Human-based Video Search

Finding the optimal mobile storyboard layout for searching.

Universiteit Utrecht

Rob van de Werken
Thesisnumber: ICA-3363066
August 11, 2015

Contents

1 Introduction

Searching for content within videos can be done in two ways, either using an algorithm and a computer to process the data or by giving a good visualization of the data and letting a human perform the search. The "Video Search Showcase" (VSS) is a contest where innovative and state of the art techniques are used to find a given video fragment or a description within a data set. During the contest different teams compete at doing such "Known Item Search" (KIS) tasks using their own methods and find the locations of the items as quickly as possible within the data set. The aim of the VSS is to evaluate video browsing tools for their efficiency at KIS tasks.

The difficulty of finding a known item within a data set using a computer based search method comes from the semantic gap between computers and humans. Computers are good at searching through large amounts of abstract data quickly when given a query. Whereas humans are good at interpreting the meaning (semantics) of the data they process, but have more difficulty processing large amounts of data. [SLZ+03]

The goal of this project is to find the best visualization of a storyboard with a good user interface design (UI) and good interactions. This allows humans to easily search for a small video fragment or description within an archive of videos. A good visualizations allows humans to process large amounts of data more easily, allowing it to compete with computer based search methods.

Winning, or doing well, at a competition is ofcourse also a nice goal for a project. But more important are the insights into why certain performances are achieved. The focus of this project lies purely on the creation of an optimized UI for participating in the competition of 2015 and scientifically studying the results, which are described in section 2. Additional experiments are done to study the influences of design decisions and the search speed (scalability), which are described in section 3.

2 Part one – The VSS2015 competition

The VSS takes place annually at the begin of the year. In order to participate we started developing a system, based on estabilished research, our own previous related work and some informal prototype studies.

This part starts with an introduction of the VSS2015, which can be found in section 2.1. Followed by the development our contributing system in sections 2.2-2.4. This part will end with the outcome of the actual event and our performance during the contest, in section 2.5.

2.1 The Video Search Showcase

The Video Search Showcase (VSS, formerly Video Browser Showdown) is an annual live video search competition. International researchers evaluate and demonstrate the efficiency of their interactive video search tools during this competition. It takes places as a special session at the International Conference on MultiMedia Modeling (MMM) since 2012. The participating teams first present the details of their systems and then perform several interactive video search tasks. In 2015 the Vide Search Showcase was held at MMM 2015 on January 4, 2015 in Sydney, Australia [BS14, Sch14].

2.1.1 Goal of the competition

The goal of the VSS is to evaluate video browsing tools for their efficiency at "Known Item Search" (KIS) tasks with a well-defined data set and directly comparing them with other tools. The searchers need to find a short video fragment in a single video or a video collection within a specific time limit for each task. The video fragment that needs to be found is given in either the visual version (where the video fragment is played) or a textual description of the fragment. The tasks are performed by not only the participating teams themselves (expert round), but also by people from the audience (novice round).

Originally the contest would consist of two types of search tasks, namely searching in a single file and searching in an archive of ten videos. For 2015, this was changed to searching in a large data set, consisting of 153 videos resulting in a total of around a 100 hours of video. The data set consists of videos made available by the BBC for this contest. Hence it was comprised of videos created by the BBC, consisting of many different types of videos, from documentaries, commentaries, talkshows, concerts to drama series. During the VSS2015 the time limit was set to five minutes per task.

2.1.2 Performance evaluation

The performance of the participating systems is determined by a scoring system, where each task can give a maximum of 100 points. The number of points is based on the submission time and the number of wrong submissions. It linearly decreases from 100 to 50 over the time that is available for the task to solve. It is allowed to make two wrong submissions per task without getting a penalty. If there are more than one wrong submissions, the number of points as a result of a correct submission will be divided by the number of submissions minus one. Formula 2.1 shows the points awarded per task in a formula.

$$\begin{aligned} \text{points}(s,t) &= \begin{cases} 100 - 50 \times t & s < 2 \\ (100 - 50 \times t)/(s-1) & \text{otherwise} \end{cases} \\ \text{where } t &= \text{time passed in percentage} \\ s &= \text{number of submission} \end{aligned} \tag{2.1}$$

For the submission to be correct, a single frame as a result is allowed, so only a single frame of the requested video fragment needs to be found. The submission is still considered correct if the submitted frame is within 125 frames before the start or after the end of the fragment. Formula 2.2 shows the correctness of a task in a formula.

$$\begin{aligned} \text{submission}(f) &= \begin{cases} \text{correct} & b - 125 < f < e + 125 \\ \text{incorrect} & \text{otherwise} \end{cases} \\ \text{where } f &= \text{submitted frame} \\ b &= \text{first frame of requested video fragment} \\ e &= \text{last frame of requested video fragment} \end{aligned} \tag{2.2}$$

2.1.3 Contribution

Humans can process and compare images fast and make a decision based on the content in a split second. When a possible similarity is found the person can then do a more thorough search to find out if it indeed contains the requested information. An additional advantage of the human processing power is the dynamic interpretation of the content. This allows for a more dynamic search, while a computer will only do a search based on what it is told. Therefore humans have a possibility to search based on a description with a better result than a computer can at the moment. The method described here lets the person controlling the device do the search while trying to allow this person to process as much data as possible.

Since the original tasks were really easy for a human to perform, as the search task consisted of only a single video and was therefore not very large, the contribution would be to really compete to the computer based search methods. But as the competition changed to only searching within a very big data set meant that a human would need to process 20 minutes of video data every second to only browse all the data within the time available for the task, the expectations were very low. Therefore it was decided to participate in the contest as a form of benchmark, and not actually to compete. The benchmark would then show the human capabilities in comparison to the computer based methods.

2.2 Initial idea for single file search

The initial idea was to use a storyboard for the contest, with a basic interface and simple button based interactions. A storyboard is a visual representation of a video using images taken from the video. The idea is based on the simple task of finding a video fragment in a limited number of images (couple of screens on a tablet device).

This simple idea is based on experience gained from the competition held previous years. Results from the competition in previous years show that simple designs without data analyses often outperform complex indexing and querying systems [SB12].

2.2.1 Design decisions

Our implementation would be keeping the UI as simple as possible and making the data easily accessible, so the person searching would be able to process the data fast. To achieve this, a few design decisions had to be made.

Images to use

Firstly, the images used to make the browsing possible have to be taken at a regular interval from the source video, as no data analysis is performed on the source videos. The interval chosen was 1 second of video between the images. This is done to make sure there are enough images to search for when the video fragment of 30 seconds is shown, but also not to many. This would result in at least 29 images from the video fragment to be in our used representation.

Image size

Secondly, the size of the images to be displayed needs to be large enough to see the contents without the need to zoom, while also being small to show a large amount of images on a single screen. Literature and previous studies show that these kind of images can be really small, while still being usable [HSST11].

Interactions

Thirdly, the interactions would also need to be kept as simple as possible, in order to keep it also simple for the person using the application. This would reduce the amount of attention the person using the application would lose while using the interface, which could be better used for searching the data visualized [HD12]. The buttons available in the interface allow for scrolling the storyboard, and the storyboard can also be dragged using touch gestures.

Layout

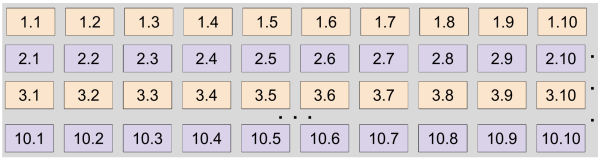
Finally, there were three options being considered for the layout of the storyboard. The first layout considered was a default storyboard with vertical scrolling as can be seen in Figure 2.1a. This is a common design based on the direction of text, namely from left to right, top to bottom. Each line is filled with consecutive images and lines are put beneath each other.

The second layout is based on the first layout, but with a vertical clustering. This vertical clustering can be seen in Figure 2.1b. Compared to the default storyboard layout, the clustered layout has a block-like grouping of the images instead of lines. This makes the spotting of relevant scenes easier (eg. frames 8 to 14).

The third storyboard layout divides all the images across 10 rows to make the interface row-based and horizontally scrollable. This makes a single scrolling action go through the video at 10 different positions in the video at the same time. This makes it possible to scroll through a video very quickly. A schematic drawing of this last storyboard layout can be seen in Figure 2.1c.

(a) Default storyboard

(b) Clustered storyboard



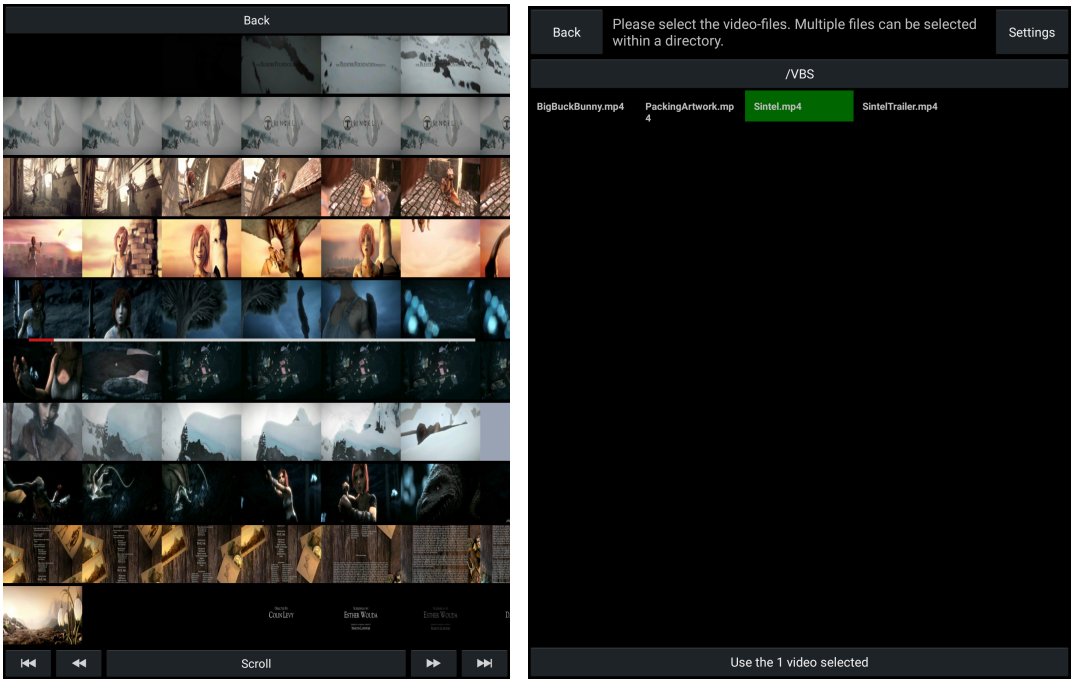
(c) Row based storyboard

Figure 2.1: Schematic representations of initial storyboard layouts

2.2.2 Application

To participate in the contest an application was created, with the initial idea in mind. It would be as simple as possible to achieve the best performance while keeping the code simple to modify. A simple and fast OpenGL environment is used to show a scrollable grid of images in a 2D-plane. A screenshot of the implementation of the initial idea with horizontal scrolling (as seen in Figure 2.1c) can be seen in Figure 2.2a.

The application would also need an option to select a videofile from which the images need to be loaded. Therefore a simple video file selection interface was created using the default android application framework. A screenshot of the implemented filebrowser can be seen in Figure 2.2b. When a video is selected and used, an intent (android event) starts the main application with the selected video.



(a) Row based storyboard (b) File selection dialog

Figure 2.2: Screenshots of initial application

2.2.3 Optimizations

The environment was optimized for a Nexus 7 2013, since the original plan was to use this device during the competition. The Nexus 7 2013 is fast enough to show the grid of images with it's interactions smoothly. This was possible by loading all of the images into memory, when the interface was loaded. This takes a few seconds, but this could be done before the start of the search task. The participants have time to prepare each search task, which would not be counted as search time.

The Nexus 7 2013 has enough memory to be able to load all of the images from a single video into its main memory. Another optimization was done to the interactions. The image grid interactions usually mean a person clicks on a image and performs an action, this can be done in a few ways. The most straightforward implementation is to do a ray trace to find the OpenGL-object intersecting the ray from the view through the position where the person clicked. This is however harder to implement than it sounds, it is also an expensive operation when all the OpenGL-objects need to be checked. A different method is to do a simple calculation to find the image by calculating the position in 2D-space, as the images are all placed in a 2D-plane (with no difference in depth).

The lightweight application, with the in-memory images and the optimized object selection, allows the application to run smoothly on the used device. This speed is of great importance to let the participant search through the images without distractions caused by the application.

2.3 Revised version for ten file search

Shortly before the paper submission deadline, the VSS organisation made the decision to drop the single file task for the 2015 competition and exclusively test the video archive search task. The archive search task in 2014 consisted of finding a 30 second video clip within ten given videos. Hence, our tool needed to be changed in a way that enables quick and easy access to such a set of video files. While this larger data set limits the chances of a pure human-based tool, it is still expected to show a good performance and will give insight into the relevance of a good UI design in the general video search process.

2.3.1 Design decisions

The application for the contest was therefore changed to show 10 videos underneath each other, where each video is shown in a single row. This would be similar to the third layout of the initial ideas. But instead of using one video across multiple rows each row would consist of a single video. The device would also be used in portrait mode to fit the images more nicely on the screen. The time between every image is again one second.

To make the scrolling more controllable and faster to operate, the decision was made to add a button with relative scrolling speed. The speed of the scrolling was determined by the location of the press on the scroll-button. The further away from the center the faster the scrolling goes. Fast scrolling showed a negative side-effect, as continuous scrolling causes a blurred image for the viewer.

The different videos are also sorted by length, with the longest on top and the shortest on the bottom. This gives the advantage of ignoring the videos on the bottom when they have been scrolled. The videos that have been scrolled can be recognized by obvious images that are shown when all the images of the video have passed. So it becomes easier to focus on only the videos that still need to be processed.

2.3.2 Application changes and optimizations

On the programming side the larger set of images also caused the need for a optimization, since the number of images that had to be loaded increased by a factor of ten. This meant that not all images would fit inside the memory of the device. Therefore only a list of all the filepaths of the images were loaded in an arraylist of arraylists, instead of all the actual images. Then only the images on screen or close to screen position would be loaded based on the file paths. This was done in the background when they were about to be shown, to avoid an delay due to the loading of the images. This meant that for each video two more frames were loaded to the left and to the right of the screen. This showed a huge speed improvement. And allowed the interface to run smooth again, with the larger dataset. A screenshot of the implementation can be seen in Figure 2.3.

The ten videos also needed to be selected to make it possible to participate with the main application. This meant that multiple files had to be selected. In order to make the application more dynamic, it was also made possible to let the application create the preprocessed data of

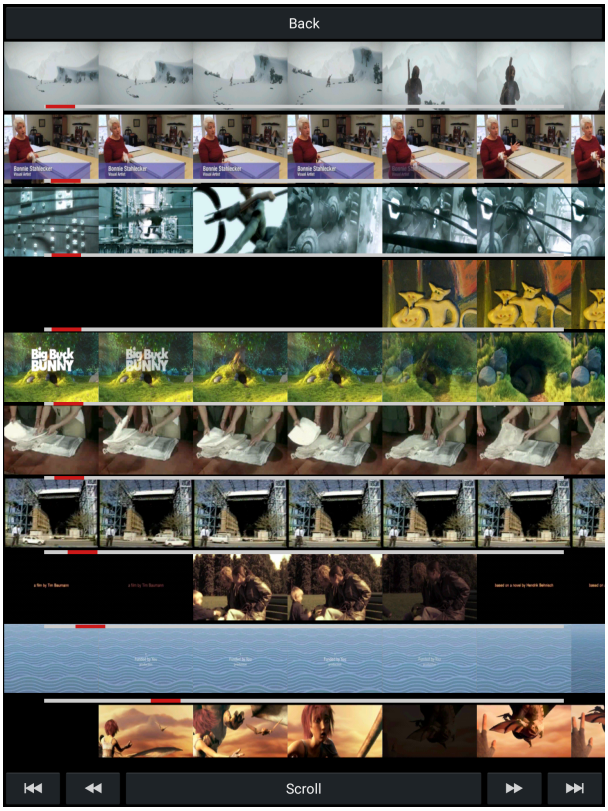


Figure 2.3: Screenshot of 10 film-strips

all the images from each video. This makes it possible to use the application for every video on the device without the need of preprocessing the video on a computer. This takes a long time, as a video is played in the background and the different images are extracted. This was done using the android-ndk with the ffmpeg library, and calling the function using C++ instead of the java-code used normally for android applications.

2.3.3 Observation

This revised version allows the contestant to scroll through the videos at a dynamic speed. But as the total number of images is much larger than with the single video it was noticed that it becomes harder to keep track of the images passing the screen during fast scrolling. This would result in a overwhelming amount of information for the contestant. Increasing the speed even more showed a phenomenon, which makes the task of searching within each row easier. At certain speeds (where the amount of scrolling within a single refresh of the screen is equal to a multiple of the width of an image) the row looks as if it is not moving, but the images would resemble a fast-forwarded version of the video. Using such a fast-forward method could make it easy to get an impression of the contents of the videos.

2.4 Final revision for large video archives

Surprisingly late, the organization of the VSS2015 made the decision for another change in the competition. Compared to the archive search task of 2014 the task was extended from just ten videos to 153 videos. The 153 videos result in a total of about 100 hours of video.

It was clear that a pure human-based search approach would likely fail. Yet, we were still convinced that the related knowledge gained about, roughly said, "how far can we get" relying purely on interface and interaction design and related human browsing abilities would be valuable. This would mean we were not going to compete against the other teams, but participate to create more of a "human benchmark".

The increased number of videos also meant that the application had to be changed in a way that would allow for searching in a very large archive. Therefore new design decisions had to be made.

2.4.1 Design decisions

Based on the previous designs, two ideas have been implemented. In the first one the initial version of a classic storyboard would be used, by concatenating all the videos in the vertical direction. This creates a very long storyboard which can be scrolled in a linear way, similar to the representation of Figure 2.1a. The concatenation could be performed in different ways, namely in an random order arranged or an ordered way with either the shortest video or the longest video first. The other main interface would be based on the row-based design, where it is possible to scroll in a horizontal way to search linearly within a set of ten videos and vertically to search within the different videos, resulting in a 2D interface (as can be seen in Figure 2.4). This way all of the data from all the videos is accessible.

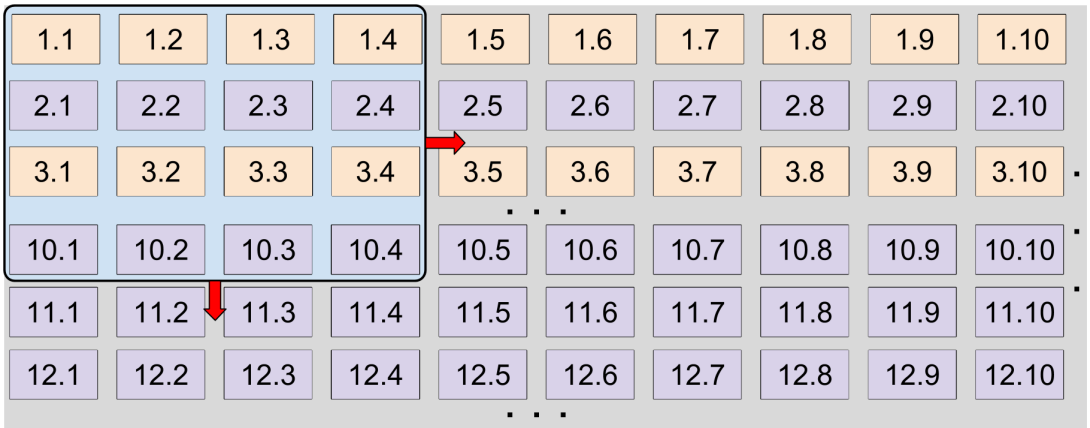


Figure 2.4: Schematic representation of 2D storyboard.

But not only the task changed for the third implementation, the device also changed to a newer tablet. The new device is a Nexus 9, which has a more powerful processor, a higher resolution, a higher pixel density and a different aspect ratio (4:3 instead of 16:9). This allows the new implementation to have more and smaller images on the screen while still having lots of detail per image which should not have a great impact on the searching performance [HSST11]. This

would allow the device to be used in landscape mode again and increase the number of videos on a single screen from 10 to 25 videos. This also increased the number of images on a single screen from 50 images to 625 images. This way there would be more information on the screen, reducing the amount of scrolling required to get through all of the data.

During the testing of this method it felt not as usable as expected, and posed quite some difficulty to browse all of the videos. This was mainly caused by the difference in length of the videos. An option would be to concatenate the different videos into strips with roughly the same total length and remove the vertical scrolling. But this would impose a different problem, where the person scrolling would not directly notice the change between video files within one strip. This could influence the performance because in many cases one would ignore the strip of frames if it is obvious the requested fragment would not be in the video-file. For this reason the different videos would be put beneath each other and use the vertical scrolling.

The tests also showed that 25 different videos beneath each other is hard to follow. A clustering of the images would reduce the amount of videos on a single screen and also gave the impression that the scenes were easier to recognize. So a clustering of five images beneath each other would be used for each row of images, reducing the amount of videos on a single screen to five.

2.4.2 Application

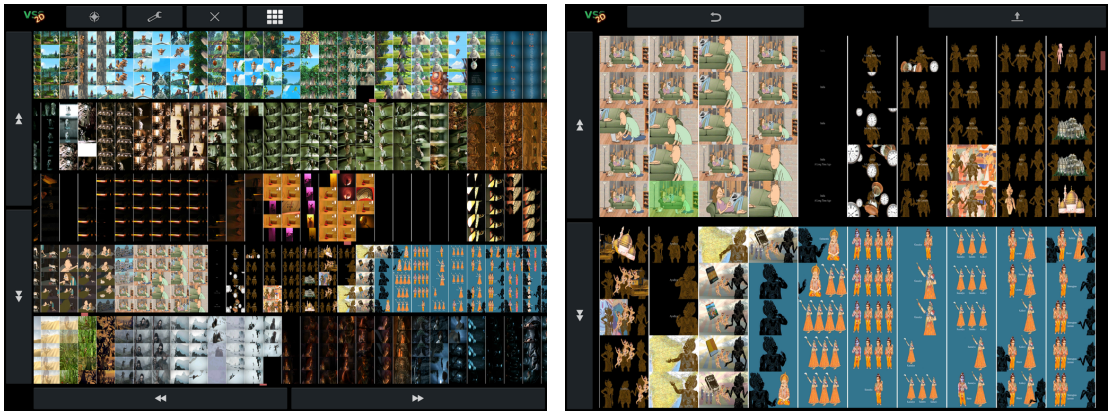
The third application is based on the revised version from section 2.3 and extends it in the vertical direction to make it also possible to scroll vertically through the different videos instead of only horizontally through the images. Some of the earlier optimizations, which were specifically designed for the Nexus 7 device, also needed to be finetuned to work nicely on the Nexus 9. However, as the amount of data that had to be loaded on the screen of the device again increased by a large amount (around 13 times), new optimizations were also required to make the application run smooth once again.

2.4.3 Optimizations

Although this version was created for the newer Nexus 9, the increase of the number of files on the screen also caused a huge slowdown as the limit of the file-loading speed was reached on the used device. This required a change in fileformat from simple image-files to optimized textures. The texture-format chosen for the images is ETC1, since the used device would be able to load those compressed textures directly into the video-memory without the need to decompress or change them before displaying the textures on the screen. Even with the optimizations in place it showed a huge slowdown since a huge amount of files need to be loaded and showed. This would not show a huge impact when continuous scrolling at a low speed where the images can be viewed. However, using the paged scrolling would increase the amount of file-loading done for each scrolling action to an amount at which the device could not handle it unnoticed in the background. This problem also exists when the continuous scrolling speed increased. Therefore another optimization was required to allow for a faster loading of the images. A different optimization method would be to use textures containing multiple images. This reduced the amount of loading time dramatically to again allow for an unnoticed loading of the images and allow smooth scrolling.

The preprocessed textures also made it easier to implement the clustering. Only a minor change in the application was needed for the calculation of the image selection to simulate a single image selection within a texture. This also meant that the dataset had to be preprocessed again, with a different grouping of the images in the textures.

A screenshot of this 2D version can be seen in Figure 2.5a. It shows five different videos beneath each other. In order to take a closer look at the small images, a zooming feature was also implemented. This zooming feature can be seen in Figure 2.5b.



(a) Normal (b) Zoomed

Figure 2.5: Screenshot of 2D layout

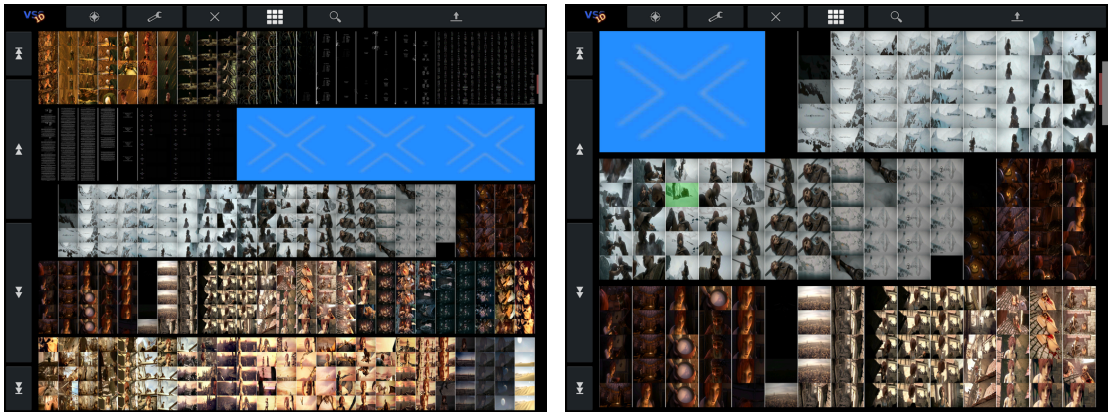
2.4.4 Observation and redesign

Although this implementation has some potential, it posed the difficulty of scrolling in two directions. This called for a rethink of the used layout. The 2D implementation showed it was possible to give a full representation of the videos, but browsing it would be too cumbersome and complex under the time pressure imposed by the competition. In order to make the navigation simpler, and thus presumably faster, it was therefore decided to remove the horizontal scrolling and use the initial version with all the videos put beneath each other. The resulting application would then be a 1D implementation, which also meant another revision of the code. The layout of the different images would be the same as the initial version, but would require the optimizations present in the 2D implementation.

Due to the small size and the huge amount of images on a single screen, it was noticed only a glance at the images was enough to decide if it would be necessary to take a closer look at a certain set of images. This is because mainly scene changes are used to decide if a requested fragment is within the set of images on a screen. Motivated by this observation, we therefore decided to also use the clustering with the new design. The clustering would again allow the person using the application to more easily recognize groups of images based on scenes.

2.4.5 Final application

As a result of the redesign, another application was created. But since it is a combination of multiple earlier created designs with their optimizations, it was easy to create a new application that runs smooth on the used device. A screenshot of the resulting application can be seen in Figure 2.6a.



(a) Normal (b) Zoomed

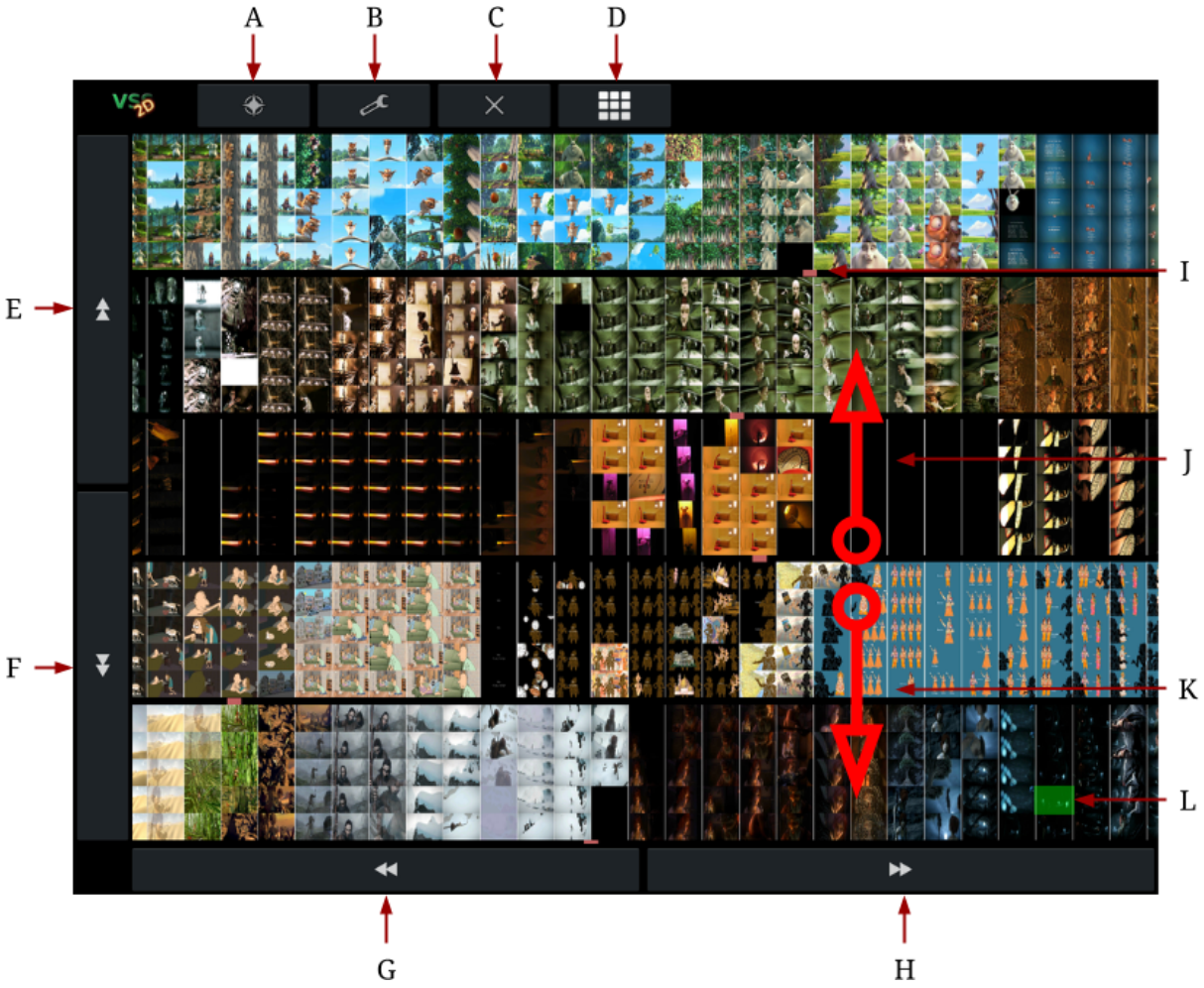
Figure 2.6: 1D storyboard interface

2.4.6 Final observations

Given the time restrictions imposed by the VSS competition, we relied on some informal testing of the clustering layout and delayed a detailed evaluation to a later date (cf. Part two). Because the results of these informal studies suggested an added benefit of the clustering, a layout with five images beneath each other would be used for each row of images with those two main interfaces. This gives in both cases most likely the benefit of easier scene recognition by the user, but in the film-strip case it would also reduce the number of videos that could be shown simultaneous on a screen. But as there would be more videos on the screen, it would also be harder to follow them. Therefore a reduced amount of videos on a single screen would most likely be a benefit.

The 1D version was the one that was ultimately used during the actual event, but both the 2D and 1D version were available for use during the competition. In the following, a complete description is given of the layout and functionality of the tools that have been created for the VSS2015.

2D Interface



A. Scroll back to the top.

B. Open the configuration dialog, giving access to the configuration for the communication with the contest server.

C. Close the application.

D. Grid configuration options, giving access to the sorting order of the videos. There are additional layout options implemented for testing. These additional options are for number of files per screen, cluster size and the time between the thumbnails.

E. Scroll up by a screen.

F. Scroll down by a screen.

G. Scroll left by a screen.

H. Scroll right by a screen.

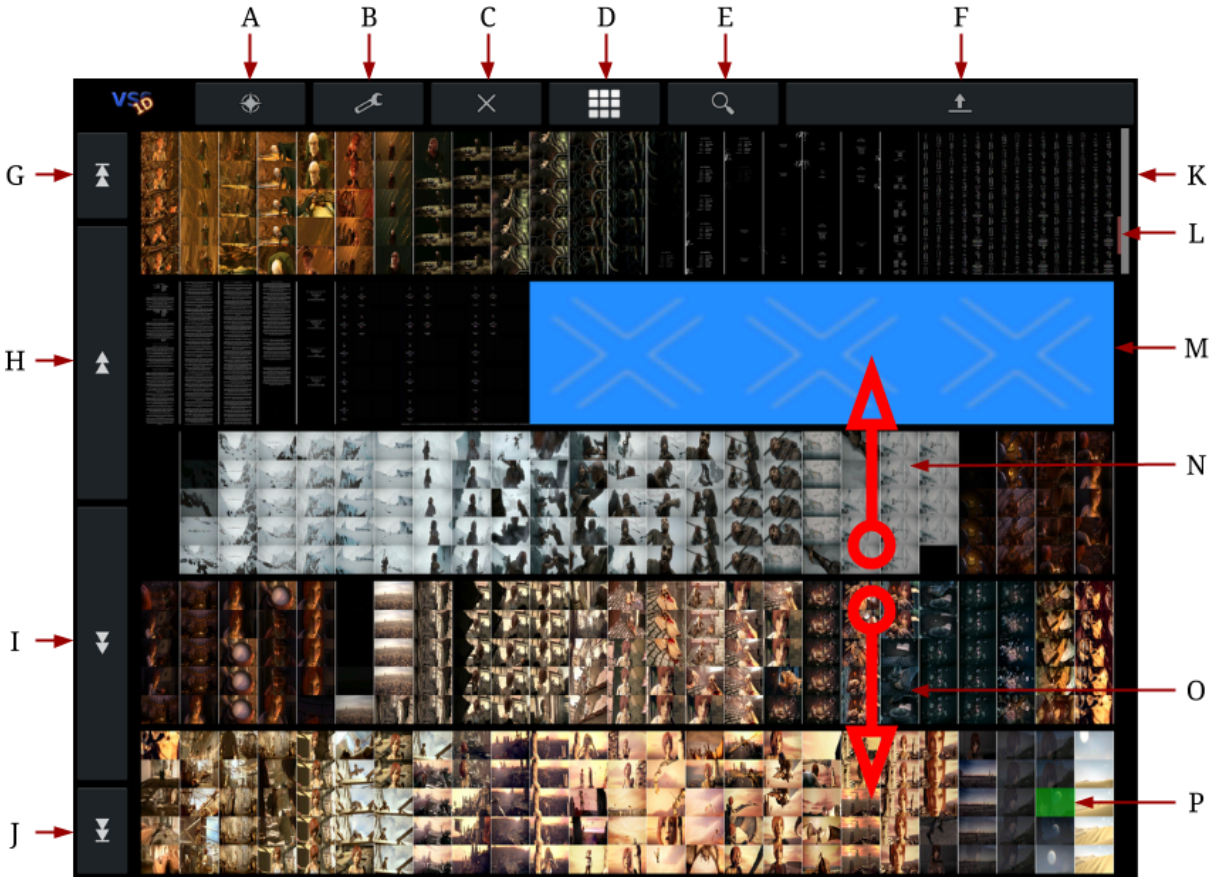
I. Scroll indicator for the file above the indicator.

J. Scrolling interaction down, can be done anywhere on the grid of images.

K. Scrolling interaction up, can be done anywhere on the grid of images.

L. Highlighting the selected images. Selection is done by clicking an image.

1D Interface



A. Scroll back to the top.

B. Open the configuration dialog, giving access to the configuration for the communication with the contest server.

C. Close the application.

D. Grid configuration options, giving access to the sorting order of the videos.

E. Zoom in.

F. Submit selected frame.

G. Scroll up to previous file.

H. Scroll up by a screen.

I. Scroll down by a screen.

J. Scroll down to next file.

K. Scroll indicator for current file.

L. Scroll indicator for entire dataset.

M. Filling images, indicating the previous file ended.

N. Scrolling interaction down, can be done anywhere on the grid of images.

O. Scrolling interaction up, can be done anywhere on the grid of images.

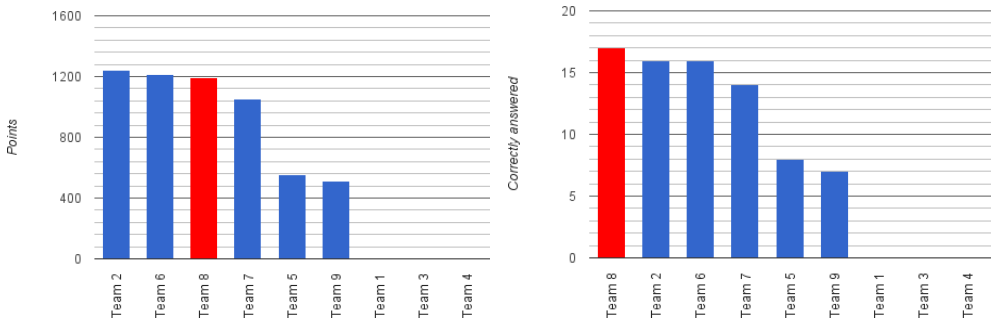
P. Highlighting the selected images. Selection is done by clicking an image.

2.5 Contest results

2.5.1 Aim and expectation

With a database of 100 hours of video, it seems impossible to seriously compete against systems doing automatic video analysis and, for example, query-based filtering using pure human-based searching that solely relies on an optimized interface and interaction design. Our aim in participating in the VSS 2015 event was therefore not actually to compete with the other systems, but to rather create a "human benchmark", where we verify how far one can get without machine support. By doing so, we were hoping to gain further insight into the potential, but also limitations of good interaction design for this kind of tasks. By achieving a decent or even good performance, we were also hoping to create awareness of the relevance and potential of good interface design. This is an issue that in our opinion is currently too often overlooked by the related research community. While we were well aware of the task's difficulty, and even the risk of total failure by scoring low and ending last place, our experience suggested that human browsing performance combined with our optimized design might actually perform quite well. With chances we hoped to end up, in the second third of the field or maybe even the top 50%. Yet, even this seemingly optimistic expectation turned out to be a huge underestimation. The result of the competition proved our point about the value of good interface design to a degree that even surprised us and exceeded even our most optimistic expectations. We present the results in the following subsections.

2.5.2 Contest results



(a) Scores (b) Correctly answered

Figure 2.7: Final standings

Figure 2.7a shows the final score achieved by all participating teams. Our system is represented as team 8 during the contest. And as can be seen, our system came in on a more than surprisingly good third place, very close to the winning team. In fact, looking not at the score but the number of correct files found, which is shown in Figure 2.7b, we see that our approach actually found more solutions than any other competing team. Figure 2.8 illustrates the development of the score over time. Figure 2.8 showes the total progress of the scores, where the white background stands for expert visual, grey background for expert textual, yellow background for novice visual and a dark yellow for novice textual tasks. We see that our system started rather low, likely explained by a necessary learning curve, but quickly caught up and came quite close to the top performers. Next we will look more closely into the different parts of the contest.

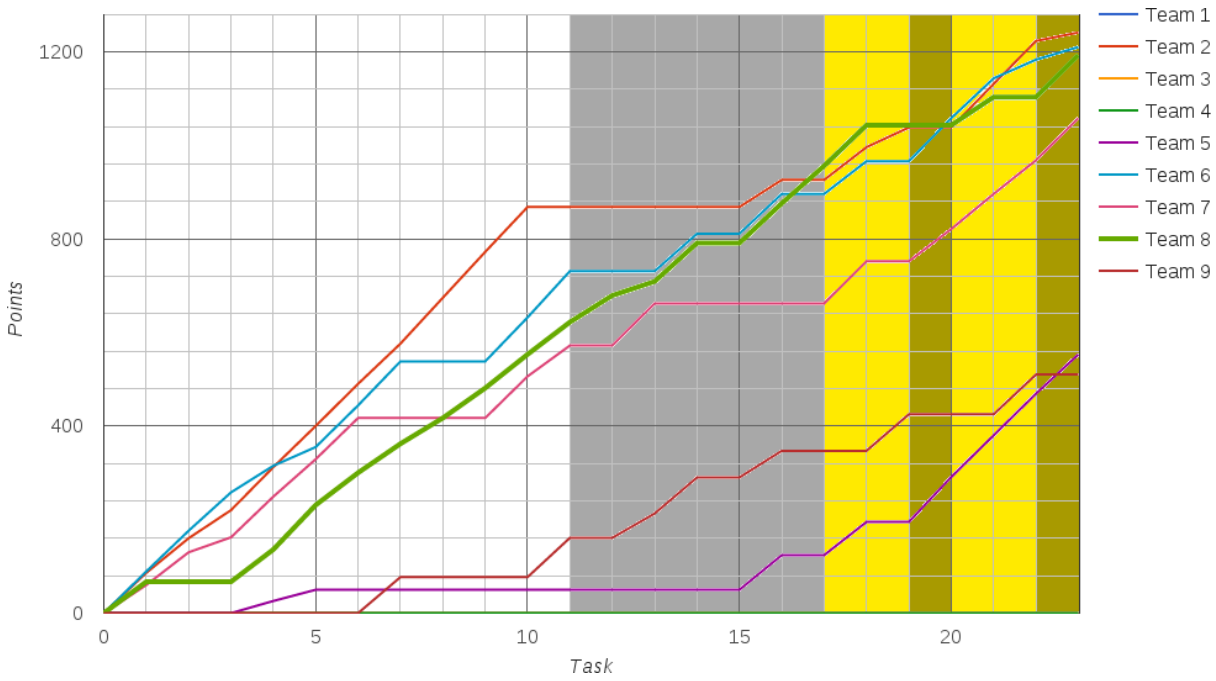


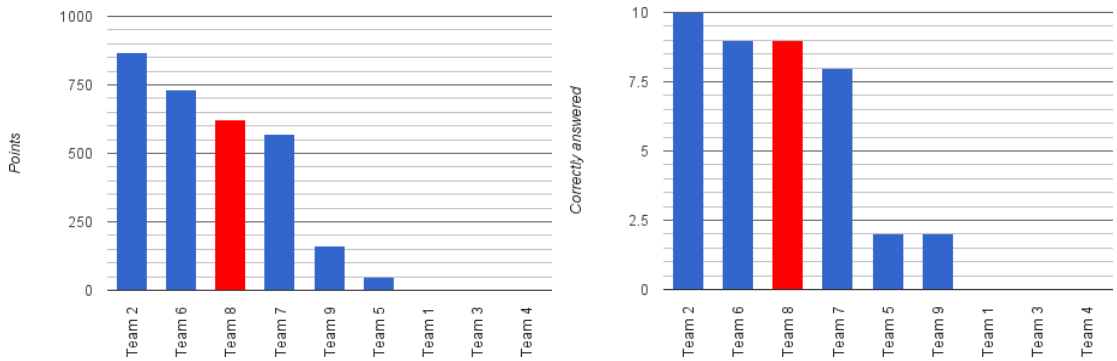
Figure 2.8: VSS 2015 score

Expert results

Looking at the expert results more closely shows a good third place for the visual tasks (Figures 2.9a and 2.9b). When the tasks switched from visual to text, our system clearly outperformed the competition (Figures 2.9c and 2.9d). This performance leap seems understandable given the semantic gap between the textual description and the visual processing most analysis systems relied on.

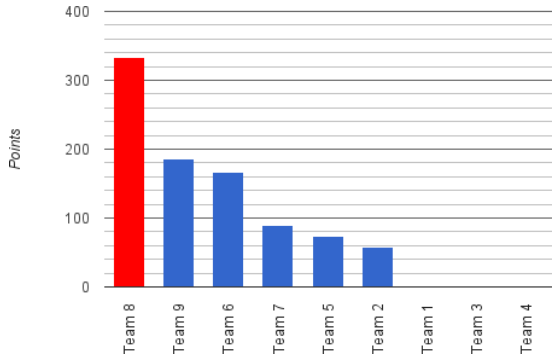
Novice results

Performance dropped a little when the tasks switched from the expert to the novice user round (tasks 17-23, the yellow area in Figure 2.8), which was expected given the unfamiliarity of the user with the system. Although the novices were unfamiliar with the system, they still surprisingly managed to get a few tasks answered correct in both the visual and textual tasks (as can be seen in Figures 2.9e, 2.9f, 2.9g and 2.9h).

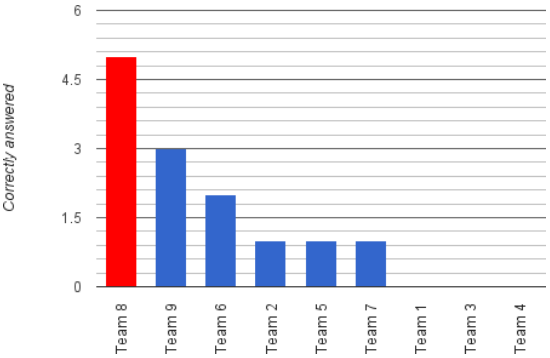


(a) Visual score experts

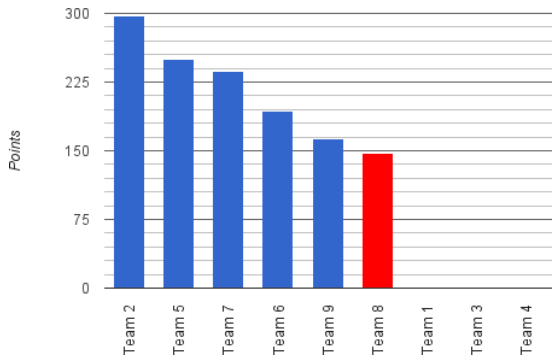
(b) Visual correct experts



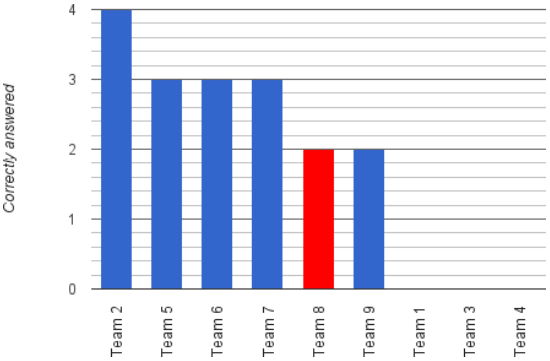
(c) Textual score experts



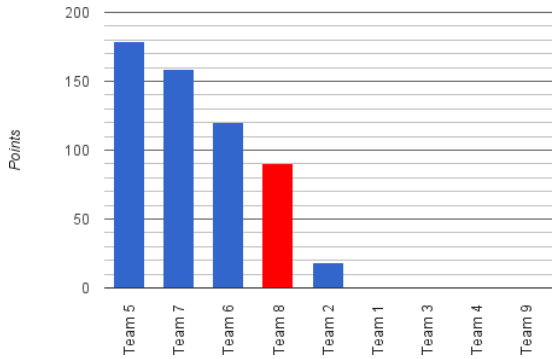
(d) Textual correct experts



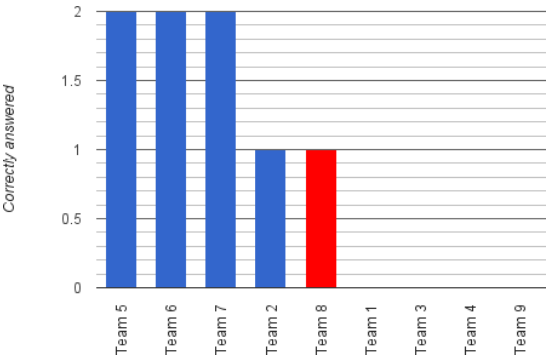
(e) Visual score novices



(f) Visual correct novices



(g) Textual score novices



(h) Textual correct novices

Figure 2.9: Scores split by task and group

2.6 Sub conclusion

The third place in the contest shows a remarkably good result for the human based search, in comparison to the computer based competition. There are several possible explanations for this unexpected good result. For example, the hard part of the computer based search methods is the query, as the requested fragment needs to be translated to a form which the computer can search. Not everything will be translated into the query for the computer, so there will be some information loss. There may be some relevant characteristics of the fragment which won't be put in the search query. This will decrease the chance of finding the correct part.

There are two main conclusions that can be drawn from this contest and this setup. Firstly, the created interface shows it is possible for a human to process a lot of visual information in a short time in order to search for a requested video fragment, since both expert and novice user were able to solve a good amount of the given tasks. Secondly, the experience with the interface and way of searching has most likely an influence on the performance of the user, since there was clear performance difference between the expert who performed exceptionally well, and the novice who still had a good, but significantly lower performance.

Although the results of the created system show that it scales much more than expected, especially for the amount of data that can be processed using it, it will certainly not scale indefinitely. Section 3 will look more into the influences of the design choices made during the creation and the scalability of such a system.

3 Part two – Design parameters and related impact

The extraordinary performance of our system at the VSS2015 competition came as a huge surprise, not only confirming our most optimistic expectations, but even exceeding them. Thus clearly proving our claim about the relevance and importance of a good interface and interaction design for successful and efficient video browsing.

In the following, we will further investigate this potential by addressing questions about the optimal cluster size (section 3.2), the influence of the layout (section 3.3) and potential limits with respect to scalability (section 3.4 and 3.5).

3.1 Resulting research opportunities

VSS2015 showed the potential capabilities of human based searching within a large video archive. The performance of human powered searching is influenced by the visualization and interactions used to browse the video archive. Yet, there are a lot of options to visualize a large archive of videos, for example either by showing clips or small images. A lot of information can be shown by using a grid of small images of the videos. Such a grid can be arranged in different ways. The most obvious way to arrange such a grid is in a linear way, similar to reading and the first version implemented for VSS2015 (cf. section 2.2). Other options are using some form of clustering, to make similarities within sequential images more obvious (cf. section 2.2.1). Clustering can be achieved in multiple ways, while maintaining a linear relation between the thumbnails similar to the fourth version implemented for VSS2015 (cf. section 2.4.5). A different way of clustering the images would be for example color based. This would however mean the temporal relation between the images is lost, in which case consecutive scenes could be put far apart. This would also result in a different kind of research.

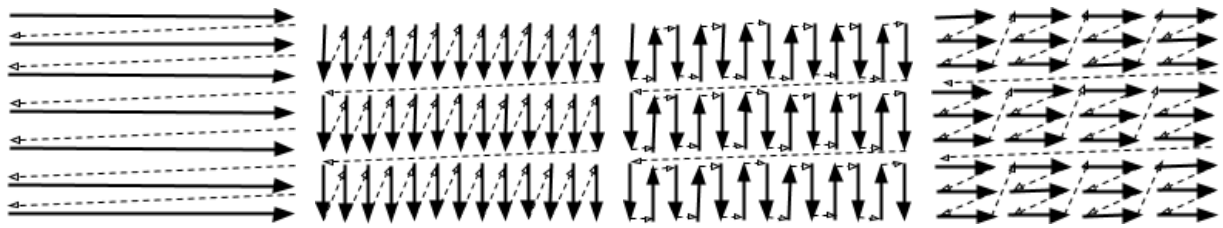
3.1.1 Motivation and parameters to evaluate

Our VSS2015 results suggest that humans can benefit from the type of ordering used in the layout of the images. Using a form of clustering could make it easier for a human to find a certain fragment within a set of thumbnails. Clustering will group thumbnails that are close together in time also in the visualization closer together, compared to a linear layout.

Yet, finding the optimal grid layout is not obvious, but requires scientific experiments to identify the difference in performance between the different types of clustering and the unclustered layout. In addition to the default layout that is based on the common left-right and top-down reading direction and the cluster version used in the VSS2015 competition, there are plenty of other possible arrangements. The most promising ones for further investigation seemed to be the following four:

- Default: no clustering, see Figure 3.1a
- Column: clustering in vertical direction, see Figure 3.1b
- Snake: clustering in vertical direction with alternating up and down direction, see Figure 3.1c
- Row: clustering in horizontal and vertical direction, see Figure 3.1d

We will investigate them in section 3.3.



(a) Default layout (b) Column layout (c) Snake layout (d) Row layout

Figure 3.1: Different layouts

For a cluster-based design, an important parameter to investigate is the optimal cluster size. The clustering can be done with different cluster sizes. The cluster size determines the number of images that are used for a cluster. So for a column clustering (like in 3.1b), this would mean the number of images vertically grouped together. This could also be of an influence on the optimal layout. We will present related experiments in section 3.2.

While our approach worked extremely well, even with the given large archive of 100 hours, it is clear that it doesn't scale indefinitely and that there has to be some limit with respect to a managable archive size. To gain some insight into these limits with respect to scalability, it is therefor useful to know the minimum time required for a human to process a screen of images. This will give an indication on how long it will at least take for a human to process an archive of videos. We will discuss related experiments in seciont 3.4 and 3.5.

3.1.2 Goal

The goal of the research is to gain further insight in how to create the optimal layout design for a storyboard, where the optimal layout design results in a fast search with a low amount of errors. The subjective experience and usability of the different layouts will also be analysed for additional information about the different designs. The speed at which humans are still capable of searching within a set of images is also part of the research, to find a upper bound at which the screens can be scrolled.

In particular, we are interested in measuring the following performance indicators:

Speed layout

Each layout will be measured in time it took to find the requested fragments. This will give an average time. The lower the average time, the better the result of the layout.

Error-rate layout

Finding the incorrect fragment is not desired, but can happen. Therefore the error-rate is measured for the layouts. If there is a high error-rate for a layout with a low average time, then the layout is less desirable than a layout with a somewhat higher average time but lower error-rate.

Speed scalability

The time it takes to make a judgement of a screen of thumbnails is measured, to indicate the time it takes to look at a single screen to make decision based on the contents of the screen. This will make it possible to give an estimate on long it will take to browse a data set.

Error-rate scalability

When the speed at which a decision has to be made decreases, the error-rate should also decrease. This is caused by the short time in which a decision has to be made based on a short glimpse at the grid layout. The goal is to find the fastest speed at which a person can still make a mostly correct judgement of the contents of a screen while searching.

3.2 Cluster size experiment

In order to compare the performance of different cluster-layouts (cf. section 3.3) to the structured representation, it is first important to gain further insight into good cluster sizes. The following cluster size experiment is designed to compare the different cluster sizes and find the best size for the column cluster (cf. Figure 3.1b).

The cluster sizes used in this experiment are 1, 3, 5 and 8. These values are chosen for specific reasons. The cluster size of 1 resembles the default layout without clustering. The cluster size of 5 is the same as used during the contest. The cluster sizes 3 and 8 are chosen close to the cluster size of five, but at the same time fill nearly as much rows of a single screen as possible with complete clusters, while at the same time being not too close to make a large enough difference.

3.2.1 Participants

The experiment is held within the course of "Multimodal Interaction". Therefore the participants are all master students from Computer Science-related subjects. They have knowledge of mobile devices and are in the age range between 20 and 30 years. Half of the group will do the cluster size experiment, resulting in a total of twelve participants for this experiment.

3.2.2 Setup

Questionnaire

A questionnaire will be used to get background information about each participant. Furthermore, the subjective experience of the user is asked to show if the measured performance compares to the subjectively experienced performance. The questionnaire used for the cluster size experiment can be found in Appendix A. The order of the clusterings in the questionnaire is similar to order in the data set, which will be described in 3.2.2.

Application

A modified version of the VSS2015 application was created to do the experiments with. It was extended with options to load task-files and show additional dialogs.

For each task the application will show the fragment, wait for the user to become ready to search and then show the clustered layout. When the participant has found the correct part, a message is shown and the next video will be shown when the participant is ready. The participant has an upper limit of one minute to find the fragment within the five screens he or she has to search.

Data set

A data set based on VSS2015 data will be used. The data set consists of 36 separate tasks, which is equal to the number of tasks a participant has to perform. Every task has to be completed by the participant and contains a video fragment (which has to be found) and a set of 3000 thumbnails, which results in a total of five screens. The thumbnails used for a task are unique within the data set and are created from videos of the VSS2015 data. To prevent partial

cluster visibility, since 25 is not dividable by 3 and 8, the screens the cluster sizes 3 and 8 will show 24 rows instead of 25. This results in the final clustered row of the cluster size 5 to be unused. It will make sure that in all cases the screens consist of full clusters. Each participant will use the same data set, but in a random order.

The order in which the different clusterings will be tested is organised in a latin square as illustrated in Table 3.1.

Participant	Cluster size order			
	1st	2nd	3rd	4th
1, 5, 9	1	3	5	8
2, 6, 10	3	1	8	5
3, 7, 11	5	8	1	3
4, 8, 12	8	5	3	1

Table 3.1: Order of cluster sizes for the different participants.

Procedure

The following procedures describe, in a step by step manner, what a participant will be doing for the experiment.

Introduction
The goal of the experiment is explained, with the help of the questionnaire. The task he/she has to do is explained together with the interface.

For each cluster size
For each cluster size the participant gets one training task with the respective cluster size and the experiment setup. The training is followed by all the tasks for this particular cluster size.

- Show a 30 second video clip
- Wait for the user to press start
- Show the layout with thumbnails, where the user has to find the correct location within 180 seconds.
- The participant gets to know that we are going to start with the actual experiment.
- The 8 tasks start. Each task consists of:
  - Show a 30 second video clip
  - Wait for the user to press start
  - Show the layout with thumbnails, where the user has to find the correct location within 60 seconds.

Final questions
At the end, the participant is asked to fill in the last part of the questionnaire, verifying the subjective experience.

3.2.3 Findings

Qualitative ratings

The results from the questionnaire are broken down in three distinct categories, namely fondness, quickness and accuracy. The fondness indicates if the participants like the cluster size or not. The quickness indicates if the participants perceived the cluster size as quick. The accuracy indicates if the participants think they made a lot of mistakes or not. Ratings are done using the following scale (from worst to best): --, -, +, ++.

Fondness

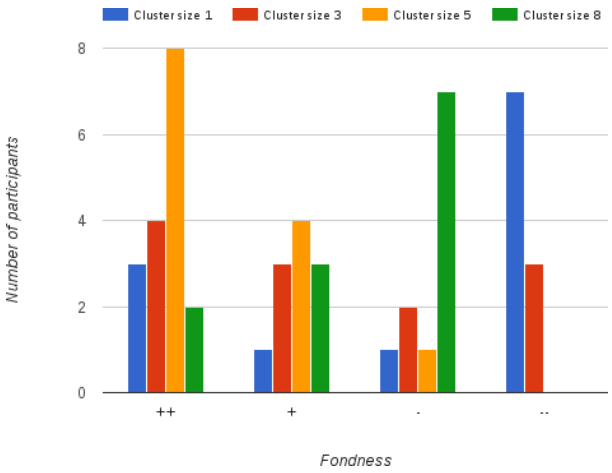


Figure 3.2: Perceived cluster size fondness

The ratings for fondness clearly indicated that the participants like the cluster size of 5 the most. And dislike the cluster sizes of 1 and 8 the most.

Quickness

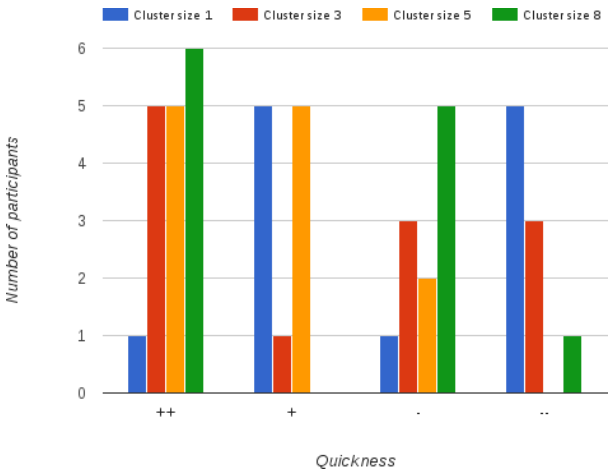


Figure 3.3: Perceived cluster size quickness

The ratings for quickness are less clear than the results of the fondness, but most of the participants think the clustersize of 5 is the fastest for the task.

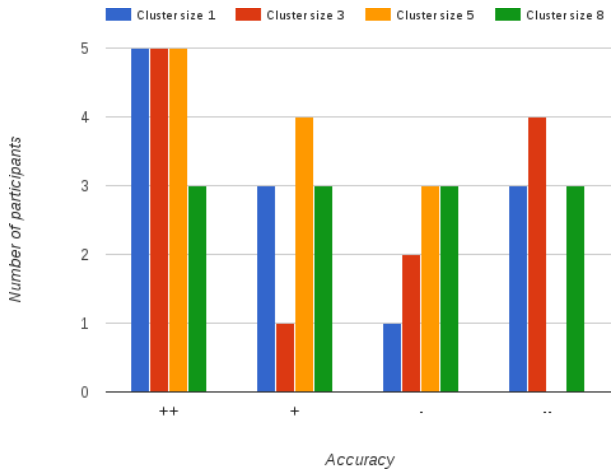


Figure 3.4: Perceived cluster size accuracy

Accuracy

The ratings for accuracy show a small preference for cluster size 5. The other cluster sizes show around the same amount of participants thinking they performed well as bad.

Quantitative performance results

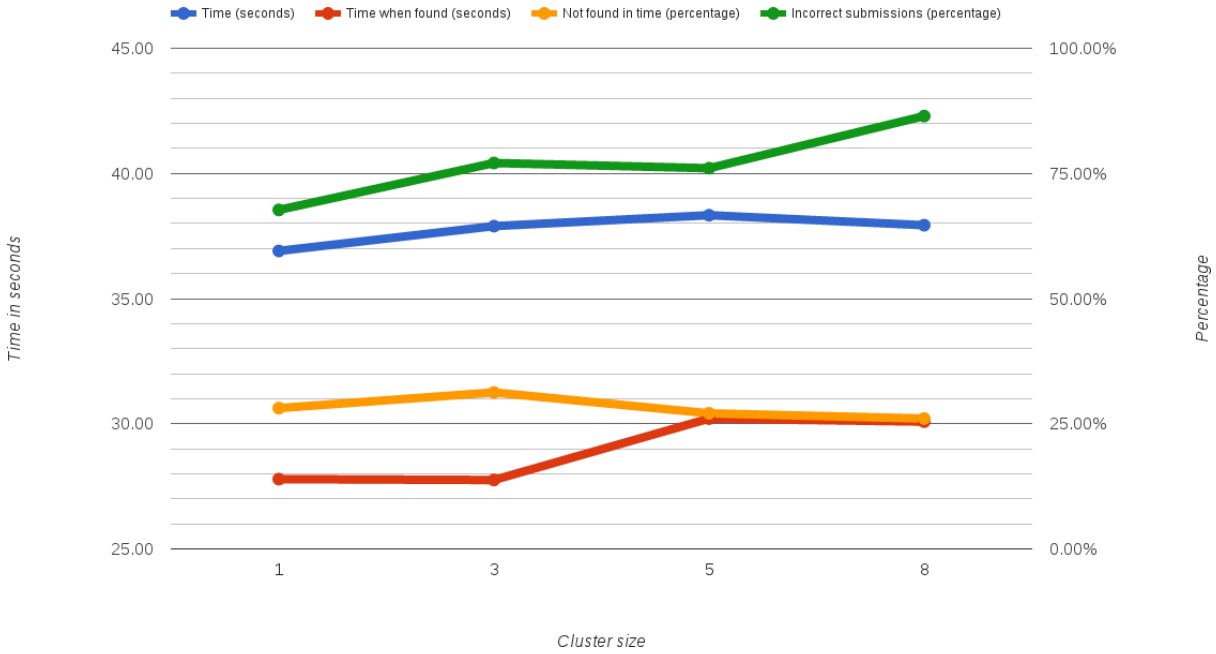


Figure 3.5: Cluster size averages

The results of the cluster size experiment can be seen in Figure 3.5. The values are averaged over all the tasks and participants. In all the cases lower values are better. The combination of "Time when found" and the "Not found in time" is important for a good score in the actual competition. The "Incorrect submissions" also needs to be low, but as this value represent the number of allowed incorrect submissions, a value under 100% would not make a difference during the actual competition, as a single mistake is allowed without a consequence for the score.

In contrast to the participants' subjective impressions, the qualitative measures show a favor for the cluster size of 1, which is similar to the default layout. In this experiment the cluster size of 3 comes on a close second place. The cluster size of 5 and 8 perform almost equally bad in this experiment, but the cluster size of 8 has more incorrect submissions.

3.2.4 Sub conclusion

Based on some informal testing (cf. section 2), we decided to use a cluster size of 5 in the VSS 2015 competition. Because of the positive result of this contest, we could assume that this cluster size is indeed good, and maybe even the best one. The qualitative statements of the participants in our cluster size experiment seem to confirm this expectation, with participants generally expressing a preference towards this cluster size. However, the qualitative results of this experiment do not confirm this. In fact, the good performance of cluster size 1, which represents the default, technically "non-clustered" representation, even puts it into question if there is any benefit of clustering at all. In order to gain more insight into the question if the cluster representation had any positive impact on the VSS 2015 results, we therefore decided to use cluster sizes 1 (i.e., the default representation) and 5 (i.e., the one used in the VSS 2015 competition) for the comparative layout tests in the next section, even if the results of the cluster size experiment presented in this section would suggest using a cluster size of 3.

3.3 Layout Experiment

In section 3.1, we introduced different layout options for storyboards. The following layout experiment aims at comparing these different layouts and gain insight into what might be the optimal design.

3.3.1 Participants

This experiment was done in parallel to the cluster size experiment (section 3.2). Hence, the characteristics of the participants is similar. In particular, we have a comparable group 12 master students participating in the experiment with similar experience and in the age range between 20 and 30 years.

3.3.2 Setup

Questionnaire

A questionnaire will be used to get background information about the participant and an impression of the user experience. The user experience is asked to show if the measured performance compares to the experienced performance. It is also important to see if the participant understands the layout used. The questionnaire used for the layout experiment can be found in Appendix B. The order of the clusterings in the questionnaire is similar to order in the data set, which will be described in 3.3.2.

Application

The application for this experiment is similar to the one used in the cluster size experiment (section 3.2.2).

Like in the cluster size experiment, each task will show the fragment, wait for the user to become ready to search and then show the layout. When the participant has found the correct part, a message is shown and the next video will be shown when the participant is ready. The participant has an upper limit of 1 minute to find the fragment within the 5 screens he or she has to search.

Data set

A data set based on VSS2015 data will be used. The data set consists of 36 separate tasks. Every task has to be done by the participant. Each task will consist of a video fragment (which has to be found) and a set of 3125 thumbnails, which results in a total of 5 screens. The thumbnails used for a task are unique within the data set and are created from videos of the VSS2015 data. Each participant will use the same data set, but in a random order.

The order in which the different layouts will be tested is different for each of the participants and illustrated in Table 3.2.

Procedure

The following procedure was applied for each participant:

Participant	Layout order			
	1st	2nd	3rd	4th
1	D	C	R	S
2	D	C	S	R
3	D	R	C	S
4	D	R	S	C
5	D	S	C	R
6	D	S	R	C
7	C	R	S	D
8	C	S	R	D
9	R	C	S	D
10	R	S	C	D
11	S	C	R	D
12	S	R	C	D

D Default layout
C Column layout
R Row layout
S Snake layout

Table 3.2: Order of layouts for the different participants.

Introduction

- The goal of the experiment is explained, with the help of the questionnaire.
- The participant fills in the second part of the questionnaire for the learnability of the different layouts.
- The application is explained.
- The task he/she has to do is explained together with the interface.

For each layout

For each of the different layouts the participant gets a training task with the layout and the experiment setup. The training is followed by the tasks for the layout.

- Show a 30 second video clip
- Wait for the user to press start
- Show the layout with thumbnails, where the user has to find the correct location.
- The participant gets to know that we are going to start with the tasks.
- Then the 5 tasks start. Each task consists of:
  - Show a 30 second video clip
  - Wait for the user to press start
  - Show the layout with thumbnails, where the user has to find the correct location within 60 seconds.

Final questions

At the end, the participant is asked to fill in the last part of the questionnaire, verifying the subjective experience.

3.3.3 Findings

Qualitative ratings

The results from the questionnaire are broken down in three distinct categories, covering the same aspects evaluated in the cluster size experiment.

Fondness

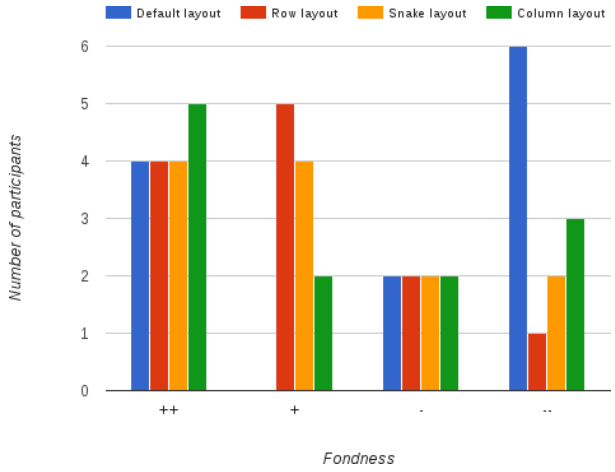


Figure 3.6: Fondness of the different layouts.

Figure 3.6 shows a chart of the answers given by the participants for the fondness of the different layouts. The answers are split per type of layout. There appears to be a small favor for the row and snake layout. Results for the default layout are inconsistent with one-third of the participants expressing a high preference, but also half of the participants demonstrating a strong dislike.

Quickness

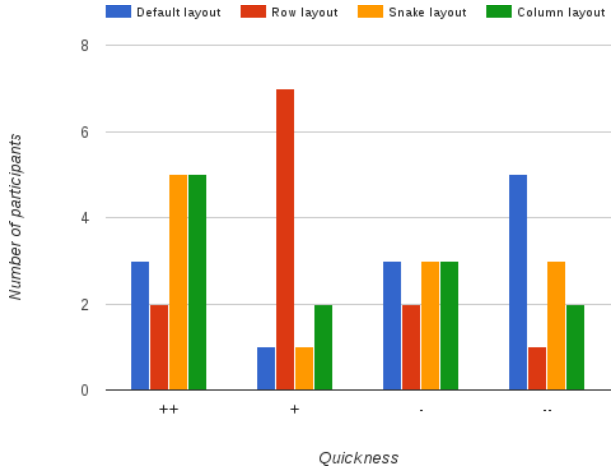


Figure 3.7: Perceived quickness of the different layouts.

Figure 3.7 shows a chart of the answers given by the participants for the perceived quickness of the different layouts. Again, the answers are split per type of layout. The row layout was perceived as relatively the quickest, whereas the default layout is perceived as the slowest. The column layout, which was used during the VSS2015 competition, was perceived in a mixed way.

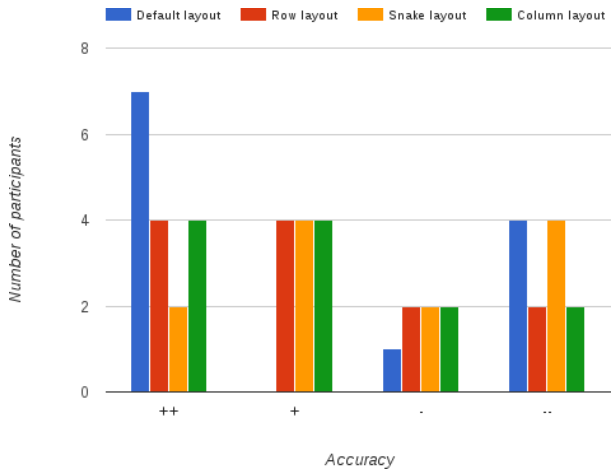


Figure 3.8: Perceived accuracy of the different layouts.

Accuracy

Figure 3.8 shows a chart of the answers given by the participants for the perceived accuracy. This describes how correct the participants thought they answered. The answers are split per type of layout. The perceived accuracy shows a favor for the default layout and a slight disfavor for the snake layout. Yet, one-third of the participants also gave lowest ratings for the default layout.

Quantitative performance results

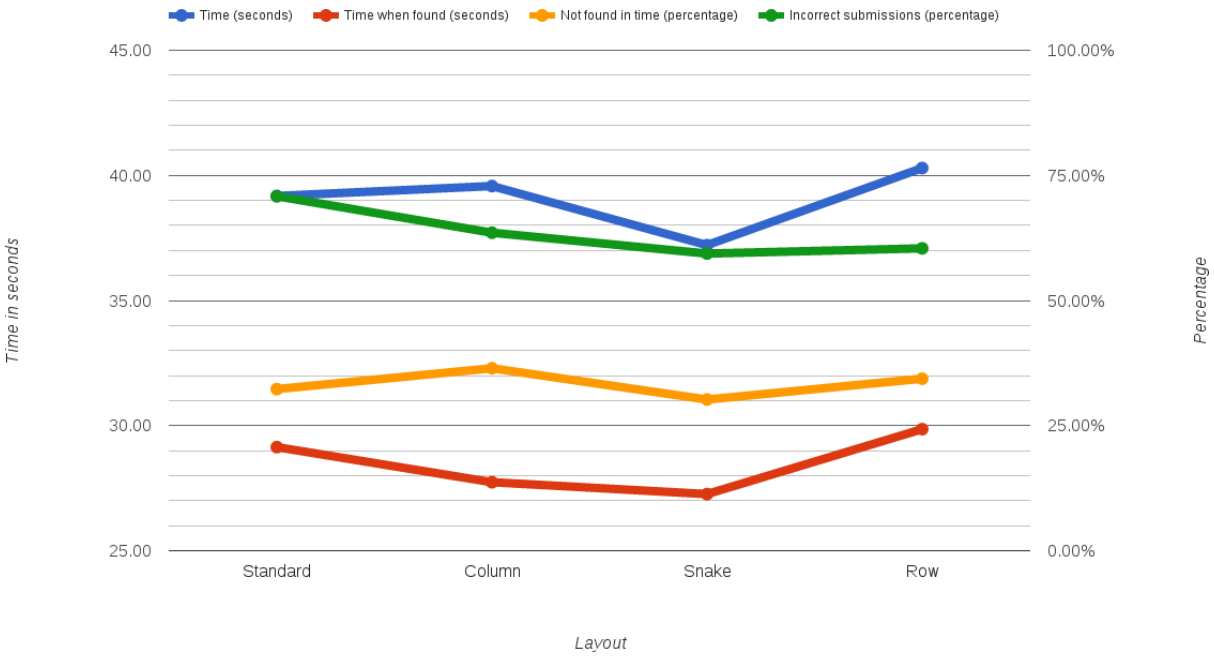


Figure 3.9: Layout averages

The quantitative results of the layout experiment are shown in Figure 3.9. The values are averaged over all the tasks and participants in the same way as described in the cluster size experiment (3.2.2).

The values show an advantage for the snake layout, by having the lowest values for all the averages. The snake layout is followed by the column layout, as this has the second best time when found. This observation is in contrast to the results of the cluster size experiment, since it suggest an albeit small yet noticeable benefit of a cluster design.

However, not all cluster designs will be beneficial, as becomes clear from the row layout, which has the worst values for the time when found. And it is not much better at finding the result within the time compared to the column and default layout.

3.3.4 Sub conclusion

This experiment compared the successful cluster design used in the VSS 2015 competition with the default, non-clustered layout and two further cluster versions. The quantitative results suggest indeed a benefit of cluster designs over the default layout, yet, the observed improvement was unexpectedly small. In addition, it was shown that not all cluster designs have a potential benefit, as illustrated with the low performance of the row-based cluster layout. Qualitative user statements also suggest a slight preference towards the good-performing cluster designs, although there was more diversity in opinions and not as clear trends as in the preceding cluster size experiment (cf. section 3.2). The performance measures achieved in this experiment seem to slightly contradict the ones from the cluster size experiment, which suggested no real benefit of a cluster design, whereas here, a slight improvement has been observed. Overall, it seems that a benefit might exist, but the actual improvement seems rather small, thus suggesting there was not much impact of the cluster design on the positive outcome of the VSS competition.

3.4 Scalability experiment

The scalability experiment is designed to gain insight into the speed at which humans can search within a screen of images. The time required to make a judgement, based on a screen of images, makes it possible to calculate how long it will take to browse a given number of screens. This in turn can give the insight if it is possible to browse an entire set of screens within a given time. For example, if it is possible to even browse the entire data set of the VSS2015 during a task.

3.4.1 Participants

The scalability experiment was done after the cluster size and layout experiments, but used the same participants. The participants from both the cluster size experiment and layout experiment performed the scalability experiment. Therefore there are 24 users participating in this evaluation, all of which have experience with the interface and searching in a video archive.

3.4.2 Setup

Application

An application is created for showing a video fragment, showing images of the layouts, doing experiments (based on a task-file) and storing the output for evaluation.

For each task the application will show the video fragment, wait for the user to become ready to search and then show the layout. In contrast to the preceding experiments there is no direct interaction with the data other than visually inspecting it. The layout progresses at a predefined speed through five screens, with each 625 images. When the last screen of the layouts has been displayed, the user is asked to answer if they think the shown fragment was within the layout, from the same video or neither.

The answers for indicating if the requested fragment was in het screens or not are obvious. However, the answer to indicate that the requested fragment is not in the screens but the screens are from the same video, indicate the person would most likely do a more thorough search during the actual competition. The more thorough search when the video fragment is either close or even in the screen is a positive response, as it will most likely result in a good performance during a contest.

Data set

A data set based on VSS2015 data will be used. The data set will consist of 52 separate tasks, put in 3 groups (correct, same video, different video). Each task will consist of a video fragment (which has to be found) and a set of 3125 thumbnails. Each task uses a different video, to prevent a learning of the data set. Each participant will use the same data set, but in a random order. Half of the group will have an increasing speed and half of the group will have a decreasing speed. This will avoid the difference in task difficulty influencing the performance.

The data set will consist of the following sets:

- 2 for training
- for each speed: 2 tasks with the video fragment within the screens
- for each speed: 1 task with the video fragment not within the screen, but the screens are from the same video
- for each speed: 2 tasks with the video fragment completely different from the screens

The lower bound for the duration (fastest) is 100 milliseconds per screen. Increasing the duration to an upper bound (slowest) of 12000 milliseconds per screen. This upper bound is the same as the time there was to search with the layout and cluster size experiments, where the participants had 60 seconds to search within five screens. The group with the increasing speed will have two speeds which are faster than the fastest of the decreasing speed group. However, the decreasing speed group has two speeds which are slower than the slowest of the increasing speed group. The distribution of predefined times are illustrated in Table 3.3.

Increasing group		Used by all participants								Decreasing group	
100	250	500	750	1000	1500	2000	3000	4000	6000	8000	12000

Table 3.3: Distribution of times used for the scalability experiment, in milliseconds.

Procedure

The following procedures describe, in a step by step manner, what a participant will be doing for the experiment.

Introduction

- The participant is explained what we are going to do and that we and the goal of the experiment.
- The application and the setup is explained.
- The participant gets to know the task he/she has to do. The interface is also explained.

Training

The participant has to do 2 tasks from the taskset for training (from the group of correct).

- Show “are you ready” button, wait for user
- Show a 30 second video clip
- Show “start” button, wait for user
- Show the five layout screens with a 12 second interval
- Show answer buttons, wait for user
- Show correct answer

For each different speed

The participant gets to know that we are going to start with the tasks. Then the tasks start. For each speed five tasks are selected, with two tasks from the group of correct, one from the group of same video and two from the group of incorrect.

- Show "are you ready" button, wait for user
- Show a 30 second video clip
- Show "start" button, wait for user
- Show the layout for a predefined time (in 6 cases the time increases, in 6 cases the time decreases)
- Show answer buttons, wait for user

3.4.3 Findings

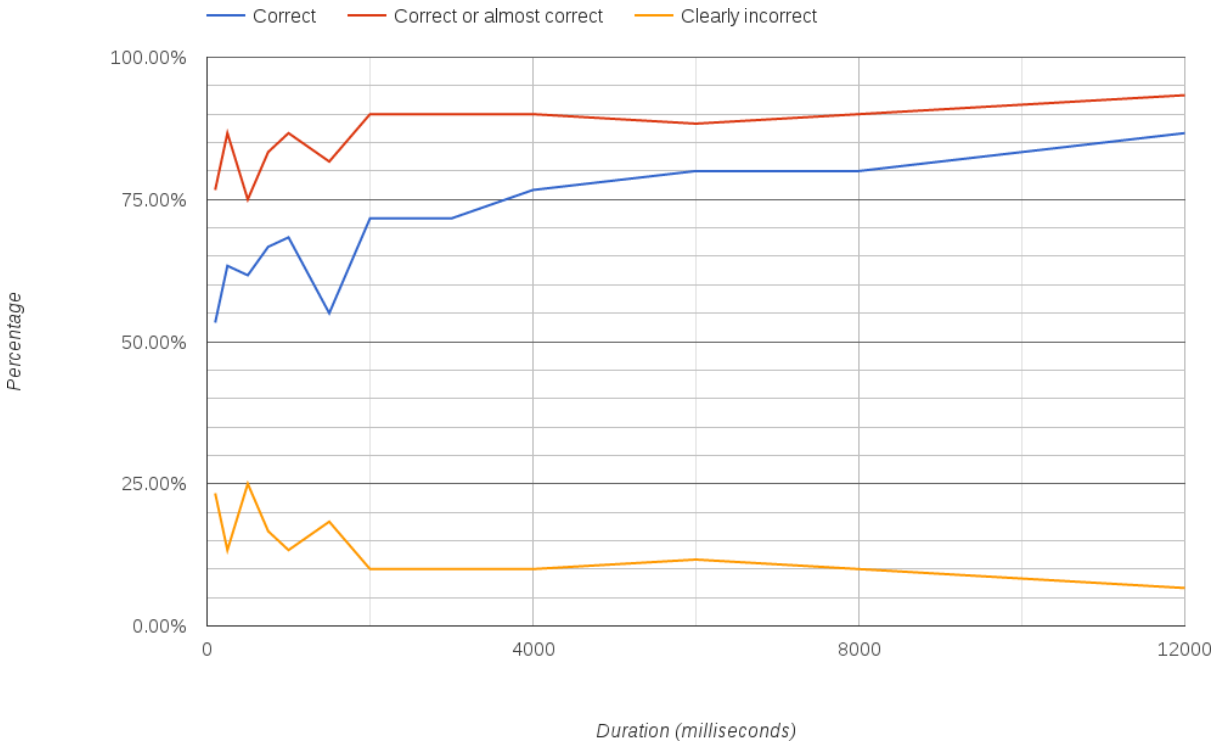


Figure 3.10: Correctness of answers given at designated speeds.

Figure 3.10 shows the averaged results of the performance of the participants at the designated speeds. As expected, as the speed increases, and the time for each screen decreases, the correctness is getting worse. At 1500 milliseconds per screen there is a drop in performance. A possible explanation for this outlyer could be a change in search behavior, ie. this drop in performance is caused by a change in search method. For the increasing speed there is a point where it is no longer possible to inspect all the images on each screen and where the participant will have to start looking at the screen of images in a different way, by only glancing at the images. For the group with the decreasing speed this likely also took place but in the opposite direction, because there is more time to view the entire screen of images instead of just glancing at the images and making a guess. Yet, there is no clear indication in the data where this might have happened.

The lines for the “correct or almost correct” and “clearly incorrect”, within Figure 3.10, are the opposite of each other. The “clearly incorrect” indicates the answer given is the opposite of the correct answer. The “correct or almost correct” answer combines the correct answers with the answers where a person would do a more careful search during the actual contest, because the correct answer is close. The clearly incorrect answers indicate a complete miss, which would result in having it unnoticed pass by or looking when it is clearly not there during the actual contest.

3.4.4 Sub conclusion

The aim of this evaluation was to find the limits of the human browsing ability and gaining insight into the scalability. A trend can be observed, where there is not much change in the performance when given 2000 milliseconds or more. However, there is a drop if the time to browse a single screen of 625 images gets below 2000 milliseconds. Yet, the drop in performance is not as strong as expected. But this is still good as people are still geting information from the screens, even when they are only showed very shortly. The saturation level, observed from the 2000 milliseconds and up, is not as strong as expected. However, it can be observed that people are able to grasp information quickly and make reliable decisions in many cases.

3.5 Group scalability experiment

3.5.1 Participants

The experiment is held within the "Mobile Devices" course, therefore 9 master students are participating. The participants have knowledge of mobile devices and are in the age range between 20 and 30. Some of the students have already got experience with searching within video archives in a similar way, because they also followed the "Multimodal Interaction" course, and therefore participated in the earlier experiments. The group is also joined by the teacher, which has advanced knowledge of the subject and experience with the tasks, as he also participated during VSS2015 as the expert of the system described in section 2.

3.5.2 Setup

Since this experiment is aimed at a more competitive setup, this experiment will not make use of the tablet device. This is because there is only one Nexus 9 available at the time. Therefore computers are used as an alternative, as those are are readily available.

Devices

During this experiment there are a number of devices required to perform the experiment. Firstly a computer with a beamer is used to show the video fragments which have to be found by the participants.

Secondly, each participant will use a computer with the layouts in which they have to find the video fragments. This device is also used to answer the questions for every task. The computers which the participants use are more than fast enough to show the interface with the set times. Each computer has the same screen, which is a 19 inch screen with a resolution of 1280x1024 pixels.

Application

Since a computer is used for the experiment instead of a tablet, like previous experiments, a different application is required. It is also necessary to have a separate server and client system to have a central control system. The server interface will show the videos that need to be found and a status overview of the clients. The clients wait for commands from the server and show the different layouts at the designated speeds. The clients will also ask the questions which the participants have to answer.

The system needs to be easy and fast to implement while still being able to handle all the clients in sync. The software on the server side was free to choose. But since the computers used during the experiment will need to be set up quick and easy, the easiest way to show a interface on the computers would be by using a webinterface with javascript for interactions. The clients will then communicate with the server using websockets. The easiest way to use websockets on a server is by using nodejs. This has also the advantage of using javascript not only on the client, but also on the server. The server interface will be the same as the client, but based on the login name the server can send the server-interface-actions to the browser. This has the added advantage of using multiple devices to control the server and having to develop only a single interface. The server uses json files to load the tasks and will use json to communicate over the websockets connection, as this is the easiest way of communicating using javascript.

Variables in the experiment

There are two variables tested in this experiment, namely the layout of the storyboard and the speed at which screens of images are visible.

Layout

The layout variable is comprised of two layouts, namely the default and column layout, which have been tested before in the layout experiment (section 3.3). They are tested by splitting the group of participants in two subgroups. These subgroups will perform the same search task but with different layouts. Group A will start with the default layout (as can be seen in Figure 3.1a). Group B will start with the column layout (as can be seen in Figure 3.1b). With the second round the groups will switch layouts, as can be seen in Table 3.4

	Round 1	Round 2
Group A	Default	Clustered
Group B	Clustered	Default

Table 3.4: Order of layouts tested by both groups.

Speed

The speed is tested by reducing the duration the screens with thumbnails are visible for each set of tasks. The same list of durations is used for both rounds, which makes the data comparable.

The list of durations that will be used to show each screen of images is illustrated in Table 3.5.

3000	2000	1500	1000	750	500	200	100

Table 3.5: Duration used for each set of tasks, in milliseconds.

Data set

A data set is created using episodes from several popular tv-series. The data set will consist of multiple separate tasks put in 4 groups (correct, same episode, same serie, different serie). Each task will consist of a video fragment of 10 seconds, which has to be found, and a set of 3125 thumbnails. The thumbnails will be preprocessed to create screenshots containing 625 thumbnails, similar to the contest and the earlier experiments. Each task uses a different video, to prevent the learning of the data set. Since all the participants will view the images that need to be found at the same time, the order of the data set will be the same for all participants, obviously.

The data set will consist of the following sets, for each speed:

- Two tasks with the content of the video fragment within the screens
- One task with the content of the video fragment not within the screen. The screens are from the same episode as the video fragment.
- One task with the content of the video fragment not within the screen. The screens are from a different episode, but the same serie.
- Two tasks with the content of the video fragment completely different from the screens.

Procedure

The following procedure was applied for the experiment:

Introduction

- The group is explained what we are going to do and the goal of the experiment.
- The application and the setup is explained, and an example is given.
- The participants do a few training rounds to get to know the application and to get a feeling of how to search.

Between the introduction and the actual experiment there will be a short break.

For both layouts

- For each speed
  - For each task
    - Show the video fragment, using the beamer.
    - Wait for participants to start the search.
    - Show the 5 screens with the layout for set duration.
    - Show question which has to be answered.
    - Short break.
  - Show question for user experience of the speed.
  - Short break.
- Between the layouts, there will be a long break.

Finally

At the end, the participants are asked to fill in the last question where the participants tell what their preferred layout is.

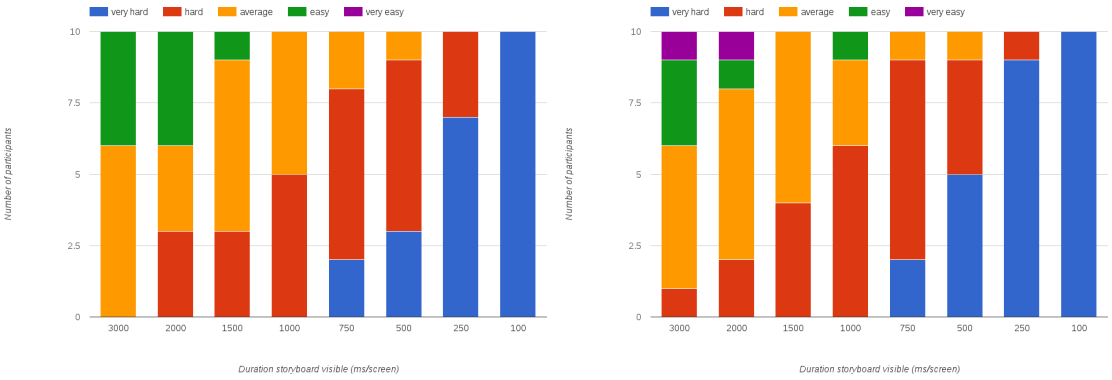
3.5.3 Findings

Qualitative ratings

Figure 3.11 shows the averaged results of the questions between the tasks. The default and the column layout show almost the same perceived difficulty. There is a small favor for the default layout, except for the two longest durations. There the column layout shows a small preference.

Quantitative performance results

The group experiment shows a worse performance than the previous scalability experiment (section 3.4), as can be seen in Figure 3.12. The overall correctness is lower than with the scalability experiment, and there is more variation in the lower speeds. This difference in performance could have multiple causes. The difference in the setup is most likely the cause, as the screens which the participants used has a lower resolution than the tablet used during the earlier experiments. Another difference could be the more competitive setup with the beamer and the shorter video fragments, where the shorter video fragments also result in a smaller section of the screen to be the part of the video fragment. Last but not least the



(a) Default layout (b) Column layout

Figure 3.11: Perceived difficulty for the 2 layouts.

Figure 3.12: Group scalability results

experience of the participants could have a big influence. The previous scalability experiment was performed by the participants after they had either done the cluster size experiment or the layout experiment, resulting in a learning effect on how they should be looking at the data.

Another observation is that there are three possible answers to give, resulting in a chance of 33% of guessing the correct answer. So it is interesting to see that the results are not much better than just guessing the answers. This is likely why the highest speeds have around 33% correctness. However, there is a strange peak in performance at the 500 milliseconds per screen. This peak could have multiple explanations. One of the explanations is the same effect as described for the scalability experiment, namely the change in search method by the participant. It is also possible that the peak is caused by the randomly selected input videos, in which case the videos could have been easier to identify.

Looking at the results in a similar way as the scalability experiment, by accepting the close answers as correct, does show a better result. However, this is to be expected as more answers are accepted as correct, and therefore it does not show any potential. It only raises the percentage by a nearly constant amount, as can be seen in Figure 3.13.

There is also not much difference between the performance of the two layouts used during this experiment. The default layout performed almost equal to the column layout.

Figure 3.13: Group scalability results, close as correct

3.5.4 Sub conclusion

As the results of this experiment are different from the scalability experiment and the performance of the participants is also worse, it is still remarkable to see what human-based searching is capable of. The participants indicated that the highest speeds are extremely fast and allow for only a glance at the images. This indicates that the upper limit at which the screens can be processed is most likely within the tested speeds. Yet, they were still remarkably well at giving correct answers, either by guessing of subconsciously answering correct. It is however also likely that the used screen has an influence on the performance of the user. Especially since the tablet device used during the VSS2015 competition and the earlier experiments has more pixels to display the images at a higher resolution while also being a smaller size.

4 Conclusion

The VSS2015 competition showed some real potential in the used layout and the speed at which humans can process large amounts of images. Reaching the third place in the VSS2015 competition shows an unexpected good result for the human based search method, in comparison to the computer based competition. Especially since this method had more correct answers than the competition.

The good performance was possible by optimizing the application as much as possible, and do as much as possible in the preprocessing of the data. This allowed the application to run smoothly during the VSS2015 contest and thereby allow the users to search at their own desired speed, however fast that would be.

There are three main conclusions that can be drawn from this contest and the used setup. Firstly, the created interface shows it is possible for a human to process a lot of visual information in a short time in order to search for a requested video fragment. Secondly, the experience with the interface and way of searching has most likely an influence on the performance of the user. Thirdly the results of the created system show that it scales much more than expected, especially for the amount of data that can be processed using it.

Because of the positive result of the contest, we could assume that this cluster size is indeed good, and maybe even the best one. The qualitative statements of the participants in our cluster size experiment seem to confirm this expectation, with participants generally expressing a preference towards this cluster size. However, the qualitative results show a good performance for cluster size 1, which represents the "non-clustered" representation, putting into question if there is any benefit of clustering at all.

The second, layout, experiment compared the successful cluster design used in the VSS 2015 competition with the default "non-clustered" layout and two further cluster versions. The quantitative results suggest indeed a benefit of cluster designs over the default layout, yet, the observed improvement was unexpectedly small. Qualitative user statements also suggest a slight preference towards the good-performing cluster designs, although there was a diversity in opinions. The performance measures achieved in the second experiment seem to slightly contradict the ones from the cluster size experiment. Overall, it seems that a benefit might exist, but the actual improvement seems rather small, thus suggesting there was not much impact of the cluster design on the positive outcome of the VSS competition.

The third experiment was aimed to find the limits of the human browsing ability and gaining insight into the scalability. A trend can be observed, where there is not much change in the performance when given 2000 milliseconds or more. However, there is a drop if the time to browse a single screen of 625 images is dropped below 2000 milliseconds. Yet, the drop in performance is not as strong as expected. But this suggests people are still geting information from the screens, even when they are only showed very shortly.

The results of the final experiment are different from the scalability experiment and the performance of the participants is also worse. However, it is still remarkable to see what human-based

searching is capable of. The participants indicated that the highest speeds are extremely fast and allow for only a glance at the images. This indicates that the upper limit at which the screens can be processed is most likely within the tested speeds. Yet, they were still remarkably well at giving correct answers, either by guessing of subconsciously answering correct.

All in all it seems that human-based searching is capable of performing really well, even compared to state of the art computer based search techniques. As long as the user interface and interactions allow the user to process the large amounts of data easily.

5 Future research

Although the studies explained our good VSS2015 results, they did not ultimately identify the optimal implementation. Hence, follow up studies on cluster design and scalability should be done in future work.

During the creation of the final application, a different possibility showed potential. This is the use of the fast-forward technique, as described in 2.3.3. This method showed some interesting side effects, where the amount of data on which the user has to focus is limited. But as the images that are going to be displayed on the point where the user is focussed are already visible next to it, the human brain can possibly already process the data subconsciously. This could allow for an extremely fast scrolling through the vast amount of data. But as the current research was looking more into the techniques as used for the VSS2015 implementation, this was not part of the current research. It did however show great potential during some simple testing.

Another way of improving the performance, or even make the system more scalable, is by adding a dynamic sorting of the list of videos. However simple this sorting would be (eg. through simple queries), this would make the system far more scalable. For VSS2016 such a system is being developed in cooperation with Klagenfurt University, where a search engine influences the data represented in the storyboard visualization. At the same time the data that has been processed by the human can be removed from the active data of the search engine, reducing the dataset in both systems as the data is being processed.

A different and obvious way of improving the performance and likely reducing the amount of images that have to be browsed, is by preprocessing the video differently. One example is the use of some sort of scene recognition, and showing limited set of images per scene instead of only using time-based image selection. This does however remove some contextual information, namely the length of scenes. It also reduces small changes within scenes to a limited amount of images. This could therefor be as much a benefit as a disadvantage.

6 References

[BS14] Werner Bailer and Klaus Schöffmann. Video search showcase (formely video browser showdown). http://www.videobrowsershowdown.org/, December 2014.

[HD12] Wolfgang Hürst and Dimitri Darzentas. Quantity versus quality: The role of layout and interaction complexity in thumbnail-based video retrieval interfaces. In *Proceedings of the 2Nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pages 45:1–45:8, New York, NY, USA, 2012. ACM.

[HSST11] Wolfgang Hürst, CeesG.M. Snoek, Willem-Jan Spoel, and Mate Tomin. Size matters! how thumbnail number, size, and motion influence mobile video retrieval. In Kuo-Tien Lee, Wen-Hsiang Tsai, Hong-YuanMark Liao, Tsuhan Chen, Jun-Wei Hsieh, and Chien-Cheng Tseng, editors, *Advances in Multimedia Modeling*, volume 6524 of *Lecture Notes in Computer Science*, pages 230–240. Springer Berlin Heidelberg, 2011.

[SB12] Klaus Schöffmann and Werner Bailer. Video browser showdown. *SIGMultimedia Rec.*, 4(2):1–2, July 2012.

[Sch14] Klaus Schöffmann. A user-centric media retrieval competition: The video browser showdown 2012-2014. *MultiMedia, IEEE*, 21(4):8–13, Oct 2014.

[SLZ+03] Nicu Sebe, MichaelS. Lew, Xiang Zhou, ThomasS. Huang, and ErwinM. Bakker. The state of the art in image and video retrieval. In ErwinM. Bakker, MichaelS. Lew, ThomasS. Huang, Nicu Sebe, and XiangSean Zhou, editors, *Image and Video Retrieval*, volume 2728 of *Lecture Notes in Computer Science*, pages 1–8. Springer Berlin Heidelberg, 2003.

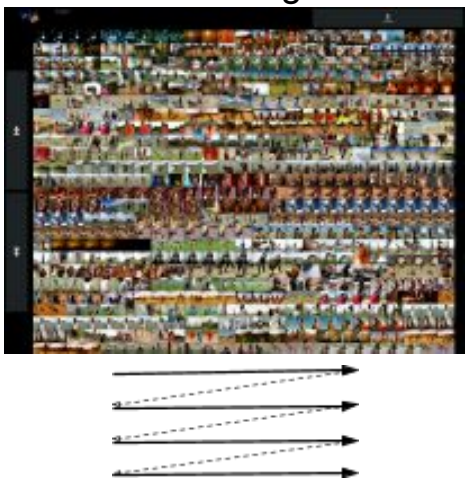
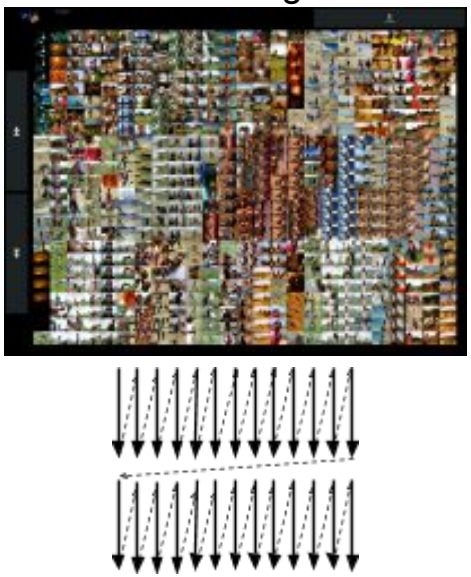
A Cluster size experiment questionnaire

This appendix contains the questionnaire as it was used in section 3.2. There were four versions of this questionnaire with each a different ordering of the cluster sizes. The participants would get the version with the order equal to the order in which they perform the tasks, as described in 3.2.2.

General information about the experiment

The goal of this experiment is to identify the fastest method for finding information in a video that is represented by a grid of still images, a so-called storyboard. These images can be put in different layouts. With those different layouts there are some variables, in this experiment the cluster size will be tested.

Below is a visualization of the clustering which is going to be tested. Make sure you see the difference between the different clusterings before continuing.

Clustering 1	Clustering 3	Clustering 5	Clustering 8
			

Experiment on device

You will be shown a video fragment, of 30 seconds, which you have to find in the storyboard. You want to be the fastest to find all fragments with the fewest mistakes. You will be guided through the experiment by on-screen instructions. First you will learn how the interface works.

Please start the with experiments on the device.

Concluding questions

	Clustering 1	Clustering 3	Clustering 5	Clustering 8
Which clustering did you like the best?				
Liked it a lot	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Didn’t like it at all	❑	❑	❑	❑
Which one do you think is best to solve the tasks as quickly as possible?				
Fastest	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Slowest	❑	❑	❑	❑
Which one do you think is best to solve the task as accurate as possible (ie. minimum number of wrong answers)?				
Lowest error	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Highest error	❑	❑	❑	❑

Participant number

Finally, get to know the participant

Age: ….
Gender: Male / Female / Other

Mobile device experience

Watch videos on mobile phones.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Watch videos on tablets.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Browse images on mobile phones.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Browse images on tablets.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Thank you for your time.

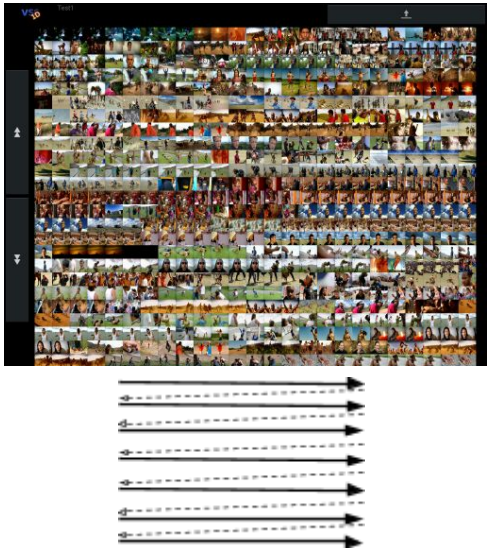
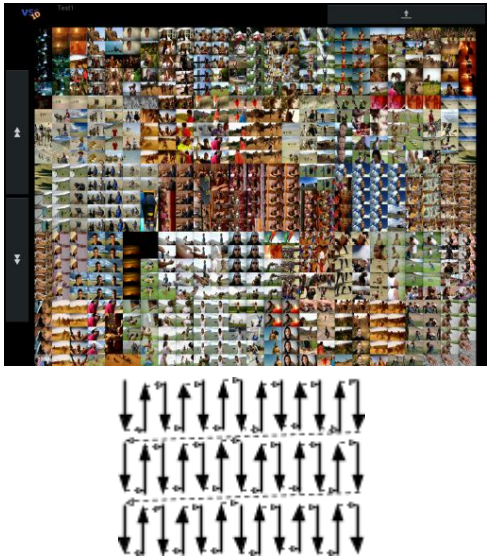
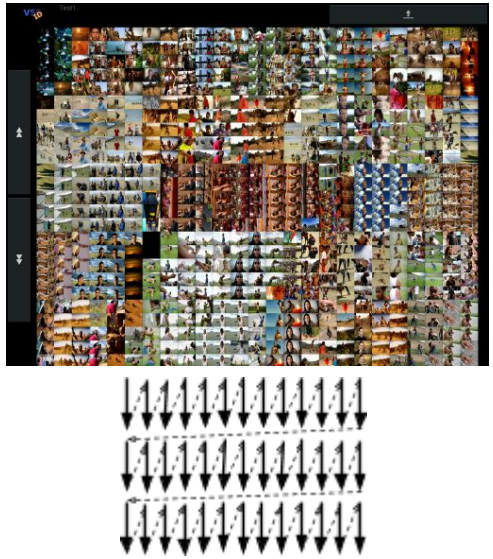
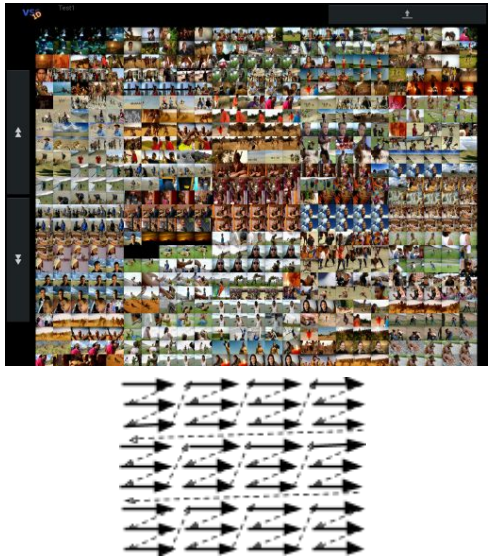
B Layout experiment questionnaire

This appendix contains the questionnaire as it was used in section 3.3. There were four versions of this questionnaire with each a different ordering of the cluster layouts. The participants would get the version with the order equal to the order in which they perform the tasks, as described in 3.3.2.

General information about the experiment

The goal of this experiment is to identify the fastest method for finding information in a video that is represented by a grid of still images, a so-called storyboard. These images can be put in different layouts. In this experiment the different layouts will be tested.

Below is a visualization of the layouts which are going to be tested. Make sure you see the difference between the different layouts before continuing.

Row based	Snake cluster	Column cluster	Row cluster
			

Experiment on device

You will be shown a video fragment, of 30 seconds, which you have to find in the storyboard. You want to be the fastest to find all fragments with the fewest mistakes. You will be guided through the experiment by on-screen instructions. First you will learn how the interface works.

Please start the with experiments on the device.

Concluding questions

	Row based	Snake cluster	Column cluster	Row cluster
Which layout did you like the best?				
Liked it a lot	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Didn’t like it at all	❑	❑	❑	❑
Which one do you think is best to solve the tasks as quickly as possible?				
Fastest	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Slowest	❑	❑	❑	❑
Which one do you think is best to solve the task as accurate as possible (ie. minimum number of wrong answers)?				
Lowest error	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
•	❑	❑	❑	❑
Highest error	❑	❑	❑	❑

Finally, get to know the participant

Participant number

Age: ….
Gender: Male / Female / Other

Mobile device experience

Watch videos on mobile phones.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Watch videos on tablets.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Browse images on mobile phones.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Browse images on tablets.

- ☐ Rarely
- ☐ Now and then, eg. monthly
- ☐ Often, eg. weekly
- ☐ Very often, eg. daily

Thank you for your time.

C Publication at VSS2015

The participation at the Video Search Showcase also meant a paper had to be submitted with the technique that was going to be used. The paper describes a different application than was actually used during the competition, due to the changes made to the competition by the organisation.

The application contains two different interfaces, one using a thumb-based interface on top a video player. The other is a standard vertical storyboard with vertical scrolling. This application can only handle a single video at a time, and therefore would not be able to compete during the VSS2015.

A Storyboard-Based Interface for Mobile Video Browsing

Wolfgang Hürst, Rob van de Werken, and Miklas Hoet

Department Information & Computing Sciences, Utrecht University, The Netherlands
huerst@uu.nl

Abstract. We present an interface design for video browsing on mobile devices such as tablets that is based on storyboards and optimized with respect to content visualization and interaction design. In particular, we consider scientific results from our previous studies on mobile visualization (e.g., about optimum image sizes) and interaction (e.g., human perception and classification performance for different scrolling gestures) in order to create an interface for intuitive and efficient video content access. Our work aims at verifying if and to what degree optimized small screen designs utilizing touch screen gestures can compete with browsing methods on desktop PCs featuring significantly larger screen estate as well as more sophisticated input devices and interaction modes.

Keywords: Mobile interfaces. Mobile video browsing. Interactive multimedia.

1 Introduction

The ubiquity of handheld mobile devices such as tablets combined with the increasing popularity of mobile video playback and the possibility to access larger video archives via fast network connections results in an increasing need for better interface designs for mobile video search and browsing. Yet, interaction design for such devices – especially for rather complex tasks such as quick and efficient video browsing – is difficult for several reasons. First, the devices' form factor results in limited screen estate, which in turn limits, for example, the ability to visualize a video's content (e.g., via storyboards) and meta-information about a video (e.g., text annotations). Second, the predominant input modes for such devices, i.e., touch and tilting actions (e.g., via touch screen and accelerometers, respectively) are often lacking the flexibility and accuracy of input devices commonly used in desktop PC environments (such as keyboard and mouse). While we can therefore not expect video browsing systems on mobile devices to achieve a similar performance as interfaces optimized for desktop PCs, scientific studies (e.g., [4, 5, 6]) as well as prototypes and concrete interface designs (e.g., [1, 3]) suggest that high video browsing performance can be achieved if such a mobile system is optimized for the task at hand and considers the presumably limiting factors in the interface design.

For example, in our preceding research, we evaluated how the size of thumbnails used to represent video content influences video search performance [5, 6]. Our results indicate that surprisingly small sizes are actually sufficient in order to achieve a high search performance, thus suggesting that the small screen sizes of mobile

X. He et al. (Eds.): MMM 2015, Part II, LNCS 8936, pp. 261–265, 2015.

devices might be much less limiting for the interface design than commonly assumed. Likewise, touch interaction has obvious disadvantages, for example, when it comes to entering content, such as typing a query on an onscreen keyboard that lacks the tactile feedback of its physical counterpart and utilizes valuable screen estate. They also often lack the accuracy of controllers or mouse interfaces in tasks that require precision and accurate placement. Yet, touch gestures have be proven to be very intuitive, efficient, and considering performance maybe even better than traditional interaction modes in situations where quick navigation of large amounts of content is required – a characteristic which can obviously be very useful for quick video browsing if related interactions and gestures are implemented appropriately. For example, in [4] we compared how a paged versus continuous navigation of storyboards via touch gestures influences video search performance, resulting in related guidelines for mobile video browsing interface design.

Encouraged by such promising results, we proposed two interface designs – one utilizing a filmstrip style visualization integrated in vertically mounted timeline sliders placed on the left and right side of the screen, and one with a storyboard design utilizing our previous related research results [4, 5, 6]. Both designs have been evaluated in a comparative study [3] illustrating their usefulness, but also demonstrating complementary strengths and weaknesses. Consequently, we propose a new interface integrating both concepts into one single design with the ability to easily switch between the two interaction modes. While our studies so far have verified the design's usability for mobile video search, it will be interesting to evaluate it in comparison to more complex desktop PC systems as part of the Video Search Showcase (VSS) 2015 event in order to gain more insight into how well mobile systems can perform compared to such traditional setups, to identify their potential and also possible boundaries. The interface that we present is based on the one for single video browsing introduced and evaluated in [3], and extended in order to also support parallel browsing in video archives of up to ten individual files, as specified in the tasks of this year's VSS competition.

2 Interface Designs

Figure 1 illustrates the storyboard-based interface design used in the comparative study in [3]. Thumbnails extracted from the video are temporally sorted and presented in a 5x5 grid layout that can extend to the top and bottom beyond the screen. Scrolling to parts of the video before or after the currently visible area is done via up and down gestures, respectively. In order to illustrate the location of the currently visible part within the whole video, a scrollbar-style icon is added to the right side of the screen.

Figure 2 shows the aforementioned filmstrip style visualization which appears when the vertically timeline slider on the right side of the screen is used. Compared to the traditionally used horizontal orientation of such a slider, the vertical placements on the left and right side of the screen enables easier access and operation when holding the device with two hands during interaction (cf. illustration on the left side of the figure), a design decision that was also utilized in the interface design presented in

[2]. In our case, the slider on the left side of the screen covers the whole content of the video, enabling quick access to a certain part of it if and only if the related position is mapped on the (rather short) slider timeline representing the whole length of the video. For longer videos, the slider bar on the right can be used, which illustrates only a fraction of the whole file thus enabling browsing at a finer granularity level.

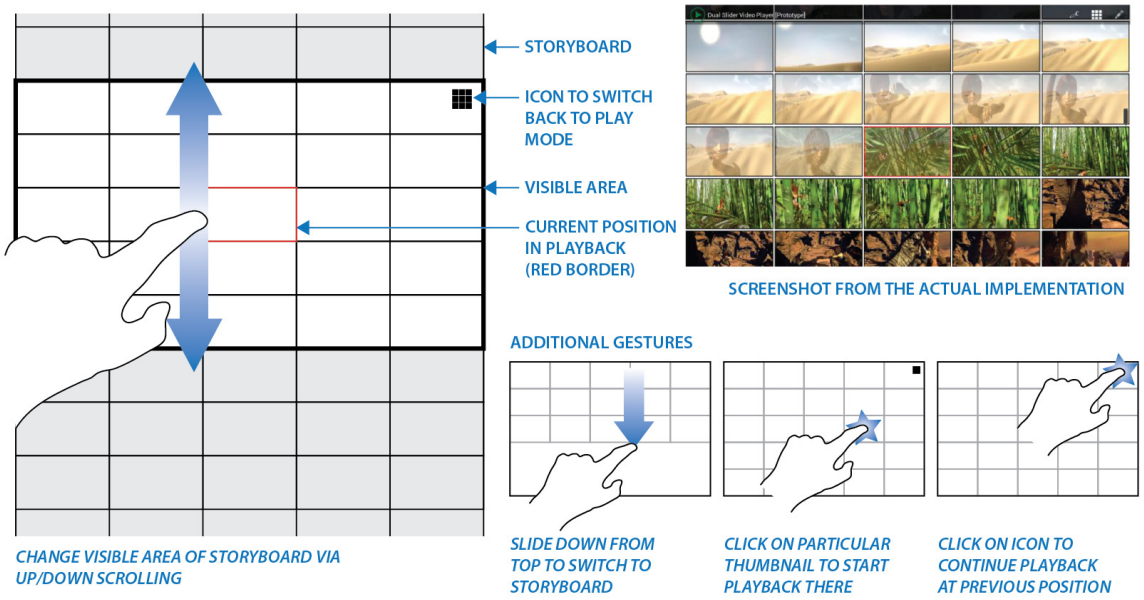


Fig. 1. Storyboard design implementation (from [3])

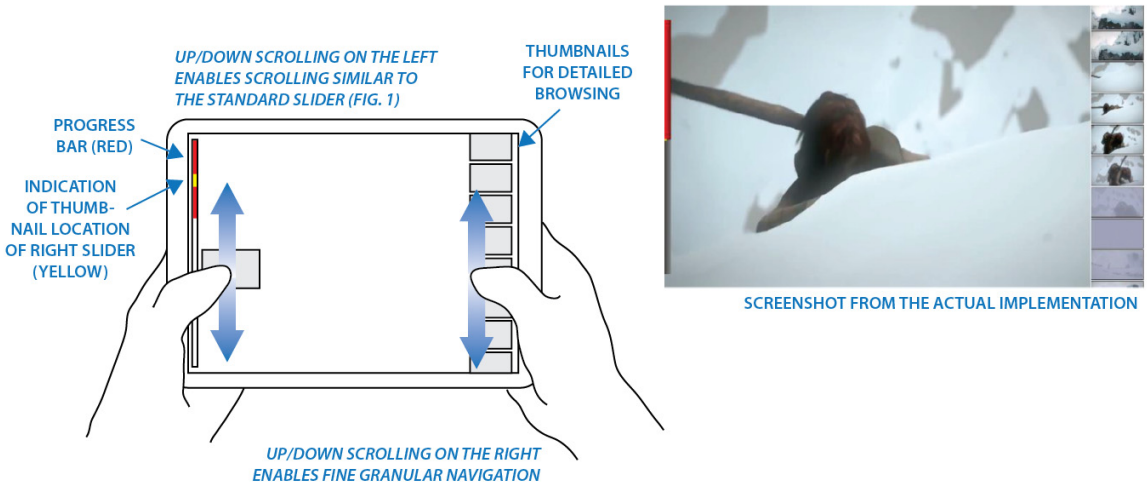


Fig. 2. Enhanced slider interface implementation (from [3])

In a comparative study using tasks slightly modified to the ones utilized in previous Video Browser Showdown competitions [7], both interfaces demonstrated their usability and power for video search (for detailed results we refer to [3]). Yet, both designs also revealed limitations and disadvantages – some of them opposed to each other. For example, the sliders obviously offer a faster access to searched locations if and only if those are directly accessible, whereas the storyboard design often

outperforms the slider interface when a more sophisticated inspection of the presented content is needed. Consequently, we propose a design that seamlessly integrates both interfaces, as illustrated in Figure 3, which we will present in the 2015 edition of the Video Search Showcase. Using gestures, users can easily activate either of the two scrolling modes (i.e., storyboard and filmstrip view) and switch between them. In particular, the storyboard view is activated by moving the thumbs of both hands slightly to the center of the screen, resembling an intuitive "zoom out" effect commonly used on tablets, for example, for maps where a comparable pinch-to-zoom gesture is also used to gain a higher-level overview of larger portions of the data. Clicking on a thumbnail in the storyboard activates playback mode again, where users can activate the filmstrip slider by simply clicking on the screen.

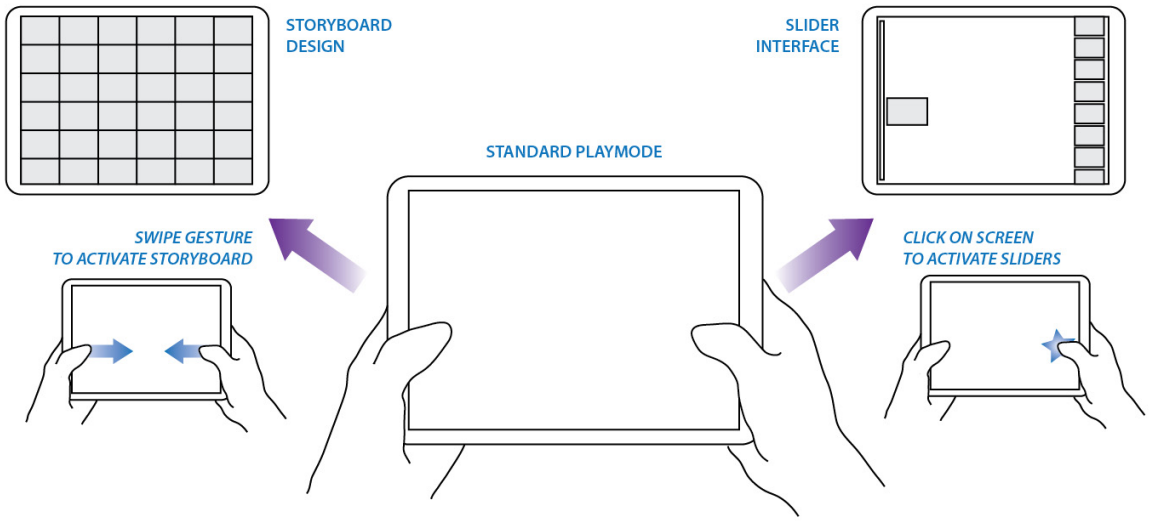


Fig. 3. Proposed design, seamlessly integrating both interaction concepts (from [3])

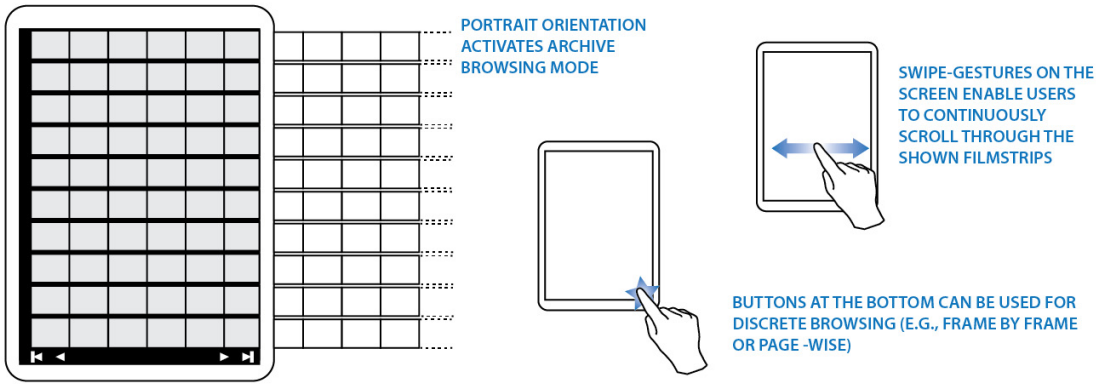


Fig. 4. Browsing video archives (ten videos in parallel) in portrait mode

Our studies confirm that this design enables quick and efficient video browsing within *one* video file and is thus well suited for tasks such as the Known Item Search (KIS) in single video files that was part of the Video Search Showcase in previous years. In order to deal with this year's tasks, which require search in ten videos from a larger archive, we propose the design illustrated in Figure 4. Turning the device from

landscape to portrait mode activates video archive browsing, i.e., the simultaneous navigation within ten video files shown as filmstrips. Navigating the content is done either by a simple left-right swiping gesture on the screen or by using buttons on the bottom of the display that result in a discrete motion. Both interactions enable a simultaneous movement of all filmstrips, so users can visually browse and inspect the content of all videos by just using these simple gestures. While our initial tests confirm that people are indeed able to simultaneously browse all videos with this approach, it should be noted that for larger archives than the ten videos used in this year's Video Browser Showdown obviously some sort of pre-filtering (e.g., via querying that creates a ranked list of video search results) is required, and part of our future research.

References

1. Cobârzan, C., Hudelist, M.A., Del Fabro, M.: Content-Based Video Browsing with Collaborating Mobile Clients. In: Gurrin, C., Hopfgartner, F., Hurst, W., Johansen, H., Lee, H., O'Connor, N. (eds.) MMM 2014, Part II. LNCS, vol. 8326, pp. 402–406. Springer, Heidelberg (2014)
2. Hudelist, M.A., Schoeffmann, K., Boeszoermenyi, L.: Mobile video browsing with the ThumbBrowser. In: Proceedings of the 21st ACM International Conference on Multimedia (MM 2013), pp. 405–406. ACM, New York (2013)
3. Hürst, W., Hoet, M.: Sliders versus storyboards – Investigating interaction design for mobile video browsing. In: Proceedings of the 21st International Conference on Advances in Multimedia Modeling, MMM 2015 (2015)
4. Hürst, W., Darzentas, D.: Quantity versus quality: the role of layout and interaction complexity in thumbnail-based video retrieval interfaces. Proceedings of the 2nd ACM International Conference on Multimedia Retrieval(ICMR 2012), article 45, 8p. ACM, New York (2012)
5. Hürst, W., Snoek, C.G.M., Spoel, W.-J., Tomin, M.: Size Matters! How Thumbnail Number, Size, and Motion Influence Mobile Video Retrieval. In: Lee, K.-T., Tsai, W.-H., Liao, H.-Y.M., Chen, T., Hsieh, J.-W., Tseng, C.-C. (eds.) MMM 2011 Part II. LNCS, vol. 6524, pp. 230–240. Springer, Heidelberg (2011)
6. Hürst, W., Snoek, C.G.M., Spoel, W.-J., Tomin, M.: Keep moving! Revisiting thumbnails for mobile video retrieval. In: Proceedings of the International Conference on Multimedia (MM 2010), pp. 963–966. ACM, New York (2010)
7. Schoeffmann, K., Ahlström, D., Bailer, W., Cobarzan, C., Hopfgartner, F., McGuinness, K., Gurrin, C., Frisson, C., Le, D.-D., del Fabro, M., Bai, H., Weiss, W.: The Video Browser Showdown: A Live Evaluation of Interactive Video Search Tools. International Journal of Multimedia Information Retrieval (MMIR) 3(2), 113–127 (2014)