VIDEOTREES: IMPROVING VIDEO SURROGATE PRESENTATION USING HIERARCHY

Michel Jansen, Willemijn Heeren and Betsy van Dijk

Department of Electrical Engineering, Mathematics and Computer Science
University of Twente, The Netherlands
{michel.jansen,w.f.l.heeren,e.m.a.g.vandijk}@ewi.utwente.nl

ABSTRACT

As the amount of available video content increases, so does the need for better ways of browsing all this material. Because the nature of video makes it hard to process, the need arises for adequate surrogates for video that can readily be skimmed and browsed. In this paper, the effects of the use of hierarchy in a pictorial summary of keyframes are explored, and a novel type of video surrogate is presented: the VideoTree. Moreover, a prototype browser was developed and tested in a preliminary usability study. This showed that users performed better using the VideoTrees browser than using a regular storyboard-based browser. They also found it more flexible, but more difficult to use.

1. INTRODUCTION

The Internet is no longer just for text. With broadband Internet connections becoming more ubiquitous, the amount of rich media such as video available on the Internet is becoming ever larger. With the amount of video material increasing, so does the need for an effective way of searching, navigating and browsing through this material.

Because video can visually express many concepts that would be hard to capture unambiguously in words, searching by means of a keyword-based query as in search engines for text will not always work. Alternative approaches to searching exist, such as ‘query by example’ [1], where an example image or video provided by the user is used as a search basis. Unfortunately, they too suffer from a *semantic gap*: low-level visual features may not correspond to higher-level semantic visual concepts [1, 2]. As concluded by Lew et al.: “We should focus as much as possible on the user who may want to explore instead of search for media” [3]. This research therefore focuses on navigating and browsing video.

Even if a search query leads to only a small number of results, the user is faced with the task of going through each video to judge its relevance. This can be difficult and time-consuming, because video is temporal and linear in nature. This is in contrast with text, which is instead a spatial medium that can therefore easily be skimmed [4]. Since it is in most cases not feasible to watch several hours of video to find a relevant section, the need arises for an adequate surrogate that can abstractly represent video in a way that is easier to process, and therefore makes assessing a video’s relevance less time-consuming.

Extensive research has already been performed to come up with adequate surrogates for video material, e.g. [5, 6, 7]. There are also already a number of different ways for automatically generating sets of keyframes for inclusion in a video surrogate, e.g. [8, 9, 10, 11]. However, a lot of work is still to be done in optimising the visualisation and representation of keyframe sets, especially when keyframe sets become large [12]. This research will therefore assume the existence of methods for acquiring keyframe sets and focus on the presentation. We aim to improve the user’s experience and performance while browsing video, by providing better visual abstractions as surrogates for the actual video.

One promising approach to visually presenting keyframe sets is by introducing hierarchy to control the amount of required screen real estate and consequently the amount of cognitive effort required to process the surrogate. Users can then start exploring video on a high, abstract level and ‘drill down’ to more detail. By testing a prototype that incorporates these principles, the following question was investigated: *What are the effects of hierarchical presentation of keyframes in video surrogates on user performance and satisfaction?* We hypothesise that by using hierarchy when abstractly representing video, users can find what they are looking for in a video more easily and with a more pleasant experience.

This paper starts with a short review of methods and tools required for creating video surrogates in general, and hierarchical video surrogates specifically. Then, hierarchical data visualisation and hierarchical video browsing are explained. Subsequently, a prototype hierarchical video browser using VideoTrees, a novel video surrogate, is presented. Finally, the evaluation results of this prototype are given and discussed.

2. CREATING VIDEO SURROGATES

In order to make videos easier to browse, coming up with an adequate surrogate or summary for that video is essential.

The research leading to this document has received funding from the European Community’s Sixth Framework Programme (FP6-027685-MESH).

978-1-4244-2044-5/08/$25.00 ©2008 IEEE.

Such a surrogate should be less complex than the original, while retaining as much of its informational value as possible. Ideally, any summary should have the following four properties [4]: (i) *Conciseness*, be as short as possible; (ii) *Coverage*, contain all relevant information; (iii) *Context*, select and present information such that its context is preserved; and (iv) *Coherence*, make the flow of information fluid and natural.

There are a number of approaches for creating video summaries and for any of these approaches, the original video has to be segmented in some way, to select relevant sections for inclusion in the surrogate. In the rest of this section the different types of video surrogates are discussed, followed by techniques for segmenting the video.

2.1. Types of video surrogates

There are roughly two approaches for creating video surrogates [12]: temporal and spatial summaries. Temporal summaries compress a video into a much shorter representation and are more commonly referred to as *video-skims*. Spatial summaries spread the video's contents over a two-dimensional or three-dimensional space and are often called *pictorial summaries*. Both types of summaries will be discussed briefly.

One way for creating an abstract video surrogate, is by literally making a shorter version of the original; a temporal video-skim. Such a summary can provide its viewer with a "fast forward" skim through the original, while taking a lot less time to watch. By selecting the most salient segments of a video and placing those in sequence, a movie-trailer-like result can be obtained [13].

Because a video-skim has the same modality as its original, it can retain many of its properties, including its expressiveness [12]. The temporal order of events can be preserved, as well as any audio present in the original, so it is easier to preserve context and coherence. This means that video-skims can provide viewers with a good gist of the original video [7]. Also, since the audio information is preserved, video-skims are generally good at summarising material with little informational value in the visual channel, but a lot of information in the audio, such as presentations and lectures [4]. For such cases, automatically generated summaries have shown to perform very close to summaries produced by human experts [4].

However, in retaining the modality of the original video, a video-skim also retains its disadvantages. The surrogate is still temporal and linear in nature, and requires a user to watch it to gain information from it. Even if the original video's length is reduced to a fraction of its duration, this may still be quite long. Furthermore, since shortening the original video almost inevitably means throwing away information, increasing the conciseness of a video-skim comes with the trade-off of losing coverage and coherence [12]. Because of these disadvantages, video-skims are not very flexible for browsing or navigating a video. Therefore, they are not very suitable as stand-alone surrogates for such a case.

Another approach for creating an alternative representation of a video is by spatially laying out its content in a spatial pictorial summary. By placing keyframes together on the screen, optionally enhanced with additional content such as textual transcripts, a pictorial summary or storyboard can be created. Although such a static storyboard does not retain the same amount of expressiveness as its video source, because motion and sound are lost, its spatial rather than temporal nature gives the viewer an overview of the video 'at a glance', without having to sequentially go through the video [10].

Moreover, the spatial approach allows the keyframes to be used as an index to the video. In existing systems, such Boreczky's manga summaries [14] and the CueVideo system [15], clicking a keyframe makes the player automatically jump to the desired fragment. Because of this property, and because users prefer spatial surrogates over temporal ones [7], pictorial summaries are often used in video retrieval systems.

Unfortunately, spatial summaries suffer from drawbacks of their own. Eventually, the amount of space or screen real estate available is a limiting factor for any pictorial summary. If a user is to quickly grasp the contents of the surrogate, the amount of keyframes in the summary has to be kept under control. However, reducing the number of images also reduces the level of detail [12]. As a consequence, there is a trade-off between conciseness and coverage.

It goes beyond the scope of this paper to cover all approaches to this problem, but there are roughly two ways to improve the coverage of a pictorial summary. On the one hand, increasing the salience of the images selected for inclusion in the surrogate by means of a good segmentation method, makes that surrogate more representative of its source video [8, 16]. On the other hand, improving the layout of the selected keyframes by adding clues to the relations between and the importance of represented segments, increases the information value of the summary [14, 10]. As we will see later, the approach used by VideoTrees is based on a combination of these two factors: a semantic spatial layout combined with semantic video segmentation.

2.2. Video segmentation

For any type of video summary, a selection of which parts of the video to include in the summary has to be made. For a video-skim, this selection consists of a set of video segments, for a pictorial summary it is a set of keyframes. The process of extracting relevant segments or keyframes can be performed at many levels, each of which are described briefly here.

At its most fine-grained level, any video consists of a number of *frames* placed in sequence [9]. For the video to appear smooth to the human eye, one second of video generally consists of 24 to 30 frames [17], which means that a mere 5 minutes of video would result in at least 7,500 images. For specific applications like video editing, this may be suitable, but for most applications the amount of frames will be too high [8].

More often, a *shot* is considered the smallest building block of video. A shot can be seen as the sequence of continuous action from the start to the end of a single camera operation [10]. Segmenting a video into shots can be done automatically using shot boundary detection [18]. On average, most video types were observed to have about 200 shots for 30 minutes of video [10]. Considering the screen space available on most modern computers, however, this many images can still not be displayed without resizing them to very small dimensions or requiring the user to scroll.

A level of granularity that is yet coarser than the shot level, is that of scenes. The cinematographic definition of a scene is *"a subdivision of an act of a play in which the time is continuous and the setting fixed and which does not usually involve a change of characters"* [19]. Automatically detecting scenes is more difficult than detecting shots, because it generally involves the extraction of higher level visual features, such as the background setting [1]. In the video material used for this research, an average of 20 scenes were found in a 25 minute documentary. For some types of video, even higher conceptual levels exist, such as clusters of scenes grouped by semantic similarities [11]. For example, a news program consists of a number of news items, which in turn may be grouped by subject into politics items, sports items and so forth.

Using segments extracted from various levels of granularity, a *concept hierarchy* [11] can be built. This concept hierarchy will be one of the foundations of the VideoTree design.

3. HIERARCHICAL VIDEO BROWSING

Based on the concept hierarchy just mentioned, video can be seen as a hierarchical data structure. This allows for the application of existing techniques for visualising such structures. Before the VideoTree hierarchical browser is introduced, some background on hierarchical visualisation is given as well as examples of video browsers that include a notion of hierarchy.

3.1. Hierarchical visualisation

Considering the video concept hierarchy as a regular hierarchical data structure, techniques for representing and navigating hierarchical data can be applied to it. In this section, two major issues for any interactive graph visualisation are discussed: layout and navigation [20].

To begin with layout, the main challenge in drawing any hierarchical graph is viewability [20]. Given a limited amount of space, all nodes in a graph have to be drawn, while keeping them discernible from each other and keeping the relations between the nodes intact. Herman et al. distinguished a number of popular graph layout techniques [20].

A classic *tree layout* positions each of a node's children nodes below their parent. In this way a classic tree has a clear 'top down' direction, and it is easy to determine any node's position in the hierarchy, but it also suffers from space inefficiency. Because the 'breadth' of the tree grows with each level of the hierarchy, the graph gets exponentially wider or more cramped. Variations, such as the *radial trees* and *cone trees* exist to counter this disadvantage, at the cost of being harder to comprehend or requiring a 3D view to be understood. Another visualisation method is that of *tree-maps*. Tree-maps differ from the previous methods in that they do not draw the nodes of a tree connected by edge lines, but instead represent trees as sequences of nested boxes. Because of this, tree-maps are very space-efficient. A drawback is that the structure of the tree is difficult to perceive.

The second challenge of interactive graph visualisation is navigation. In an interactively navigable hierarchical visualisation, it is important that it is visible which node has focus, and which nodes are related to that node. Using good navigation techniques, even trees that would normally be too large to fit in one view can be made accessible [20]. Two techniques that are traditionally very important in graph visualisation are zooming and panning [20]. By zooming in on a subsection of a graph, details that would be too small to be visible in the entire graph can be revealed. Panning refers to the action of moving the zoomed-in view on a graph around.

A well-known problem with zooming is that in a zoomed-in view contextual information is easily lost [20]. One solution often used, is including a thumbnail 'navigator' view with a smaller rectangle inside to represent the current viewport [20]. Alternatively, some techniques have been developed to preserve the context while still being able to focus on a subsection of the graph. These techniques have been called *focus+context*. One very powerful example is the use of *hyperbolic geometry* [21]. By drawing the graph in hyperbolic space and projecting it onto a circular display region, a distortion resembling a fish-eye effect occurs. Anything near the centre of the circle, the focus, is magnified. The surrounding space is distorted and compressed towards the sides of the circle, yet still visible. By combining zooming and panning with focus+context techniques, users can navigate a map quickly, while retaining a sense of the context around focused nodes, at the cost of distorting the graph view.

3.2. Hierarchical video browsers

A number of video browsers already use the concept of hierarchy to improve the usability or informational value of their content. For example, the Hierarchical Video Magnifier allows the user to select a range on a time line to have a storyboard view pop out, and show a graphical storyboard corresponding to the selected time frame [22]. This storyboard in turn has a time line of its own allowing the magnification step to be repeated. The keyframes in each storyboard are linearly sampled from the source video.

Video Posters [10] allow the user to click on a pictorial representation of a video segment to 'drill down' to its contents. At the top level, a video consists of 'stories' represented by collages of keyframes. When a user zooms in on a story, a list of keyframes for that story is displayed. The ClassView system uses hierarchical video shot classification to generate a segmented view of different levels of semantic concepts for inclusion in a storyboard view [1]. Similarly, the hierarchical movieDNA system clusters segments by semantic concepts, but displays them differently [23]. In hierarchical movieDNA, segments are depicted by dots on a strip, as in a DNA readout, which can be 'brushed' with the mouse to drill down in a pop-up. A final example of a video browser that uses a video's conceptual hierarchy for drilling down, is the manga representation by Borezcky et al. [14]. By displaying the contents of a video as an interactive comic book, their system was preferred by users over a storyboard-based alternative.

3.3. The VideoTree Hierarchical Browser

In this section we propose VideoTrees, a novel way for representing the contents of a video, along with the VideoTree Hierarchical Browser.

The concept of VideoTrees is based on visualising the conceptual hierarchy of a video in a pictorial surrogate. By extracting keyframes from each level of the source video, and placing them correspondingly in a tree, a hierarchical representation of the video's contents is created, where each level of the tree contains more keyframes and consequently more detail. By representing each node in the hierarchy by a keyframe, and by placing the images adjacent to their parents and siblings, there is no need for edge lines to indicate relations between nodes. The tree layout is a combination of the classic tree layout and the tree-map layout: every child is placed below its parent, yet the collective width of the children is restricted to the width of the parent. This results in a composition similar to the one in Fig. 1.

For simplicity, the aspect ratio of each frame is preserved. This means that all nodes are treated equal, even if their duration differs significantly. In other words, the information about the duration of a node is not encoded in the resulting composition. Preserving the aspect ratio of the frames also has the consequence that if any part of the tree is unbalanced, nodes of the same level may not remain vertically aligned with nodes of a different parent, as visible in Fig. 2.

Remembering the four properties of a good summary mentioned in Section 2, VideoTrees have many desirable properties for a video surrogate. First, the temporal order of the segments is preserved. Moving from left to right in the tree means moving forward in time. Therefore, the resulting surrogate is very coherent: it keeps a close relation to the video and can be seen as an alternative to the 'slider' control present in most video players. Second, because a VideoTree is built as a hierarchy of sub-trees, the tree as a whole can contain a lot of detail. The level of coverage is very high, while nodes on each level in the tree still contain a limited number of children and can therefore remain concise. Depending on the branching factor of the tree, the area of an entire VideoTree can easily be contained to a region with the width of the root frame and twice its height. The screen usage is therefore very efficient.

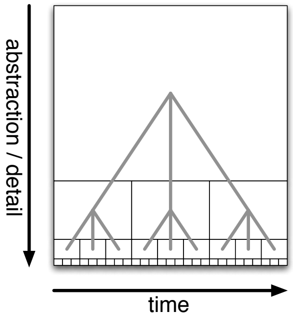

Fig. 1. The VideoTree presentation style.

Contrary to the visualisation methods used by the video browsers mentioned before, a VideoTree represents the entire video *in a single composition*. No information is hidden behind mouse clicks or pop up windows. A VideoTree contains keyframes for each level of the concept hierarchy, up to as much detail as the shot level. For a user to be able to reach this level of detail, a good method of navigation is crucial. The VideoTree browser, which is described next, is a first attempt at creating such a browser.

The most challenging part of creating a browser based on VideoTrees is allowing the user to browse the tree to see the details in the lower branches, without getting lost. This challenge can be reduced to three questions a system should answer for the user at any given time [23]: (i) *Where am I?*, (ii) *Where can I go?*, and (iii) *Where is X?*. To support the user in browsing the surrogate, navigation was implemented by making the VideoTree dynamic. At any given time, the node that has the user's focus is centred. Fig. 2 is a screen-shot of a typical situation where one node has the focus. The adjacent segment nodes on the same level are shown on the left and right, the parent node is shown above the focused node, and all children with respective sub-trees are shown below.

Clicking any node adjusts the focus by panning and zooming, so the focused node is centred on the screen. Changing focus between nodes on the same level results in a panning motion. If the clicked node is on a higher or lower level, the view will zoom out or in as needed. All motions are smooth so it is clear what is happening. Clicking a node also results in the playhead position of a video player, placed on the side, to jump to the time index corresponding to the node. The position of a user in the video is always a combination of the focused node, which is centred in the navigator view, and the playhead position, which can be determined from the slider.

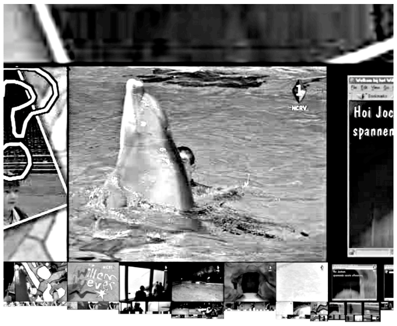

Fig. 2. Part of the VideoTree user interface while browsing.

To show the user his options during navigation, the mouse cursor changes upon hovering over any node. As can be seen from Fig. 2, there are four directions: up, down, left and right. The context to the left, right and up is limited. There is always at most one sibling node on each side, so the user can only pan one segment left or right at a time. For going up there is usually also only one option: the parent of the selected frame, except for edge nodes, where diagonal movement to an 'uncle' node is also possible. For 'drilling' down, full context is available: all nodes below the one selected, including their children, can be clicked for focus. Due to time limitations, a 'navigator' view or focus+context distortion were not implemented in the browser prototype.

4. USABILITY STUDY

In this preliminary user study, a prototype of the hierarchical VideoTree browser was compared with a regular storyboard-based video browser, which served as a baseline.

4.1. Methodology

To test the user performance and satisfaction of the VideoTree browser, a within-subject laboratory experiment was conducted. Participants were presented with a number of search tasks, which they were asked to perform using the VideoTree prototype described in Section 3.3 and a prototype with an identical layout, yet with a flat 'storyboard' overview instead. User satisfaction was measured using a questionnaire before, during and after the performance tests. The study was conducted with a group of 15 students, with an average age of 22, and with various levels of previous experience with video browsing. None of them had worked with VideoTrees before.

4.1.1. Material and test set-up

Both the VideoTree prototype and the storyboard prototype were configured to operate on a different episode of a Dutch educational program taken from the TRECVID 2007 set. It was segmented into shots using the reference shot-boundary indices provided for TRECVID [24].

The shot segments were manually grouped into scenes, based on changes in the scenery or setting. The resulting scene segments were grouped into logical semantic segments. In each episode of the program, a number of questions are treated and transitions between these questions are indicated by a short cut-scene of the program's logo and tune. By manually segmenting the video on occurrences of this cut-scene, five semantic segments per video were obtained. Two segments for both the intro and the outro of the episode, and three segments dealing with the questions. Finally, because the number of shots per scene turned out to be high compared to the number of scenes per item and the number of items per movie, an extra level of segmentation was added. By grouping the N shots of each scene into $\sqrt{N}$ clusters of size $\sqrt{N}$, a shot-cluster level was placed between the shot level and the scene level.

Since only the VideoTree prototype used all of these levels of abstraction, the complete segmentation was only applied to one episode. This resulted in 5 semantic segments, 16 scenes, 58 shot-clusters and 262 shots. The storyboard browser only made use of the 23 scene-level segments extracted from its episode. For each of these segments, the first keyframe of the segment was taken as a representation. The duration of the material used in the VideoTree and the storyboard-browser was 25:33 and 24:05, respectively.

For the usability study, both the VideoTree browser and the storyboard browser were implemented in Flash and loaded in a Web browser (see Fig. 3). The user interface of both prototypes was built up of two components: a video player, on the right, and a browser pane, on the left. Both prototypes were kept identical except for the layout of the contents of the browser pane. Both browser panes had a size of 640 by 720 pixels. In the case of the VideoTree browser it contained a navigable VideoTree. In the 'flat' prototype it contained a static storyboard view. A click on one of the keyframes in either browser pane resulted in the video player jumping to that position in the video and starting playback from there. The video player was sized to display a 352 by 264 pixels video, as is common for many on-line video applications. It was given a horizontal slider as its only method for control, to prevent users from resorting to controls they were already familiar with. Actions on the slider had no influence on the browser pane in either prototype.

For time registration, a button labelled 'Register' was included in the interface of both prototypes. It was used solely for tracking when users started and completed each task.

4.1.2. Measuring performance and perception

Each participant performed search tasks with both the VideoTree prototype and the storyboard browser. To prevent order-effect biases, the order of the prototypes was balanced across participants. For each of the two videos used in the task performance test, four questions were developed. For symmetry,

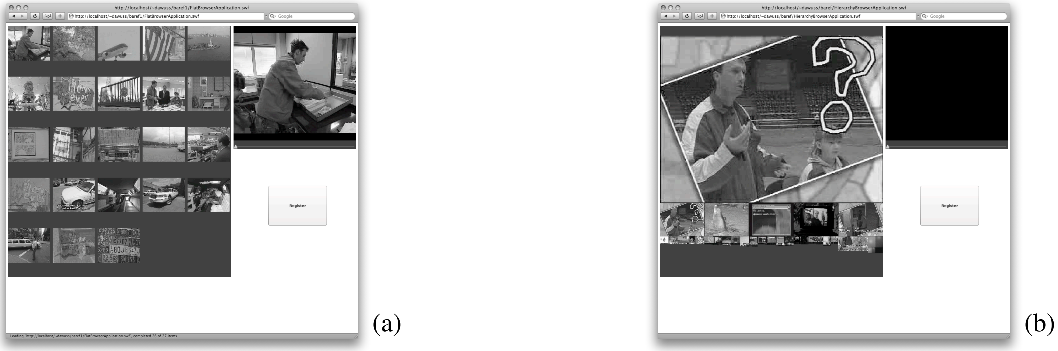

Fig. 3. Screen-shots of the ‘flat’ Storyboard browser (a) and the Hierarchical VideoTree browser (b) used in the usability study.

the nature of these questions and their answers were chosen to be as similar as possible. For instance, the answers to the first question for both prototypes were located about 20 seconds from the start of the second scene of the second semantic block. The questions were presented in order and one at a time. Only after a participant had completed answering one question, he was allowed to continue with the next question.

Before working with each prototype, participants were told they would be given four questions, all of which were answered in the video. They were asked to find the fragments where the answers were located as fast as possible. There were no penalties or rewards involved, and participants were told they could do anything they wanted with the prototype. Since each prototype had only been prepared to work on one video, users could not be given any time to explore the prototype in advance, because they might already find the answers to the questions they would be asked. During the tasks, two performance variables were measured: the *total time taken to complete all tasks*, and the *percentage of video watched*.

Another important question is whether hierarchical presentation influences user satisfaction. To answer this question the participants completed four short questionnaires. The first was presented before the tests, asking some general demographic information. After working with each prototype, participants were asked to rate both the ‘flat’ and the hierarchical prototype on a number of factors. Since we are testing a novel approach to video browsing, it is important to know if users are willing to accept this new kind of browser. The Technology Acceptance Model (TAM) [25] defines two variables that are of influence on the acceptance of any new technology: *Perceived Usefulness* and *Perceived Ease of Use*. Since browsing is essentially a way of controlling what information is displayed, the *Perceived Level of Control* also influences a user’s satisfaction with a given video browser. For each of these variables, a number of subjective terms, based on those used in the Questionnaire for User Interface Satisfaction (QUIS) [26], were included in the questionnaire. These terms are given in Table 2. Each of these terms were measured using a semantic differential in the form of a question containing a five point scale with the negative and the positive adjectives anchored on either side. To prevent ‘response sets’, the position of the positive and negative labels of some questions were reversed.

Table 1. Task performance results.

Variable	Storyboard	VideoTree
Time spent (s)	553.5	487.4
Percentage watched*	33.7%	26.3%

Table 2. Mean results of the user satisfaction questionnaires.

Question	Storyboard	VideoTree
Flexibility*	2.7	3.6
Ease*	3.8	3.0
Efficiency	3.1	3.4
Effectiveness	3.1	3.5
Satisfaction	3.6	3.3
Usefulness	3.9	3.7
Clarity*	4.0	2.5

Finally, participants were encouraged to ‘spill their mind’ at the end of the survey, through a comments field asking them for any remarks regarding their experience working with the prototype. They were also asked to indicate which prototype they preferred.

4.2. Results

All 15 participants completed the search tasks for both prototypes in less time than the duration of the original video. One participant did not complete the questionnaires for both prototypes, so his answers were not included in the comparison of the user satisfaction scores. The results for search task performance are shown in Table 1 and those for user satisfaction in Table 2. The values in Table 2 range from 1 for the most negative term to 5 for the most positive term. To compare the answers given for the different prototypes, Wilcoxon Signed Rank Tests were run on the equal pairs of questions. The variables for which significant differences were found are indicated with an asterisk (*).

On the task performance tests, the VideoTree prototype performed better for the percentage of video watched ($Z=-2.0$, $p<.05$). Out of 15 participants, 11 watched a smaller percentage of the video with the VideoTree browser than with the

Storyboard browser. Moreover, the average total time spent was less with the VideoTree browser than with the control browser. The difference, however, was not significant.

On the perceived usability measures, users rated the VideoTree prototype higher on flexibility than the Storyboard prototype (Z=-2.0, p<.05). When asked to rate the prototypes between Easy and Difficult, participants rated the VideoTree prototype significantly often as more difficult than the Storyboard prototype (Z=-2.1, p<.05). The VideoTree prototype was also rated less clear than the Storyboard alternative (Z=-2.6, p=.01). As can be seen from Table 2, the difference for this question is rather large: the mean score for Confusing vs. Very clear was 2.5 for the VideoTree browser compared to 4.0 for the Storyboard prototype.

5. DISCUSSION

In line with our expectation that the use of hierarchy in video abstractions helps users to find what they are looking for more easily, we found that users watched less video with the VideoTree prototype, and tended to be faster at completing their tasks. As for the user experience, we found that they rated the VideoTree browser as more flexible, but they also found it more difficult to use and more confusing.

Regarding the difficulty to use, multiple participants stated that once they figured out how to work with the VideoTree browser's navigation, they could work with the system a lot faster. This learning curve on VideoTree browsing may make it more suitable for situations where the detail it makes accessible is needed, as in researching shots. Because users were not given any introduction or exploration time beforehand, some users did not figure out the interactive navigation. This, however, was the case for both the Storyboard prototype and the VideoTree prototype. Users who had this problem, did all the navigation using the very limited slider control of the video player in the top right corner. Since the length of the entire video, i.e. over 24 minutes, was represented by the small slider, they struggled with its low accuracy. This may have attributed to an overall lower appreciation of both browsers. This was aggravated by a bug present in both prototypes, which caused the slider to lag when seeking backwards. Since the slider was the same in both prototypes, however, this does not affect the comparison between the prototypes.

Concerning the lower clarity of the VideoTree browser, it was expected that adding more detail and requiring navigation by zooming and panning would attribute to the browser's perceived complexity. Still, there is at least one other factor that may have contributed to participants finding the VideoTree browser confusing. In its present form, the browser does not visualise duration. Two nodes are the same size, even if they significantly differ in length. This makes it hard for users to estimate the duration of a segment. For example, a number of users were found to explore the intro sequence, which had a lot of shot changes in a very short time. Because there was no visual cue telling users that this segment was actually a lot shorter than the next segments, it took them relatively long to figure this out. Adjusting the width of the nodes to encode their relative duration may provide such a cue, at the cost of losing the original aspect ratio. The choice of preserving the aspect ratio of each keyframe also resulted in some nodes ending up on a different height, even though they were on the same level. At least one user was bothered by this, stating he was confused by the nodes not being aligned.

We also identified a number of additional improvements to be made to the navigation using VideoTrees from observing and questioning the participants. A first issue for improvement in the browser is the current lack of progress indication. Users found it hard to relate the slider to the VideoTree navigator view to see what part of the video was playing. They often made jumps backwards from where they were playing, only because they did not realise they had already passed that segment. A progress indicator in the VideoTree display would also aid in determining the length of different segments. A second point for improvement was found for shots of a very short duration, such as the ones in the introduction. For any segment shorter than a few seconds, users found it easier to just watch the fragment than to navigate inside it in the VideoTree. This effect was especially apparent in conversations, which take up a lot of nodes in the tree, while their total duration is short. This suggests that nodes of a duration below a certain threshold might better be merged or removed. Finally, it was not clear to users how far they were zoomed in. A thumbnail navigator view might have helped [20], but for a future version of a browser based on VideoTrees, more advanced focus+context techniques should also be considered.

6. CONCLUSION AND FUTURE WORK

We explored the benefits of hierarchical presentation of keyframes in video browsing by presenting and evaluating a novel type of video surrogate: VideoTrees. VideoTrees are capable of coherently and concisely representing video in detail in a hierarchical form. The VideoTree browser was developed for navigating VideoTrees and a prototype of this browser was compared with a storyboard browser in a preliminary usability study. Users found the VideoTree prototype more flexible, but less easy to use. In addition, they needed to watch a smaller percentage of the video to complete the given tasks.

The user study further revealed a number of potential improvements, such as better progress indication and visual cues to the duration of segments in the VideoTree. The addition of these and possibly other functionalities in a future browser may help users to more easily find their way through VideoTrees. At the present time, the learning curve needed for using the VideoTree browser makes it mainly useful for situations where a high level of flexibility and detail is desirable over simplicity. The use of hierarchy in video surrogates, however, has shown potential for future developments in that direction.

7. REFERENCES

[1] J. Fan, A.K. Elmagarmid, X. Zhu, W.G. Aref, and L. Wu, "Classview: Hierarchical video shot classification, indexing, and accessing," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 70–86, 2004.

[2] M. Worring, C.G.M. Snoek, O. De Rooij, G.P. Nguyen, and A.W.M. Smeulders, "The mediamill semantic video search engine," in *Proc. ICASSP*, Honolulu, 2007, vol. 4.

[3] M.S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM TOMCCAP*, vol. 2, no. 1, pp. 1–19, 2006.

[4] L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *Proc. MULTIMEDIA '99*. 1999, pp. 489–498, ACM.

[5] W. Ding, G. Marchionini, and D. Soergel, "Multimodal surrogates for video browsing," in *Proc. ACM DL'99*, 1999.

[6] H. Lee and A.F. Smeaton, "Designing the user-interface for the fschlr digital video library," *Journal of Digital Information, Special Issue on Interactivity in Digital Libraries, Vol.2 No.4*, vol. 2, no. 4, 2002.

[7] A. Komlodi and G. Marchionini, "Key frame preview techniques for video browsing," in *Proc. DL '98*. 1998, pp. 118–125, ACM.

[8] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *Int. J. Digital Libr.*, vol. 6, no. 2, pp. 219–232, 2006.

[9] W.-G. Cheng and D. Xu, "Content-based video retrieval using the shot cluster tree," in *Proc. ICMLC*, Xi'an, 2003, vol. 5, pp. 2901–2906.

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

[11] E. Bertino, J. Fan, E. Ferrari, M.-S. Hacid, A.K. Elmagarmid, and X. Zhu, "A hierarchical access control model for video database systems," *ACM Trans. Inf. Syst.*, vol. 21, no. 2, pp. 155–191, 2003.

[12] B.T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM TOMCCAP*, vol. 3, no. 1, pp. 3, 2007.

[13] Y. Geng, D. Xu, and S. Feng, "Hierarchical video summarization based on video structure and highlight," in *Proc. SSPR*, Hong Kong, 2006, vol. 4109 LNCS, pp. 226–234.

[14] J.Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi, "An interactive comic book presentation for exploring video," in *Proc. CHI '00*. 2000, pp. 185–192, ACM.

[15] S. Srinivasan, D. Ponceleon, A. Amir, and D. Petkovic, "What is in that video anyway?: in search of better browsing," in *Proc. ICMCS*, 1999, vol. 1, pp. 388–393.

[16] H. Iyer and C.D. Lewis, "Prioritization strategies for video storyboard keyframes," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 5, pp. 629–644, 2007.

[17] M. Demtschyna, "Pal vs ntsc," http://www.michaeldvd.com.au/Articles/PALvsNTSC/PALvsNTSC.asp, December 2007.

[18] J.S. Boreczky and L.A. Rowe, "Comparison of video shot boundary detection techniques," in *Stor. and Retr. for Still Image and Vid. DB IV*, Sethi Ishwar K. and Jain Ramesh C., Eds., San Jose, CA, USA, 1996, vol. 2670, pp. 170–179, Soc. of Photo-Opt. Instrum. Eng.

[19] *The New Oxford American Dictionary*, Oxford University Press, 2nd edition, March 2005.

[20] I. Herman, G. Melancon, and M.S. Marshall, "Graph visualization and navigation in information visualization: a survey," *Proc. IEEE TVCG*, vol. 6, no. 1, pp. 24–43, 2000.

[21] J. Lamping, R. Rao, and P. Pirolli, "A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," in *Proc. CHI '95*, 1995, pp. 401–408.

[22] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," in *Proc. CHI '92*. 1992, pp. 93–98, ACM.

[23] D. Ponceleon and A. Dieberger, "Hierarchical brushing in a collection of video data," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, 2001, pp. 1654–1661.

[24] C. Petersohn, "Fraunhofer hhi at trecvid 2004: Shot boundary detection system," TREC Online Proc., TRECVID, 2004, http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf.

[25] F.D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, Sept. 1989.

[26] J.P. Chin, V.A. Diehl, and K.L. Norman, "Development of an instrument measuring user satisfaction of the human-computer interface," in *Proc. CHI '88*. 1988, pp. 213–218, ACM.