figure 12). From that result we can conclude that the JRS tool was more intuitive to use by non-experts than the FTRDBJ tool.

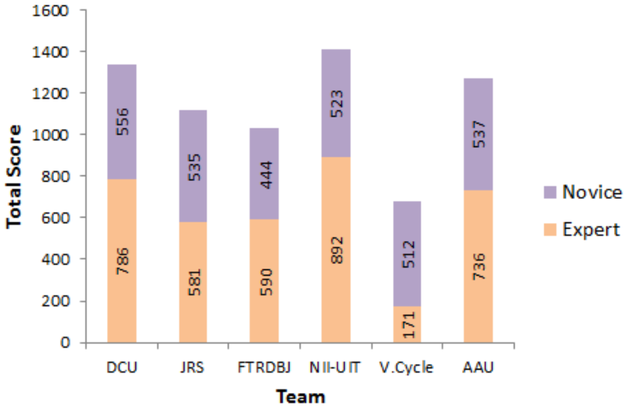


Fig. 11 Total score of teams in the VBS2013 (based on Equation 3).

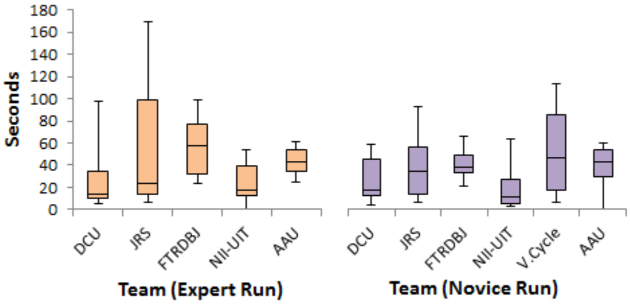


Fig. 12 Box plot of the submission time per team in the VBS2013, based on correct submissions.

The reason for the minimum search time of zero, as achieved by NII-UIT in Expert Run and by AAU in Novice Run (see Figure 12), is related to the configuration of the VBS session. Teams have to setup their systems before the target video is presented in order to enable a fair competition. As most systems provide an initial view after loading a video, it can happen that accidentally the target segment is already contained in the initial view and can be immediately submitted after the presentation of the query video has ended (the VBS server does not accept submissions during presentation of the query video).

It should be noted that the particularly low scoring of VideoCycle was due to some submission problems that were experienced during the Expert Run. Because of this problem, only two correct submissions were received from the VideoCycle team making a serious analysis meaningless. Therefore, as the comparative study in Section 5 concentrates on the experts run only, we have omitted the VideoCycle team there.

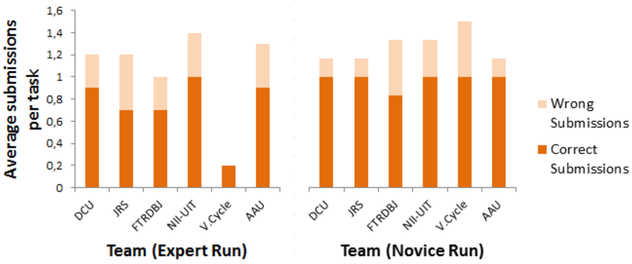


Fig. 13 Average number of submissions (correct and wrong) in the VBS2013 per team for both the Expert Run and the Novice Run.

4.2 Overall Performance

Out of the 109 submissions made in the Expert and Novice tasks, 30 (27.52%) were incorrect (19 in Expert Run and 11 in Novice Run). While these submissions are quite clearly off the target on the video timeline (mean 120 seconds, median 48 seconds in Expert Run; mean 142 seconds, median 32 seconds in Novice Run), a more in-depth analysis shows that they are not completely wrong. In terms of visual similarity, 64.8% of the false submissions are globally similar to the target clip, 70.2% share the background, and 51.3% show similar objects. These submissions would be relevant in response to a “find content like this” query, e.g., for stock footage. For the problem of finding material on a specific topic, even 83.8% of the false positives could be relevant, as they are from the same scene or news story. Section 6 discusses the false submissions of the Expert Run (Figure 16) in more detail.

Compared to VBS2012 [3], the number of false submissions was lower (48.8% in 2012), and both experts and novices were more successful in resubmitting. While in 2012 only 7 out of 88 correct submissions were made after a prior false submission (8.0%), this year 28 out of 79 submissions (35.44%) were at least a second try.

5 Baseline Study

In order to provide a more objective view of the performance of the tools used in the VBS2013, a few weeks after the Video Browser Showdown we performed a baseline study with 16 users employing a simple video player instead of a specialized video search tool. The study participants used the simple video player defined by the HTML5 video tag in a Safari web browser on Mac OS X 10.8. The provided interaction possibilities were thus limited to a play/pause button, a button for full-screen display, and a seeker-bar for navigation, as depicted in Figure 14e (the application used in the study was developed with HTML5 and JavaScript).

Exactly the same video data and the same target segments were used as in the VBS2013 competition. All the target segments were tested in the same sequence for all participants, as in VBS2013. When starting a trial, the participant was presented with an automatic playback of the target scene on the left side of a full-screen sized window, as shown in Figure 14a. During the playback no interaction was allowed and all interaction elements were removed from the video player (see close-up in Figure 14b). After the playback was finished the corresponding full video file was presented together with a count-down timer set to 3 minutes on the right side of the window, as shown in Figure 14c. The participant could search the target scene using only the default controls (start/pause, seeker-bar), Figures 14d and 14e, and used the submit button below the video player to check the currently shown frame against the corresponding target video. The window background turned red for 4 seconds to signal a false submission. If the submitted frame was contained within the target segment the window background turned green and the achieved score for the trial was presented for 10 seconds. The score was computed using Equation 1 and 2. The next test trial in the sequence was started by pressing a button that appeared in the window only after a successful submission or after the count-down reached zero before the target segment was found.

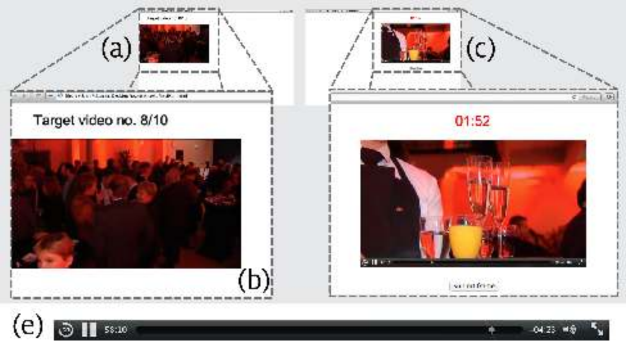


Fig. 14 The baseline study used a common video player interface. (a) and (b) the interface during the first stage of a trial with the automatic playback of the target scene. (c) and (d) second stage of a trial during search. (e) close up of the provided interaction possibilities provided by the video player.

Sixteen daily computer users (two female) aged 23 to 52 years (mean=31.6, $SD = 8.75$) participated in the study. The study application run locally on a MacBook Pro laptop with its 17-inch monitor set at a 1920×1200 pixels resolution. The interface was presented in a Safari web-browser window in full-screen mode. An optical wired mouse was used as input device.

Each participant performed ten timed trials, one attempt for each of the ten target segments used in the Expert Run of VBS2013, and two initial practice trials searching for other target segments from two additional one-hour videos. With initial instructions, practice trials, short breaks between trials, and ten timed trials, participation lasted approximately 40 minutes. More information about this study – including additional evaluation on navigation behavior – can be found in [27].

6 Baseline vs. VBS2013 Experts

For the baseline study conducted in addition to VBS2013 we present results regarding the score, submission accuracy and submission time in turn and order. We compare these results to the results obtained by the experts in the Expert Run of VBS2013. However, we first note some important differences between the baseline study and the VBS2013 that warrant caution when drawing conclusions regarding direct comparisons of the results. First, the baseline participants were not tool experts as VBS2013 participants were (developers themselves or at least multimedia experts). In contrast, the participants of the baseline study were master students from different disciplines (only a small minority studying Computer Science) but self-appointed daily computer users. However, we may assume that almost every daily computer user can quite effectively use a video player. Second, the baseline participants conducted the test in a self-paced series of trials sitting alone in a quiet room whereas the VBS2013 competition took place in a shared room at a conference and in front of audience and with a moderator that determined the pace. Third, the low number of tasks and the fact that the video browsers in VBS2013 were each only used by one single person (disregarding the participants in the Novice Run) do not allow us to use any inferential statistical methods. Thus, we limit ourselves to descriptive statistics.

6.1 Score

The box plot in Figure 15 shows the scores for each participant in both studies. The mean score, across participants and tasks, in the baseline study was very similar to the mean score in the VBS2013: 74.8 points for the baseline and 71.7 points in the VBS2013 (SD=27.8 resp. 35.0).

On average, participants in the baseline study got a score higher than 0 in 96.7% of the trials whereas the average participants in VBS2013 only scored in 84.0% of the trials. In the baseline study, 15 trials (10%) were registered with a score of zero. These failed trials were distributed between Task 1, 2, 3, 4, 5, and 10 with a frequency of 5, 2, 1, 3, 3, and 1, respectively. The eight zero-score trials in VBS2013 were distributed between Task 1, 7, 8, and 10 with a frequency of 4, 1, 1, and 2, respectively. Across both studies, only Task 6 and 9 were always successfully completed. Notable is the high number of trials with Task 1 that ended in a zero-score (5 out of 16 in the baseline study and 4 out of 5 in VBS2013). A manual investigation of the corresponding video revealed that the reason for that is quite simple. The randomly selected target segment showed a “teaser” that is contained several times in slightly different variation throughout the video. Therefore, we conclude that the participants had a hard time in finding the proper one and most of them did not succeed within the time limit.

The box plot in Figure 15 summarizes the scores for each of the 16 baseline participants and the five VBS2013 participants. In the baseline study, five participants were unable to find the target segment within 180 seconds in one of the ten tasks and five participants failed in two tasks. Six participants found the target segment within time in all ten tasks and thus received a score greater than zero in all tasks. In VBS2013, only one participant (NII-UIT) scored in all tasks. Two participants (DCU and AAU) scored in all but one task, and two participants failed in three tasks (JRS and FTRDBJ).

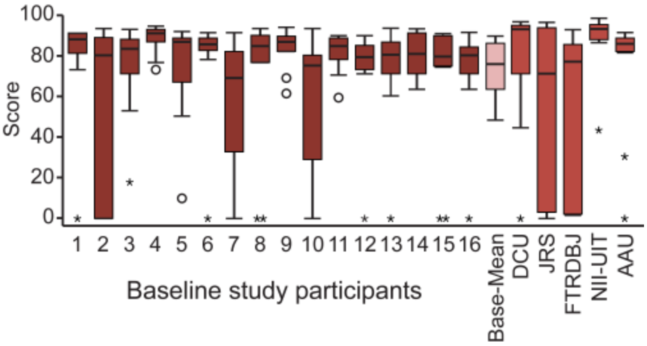


Fig. 15 Box plot of baseline and VBS2013 participants’ scores (∘: > 1.5×IQR, ⋆: > 3×IQR).

6.2 Submission accuracy

In the baseline study, a total of 50 false submissions were registered, for an average of 0.31 false submissions per trial. The VBS2013 participants made a total of 19 false submission, for an average of 0.38 false submissions per trial. However, as shown in Figure 16, the false submissions were unevenly distributed between the ten tasks. Particularly so in the baseline study where 56.8% of all false submissions were registered in Task 5. At most eight false submissions were registered in a single task by a single participant (Task 5). Among the VBS2013 participants, at most three false submissions were made in a single task (by two different participants in Task 5 and Task 8, respectively).

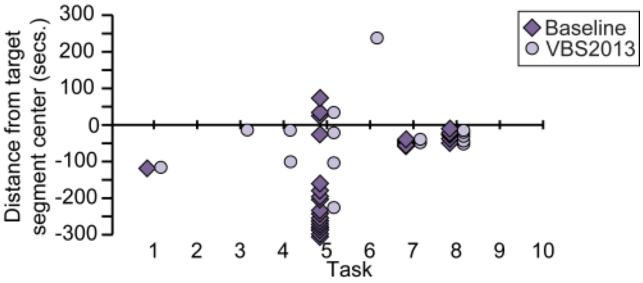


Fig. 16 Distance from target segment center for all false submissions.

Most false submissions were made for Task 5 and Task 8. The query video of Task 5 showed a cut-out from two different scenes with the second scene also appearing several times before the location of the target segment in a similar manner (different interviews done by the same news correspondent). Therefore, many participants of the baseline study, as well as some VBS2013 participants, made false submissions for these earlier scenes. The reason for the many false submissions for Task 8 was exactly the same: a visitor of a night event was interviewed and this several minutes long interview was cut into several segments, accompanied by additional recordings of the event in between. The target segment was rather at the end of the interview and many participants submitted earlier parts because they obviously could not understand Dutch.

On average, the baseline participants made 1.27 submissions (SD=0.28) in a successful trial whereas the VBS2013 participants made on average 1.44 (SD=0.21) submissions in a successful trial. In 16.3% of the successful trials (22 out of 135 trials) of the baseline study more than one submission was made. At most five false submissions were registered in one trial before the correct segment was submitted. The highest number of false submissions registered in a successful VBS2013 trial was four. In total, 26.2% of the successful VBS-trials (11 trials out of 42) more than one false submission was made.

The lower number of submissions per trial in the baseline study combined with baseline participants much higher success rate (96.7% vs. 84.0% in VBS2013), suggests that, in general, baseline participants adopted a more defensive tactic than the VBS2013 participants and were a bit more unwilling to submit unless being sure to have found the correct segment. We can also conclude that the average baseline participant, represented by the box labeled ‘Base-Mean’ in Figure 15, with the HTML5 video player could clearly not match the scores received by the three top-scoring teams (NII-UIT, DCU, and AAU) in VBS2013. However, we also point out that several of the baseline participants would have positioned themselves and the standard video player on respectable ranks if they had participated in VBS2013.

6.3 Submission time

We now turn to the submission time and focus on successful trials only. Figure 17 displays a box plot with the submission times for each baseline participant, the aggregated results of all baseline participants (‘Base-Mean’), and submission times for each VBS2013 participant. The baseline participants found the correct segment on average after 57.9 seconds (SD=36.5), VBS2013 participants after 40.5 seconds (SD=34.5). The fastest correct submission in the baseline study was registered after 13 seconds (participant 9), and the slowest was registered after 178 seconds (participant 5), just before the allowed 180 seconds had elapsed. In the VBS2013, opposite to the baseline study, participants were allowed to start searching for the target segment before the playback of the target segment had stopped, yet no submission could be made before the playback had ended. Consequently, some submissions in the VBS2013 data were considerably fast. In nine trials, successful submissions were faster than 13 seconds (between 0.2 and 12.6 seconds). The slowest successful submission in the VBS2013 data was registered after 169 seconds (JRS).

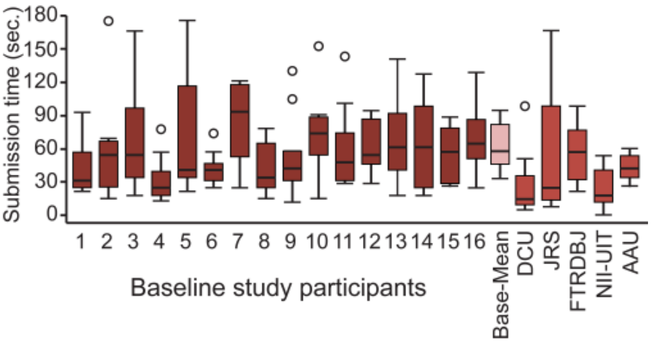


Fig. 17 Box plot of submission times for baseline and VBS2013 participants (○: > 1.5×IQR).

In terms of speed – as regarding the score – some of the best baseline participants could easily keep up with the VBS2013 expert results. However, the median time the average baseline participant needed in order to find the correct segment using the HTML5 video player is only comparable to the median time of the slowest VBS2013 participant (FTRDBJ) needed for a successful submission. All the other four VBS2013 browsers exhibited considerably faster median times.

Figure 18 plots the mean submission time for each successful trial in the baseline study (a) and in VBS2013 (b) against the target segment location in minutes from the start of the corresponding video (cf. Figure 1). In the left part of the figure showing the data from the baseline study, we clearly see a linear pattern for eight of the ten tasks (exceptions are Task 5 and Task 6) with final submission times linearly increasing with the distance of the target segment (cf. Figure 1) from the beginning of the video ($R^2 = 0.84, y = 22.627+1.397x$).

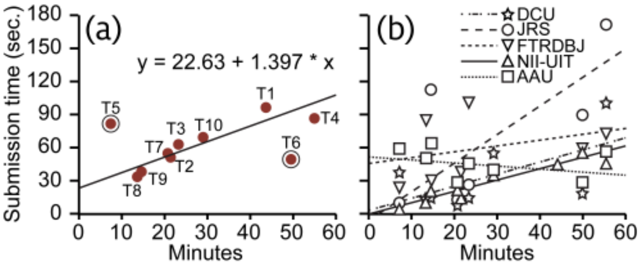


Fig. 18 (a) Mean submission time of correct submissions plotted against target segment location for each task. (a) baseline study, (b) VBS2013.

This result mirrors the search behavior we observed during the study. The majority of the participants in the baseline study searched the video in one direction – either from the beginning towards the end, or vice versa – with a certain granularity. They made this either by clicking on the seeker-bar using a more or less constant offset for each new click, or by smoothly sliding the knob of the seeker-bar towards the end. This sequential search behavior comes a no surprise given the limited manipulation and navigation opportunities provided by the seeker-bar. Furthermore, without any guiding cues about the location of the target segment within the query video, a systematic search approach ought to seem as the most advantageous one for the majority of users. However, in some cases, in particular with Task 6, some participants also seemed to employ knowledge about the general structure of news broadcasts and acted accordingly by concentrating their search on a certain part of the video. In Task 6, for example, where the target segment contained scenes from a soccer game, we saw several participants that immediately jumped past the first half of the video before starting a more fine-granular and systematic search going towards the end of the video. Supposedly they relied on their semantic knowledge and assumed the soccer scenes to be located in sports section at the end of the news broadcast. This explains the "outlier" location of Task 6 in Figure 18a. The irregularity of Task 5 in Figure 18a is explained by the problematic caused by many highly similar scenes throughout the query video, as previously discussed below Figure 16.

Finally, in contrast to the quite strong linear relationship we observe between search time and the time of the target segment for the simple video player used in the baseline study, we see in Figure 18b that the more elaborate mechanisms and navigation possibilities provided by the browsers used in VBS2013 do not bound the user to tedious sequential searches.

7 Conclusions

In this paper we have presented an overview of the Video Browser Showdown (VBS) competition, and have described the tools of last year in detail. We have analyzed the performance of these tools and compared it to the performance of a baseline study, in which only a simple video player has been used for solving the known item search tasks.

Overall, the scores achieved by the VBS2013 participants were rather high, showing that the proposed tools are able to solve these type of content search tasks. The number of false submissions was lower than in the evaluation in 2012, as well as the overall total number of submissions, indicating that the tools needed less "trial and error". Even more relevant considering practical content requests in media production, about two thirds of the false submissions have a high visual similarity, and more than 80% are from the same scene or news story.

At a first glance, the comparison of the baseline study with the VBS results seems to show, that a very simple player competes remarkably well with the quite sophisticated tools described in this paper. The scores of many participants are quite high, with a comparable rate of false submissions. However, two results help to discriminate rather exhaustive search in the baseline study with a structured approach of the VBS participants. First, the median submission time of the baseline study is only in the range of the slowest VBS team, i.e., VBS participants were able to find relevant segments faster. Second, there is a quite strong linear correlation between the time of the target segment in the video and the submission time, indicating linear search. The exceptions to this are cases where the baseline participants were able to infer the position of the segment based on the structure of the program.

Nevertheless, the performance of the video browsing tools used in the VBS is not completely satisfying when considering the fact that very simple navigation tools (i.e., video players) are only about 50% slower in terms of average search time (38.5 seconds in the VBS vs. 57.93 seconds in the baseline study). One reason could be that the state-of-the-art tools of the VBS are still too less focused on highly interactive use, such that users could more quickly and easily translate their knowledge and intentions to navigation and search features provided by the tool. It seems that video search tools should concentrate more on the user instead of mainly the presentation of results obtained from content-based analysis methods, as also stated in [38]. As our baseline study has shown that users tend to search in linear manner (see Figure 18a and more details in [27]), it is important to support chronological search behavior even in combination with content-based search features. Some tools of VBS2013 obviously support such a linear search already (e.g., the VideoCycle and the AAU tool). In addition, it could be beneficial to investigate alternative content visualization methods that allow users to see more content at a glance and hence to more quickly recognize desired content (like for example [2]). Also, the usage of visualizations that more clearly convey information about the content structure (e.g., [5,15,28]) could be very helpful.

8 Acknowledgements

This work was funded by the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit) and Austrian Science Fund (FWF): TRP 273-N15 and the European Regional Development Fund and the Carinthian Economic Promotion Fund (KWF), supported by Lakeside Labs GmbH, Klagenfurt, Austria. The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287532 entitled "TOSCA-MP".

Christian Frisson has been funded by the numediart research program in digital arts technologies, grant from the Walloon Region of Belgium, from 2010 to 2013. He would like to acknowledge his colleagues part of the VideoCycle team and doctoral jury, Thierry Dutoit (advisor) and Stéphane Dupont, that lead to the participation of the team to VBS 2013.

The content used in the experiments has been made available for the EBU MIM/SCAIE project by the Flemish public broadcasting organisation VRT (http://www.vrt.be/), Belgium.

References

1. Adams, B., Greenhill, S., Venkatesh, S.: Towards a video browser for the digital native. In: Proceedings of IEEE International Conference on Multimedia and Expo Workshops, pp. 127–132 (2012). DOI 10.1109/ICMEW.2012.29
2. Ahlström, D., Hudelist, M.A., Schoeffmann, K., Schaefer, G.: A user study on image browsing on touchscreens. In: Proceedings of the 20th ACM International Conference on Multimedia, MM '12, pp. 925–928. ACM, New York, NY, USA (2012). DOI 10.1145/2393347.2396348
3. Bailer, W., Schoeffmann, K., Ahlström, D., Weiss, W., del Fabro, M.: Interactive evaluation of video browsing tools. In: Proceedings of the Multimedia Modeling Conference, pp. 81–91 (2013)
4. Bailer, W., Weiss, W., Kienast, G., Thallinger, G., Haas, W.: A video browsing tool for content management in post-production. International Journal of Digital Multimedia Broadcasting (2010). DOI 10.1155/2010/856761
5. Barnes, C., Goldman, D.B., Shechtman, E., Finkelstein, A.: Video tapestries with continuous temporal zoom. In: ACM SIGGRAPH 2010 Papers, SIGGRAPH '10, pp. 89:1–89:9. ACM, New York, NY, USA (2010). DOI 10.1145/1833349.1778826
6. Borgo, R., Chen, M., Daubney, B., Grundy, E., Heidemann, G., Hferlin, B., Hferlin, M., Leitte, H., Weiskopf, D., Xie, X.: State of the art report on video-based graphics and video visualization. Computer Graphics Forum **31**(8), 2450–2477 (2012). DOI 10.1111/j.1467-8659.2012.03158.x
7. Christel, M., Stevens, S., Kanade, T., Mauldin, M., Reddy, R., Wactlar, H.: Techniques for the creation and exploration of digital video libraries. In: B. Furht (ed.) Multimedia Tools and Applications, *The Kluwer International Series in Engineering and Computer Science*, vol. 359, pp. 283–327. Springer US (1996). DOI 10.1007/978-1-4613-1387-8_8
8. Christel, M.G., Smith, M.A., Taylor, C.R., Winkler, D.B.: Evolving video skims into useful multimedia abstractions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '98, pp. 171–178. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1998). DOI 10.1145/274644.274670
9. del Fabro, M., Münzer, B., Böszörmenyi, L.: AAU video browser with augmented navigation bars. In: Proceedings of the Multimedia Modeling Conference, pp. 544–546 (2013)
10. del Fabro, M., Schoeffmann, K., Böszörmenyi, L.: Instant video browsing: A tool for fast non-sequential hierarchical video browsing. In: Proceedings of the 6th Symposium of the Workgroup Hum.-Comp. Interaction and Usability Engineering, pp. 443–446 (2010). DOI 10.1007/978-3-642-16607-5_30
11. Frisson, C., Dupont, S., Moinet, A., Picard-Limpens, C., Ravet, T., Siebert, X., Dutoit, T.: Videocycle: user-friendly navigation by similarity in video databases. In: Proceedings of the Multimedia Modeling Conference, pp. 550–553 (2013)
12. Girgensohn, A., Shipman, F., Wilcox, L.: Adaptive clustering and interactive visualizations to support the selection of video clips. In: Proceedings of the 1st ACM

International Conference on Multimedia Retrieval, pp. 34:1–34:8 (2011). DOI 10.1145/1991996.1992030

13. Huber, J., Steimle, J., Lissermann, R., Olberding, S., Mühlhäuser, M.: Wipe‘n’watch: spatial interaction techniques for interrelated video collections on mobile devices. In: Proceedings of the 24th BCS Interaction Specialist Group Conference, pp. 423–427 (2010)
14. Hürst, W., Götz, G., Welte, M.: Interactive video browsing on mobile devices. In: Proceedings of the 15th international conference on Multimedia, pp. 247–256 (2007). DOI 10.1145/1291233.1291284
15. Jansen, M., Heeren, W., van Dijk, B.: Videotrees: Improving video surrogate presentation using hierarchy. In: Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on, pp. 560–567 (2008). DOI 10.1109/CBMI.2008.4564997
16. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 117–128 (2011)
17. Jégou, H., Douze, M., Schmid, C., Perez, P.: Aggregating local descriptors into a compact image representation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3304–3311 (2010)
18. Jiang, J., Zhang, X.P.: A smart video player with content-based fast-forward playback. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 1061–1064 (2011). DOI 10.1145/2072298.2071938
19. Le, D.D., Lam, V., Ngo, T.D., Tran, V.Q., Nguyen, V.H., Duong, D.A., Satoh, S.: Nii-uit-vbs: A video browsing tool for known item search. In: Proceedings of the Multimedia Modeling Conference, pp. 547–549 (2013)
20. Lowe, D.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, pp. 1150–1157 (1999)
21. Marchand-Maillet, S., Morrison, D., Szekely, E., Bruno, E.: Multimodal Signal Processing: Theory and applications for human-computer interaction, chap. Interactive Representations of Multimodal Databases, pp. 279–308. Elsevier (2009)
22. Meixner, B., Köstler, J., Kosch, H.: A mobile player for interactive non-linear video. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 779–780 (2011). DOI 10.1145/2072298.2072453
23. Mueller, C., Smole, M., Schöffmann, K.: A demonstration of a hierarchical multi-layout 3D video browser. In: Proceedings of the IEEE International Conference on Multimedia and Expo Workshops, p. 665 (2012). DOI 10.1109/ICMEW.2012.121
24. Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Shaw, B., Kraaij, W., Smeaton, A.F., Quénot, G.: Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2012 (2012)
25. de Rooij, O., Snoek, C.G., Worring, M.: Balancing thread based navigation for targeted video search. In: Proceedings of the 2008 international conference on Content-based image and video retrieval, pp. 485–494 (2008). DOI 10.1145/1386352.1386414
26. Schoeffmann, K., Boeszoermenyi, L.: Image and video browsing with a cylindrical 3d storyboard. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR ’11, pp. 63:1–63:2. ACM, New York, NY, USA (2011). DOI 10.1145/1991996.1992059
27. Schoeffmann, K., Cobarzan, C.: An evaluation of interactive search with modern video players. In: Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on, pp. 1–4 (2013). DOI 10.1109/ICMEW.2013.6618282
28. Schoeffmann, K., Fabro, M.d.: Hierarchical video browsing with a 3D carousel. In: Proceedings of the 19th ACM international conference on Multimedia, pp. 827–828 (2011). DOI 10.1145/2072298.2072479
29. Schoeffmann, K., Hopfgartner, F., Marques, O., Böszörmenyi, L., Jose, J.M.: Video browsing interfaces and applications: a review. SPIE Reviews **1**(1), 018004 (2010). DOI 10.1117/6.0000005
30. Schoeffmann, K., Taschwer, M., Böszörmenyi, L.: The video explorer: a tool for navigation and searching within a single video based on fast content analysis. In: Proceedings of the first annual ACM SIGMM conference on Multimedia systems, pp. 247–258 (2010). DOI 10.1145/1730836.1730867
31. Scott, D., Guo, J., Gurrin, C., Hopfgartner, F., McGuinness, K., O’Connor, N., Smeaton, A., Yang, Y., Zhang, Z.: DCU at MMM 2013 Video Browser Showdown. In: Proceedings of the Multimedia Modeling Conference, pp. 541–543 (2013)
32. Scott, D., Hopfgartner, F., Guo, J., Gurrin, C.: Evaluating novice and expert users on handheld video retrieval systems. In: MMM, pp. 69–78 (2013)
33. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and trecvid. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 321–330 (2006). DOI http://doi.acm.org/10.1145/1178677.1178722
34. Snoek, C., Worring, M., de Rooij, O., van de Sande, K., Yan, R., Hauptmann, A.: Videolympics: Real-time evaluation of multimedia retrieval systems. MultiMedia, IEEE **15**(1), 86–91 (2008). DOI 10.1109/MMUL.2008.21
35. Tao, K., Dong, Y., Bian, Y., Chang, X., Bai, H.: The france telecom orange labs (Beijing) video semantic indexing systems. In: TRECVID 2012 (2012)
36. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
37. Wilkins, P., Byrne, D., Jones, G.J.F., Lee, H., Keenan, G., McGuinness, K., O’Connor, N.E., O’Hare, N., Smeaton, A.F., Adamek, T., Troncy, R., Amin, A., Benmokhtar, R., Dumont, E., Huet, B., Mérialdo, B., Tolias, G., Spyrou, E., Avrithis, Y.S., Papadopoulos, G., Mezaris, V., Kompatsiaris, I., Mörzinger, R., Schallauer, P., Bailer, W., Chandramouli, K., Izquierdo, E., Goldmann, L., Haller, M., Samour, A., Cobet, A., Sikora, T., Praks, P., Hannah, D., Halvey, M., Hopfgartner, F., Villa, R., Punitha, P., Goyal, A., Jose, J.M.: K-space at trecvid 2008. In: TRECVID (2008)
38. Worring, M., Sajda, P., Santini, S., Shamma, D.A., Smeaton, A.F., Yang, Q.: Where is the user in multimedia retrieval? MultiMedia, IEEE **19**(4), 6–10 (2012)