NoteVideo: Facilitating Navigation of Blackboard-style Lecture Videos

Toni-Jan Keith Monserrat[1], Shengdong Zhao[1], Kevin McGee[2], Anshul Vikram Pandey[1]
[1]NUS-HCI Lab, School of Computing [2]NUS Graduate School for Integrative Sciences & Engineering
National University of Singapore, Singapore 117417
[1]{tjkpm, zhaosd, anshul}@comp.nus.edu.sg, [2]mckevin@nus.edu.sg

ABSTRACT

Khan Academy's pre-recorded blackboard-style lecture videos attract millions of online users every month. However, current video navigation tools do not adequately support the kinds of goals that students typically have, like quickly finding a particular concept in a blackboard-style lecture video. This paper reports on the development and evaluation of the new *NoteVideo* and its improved version, *NoteVideo+*, systems for identifying the conceptual 'objects' of a blackboard-based video – and then creating a summarized image of the video and using it as an in-scene navigation interface that allows users to directly jump to the video frame where that object first appeared instead of navigating it linearly through time. The research consisted of iteratively implementing the system and then having users perform four different navigation tasks using three different interfaces: *Scrubbing*, *Transcript*, and *NoteVideo*. Results of the study show that participants perform significantly better on all four tasks while using the *NoteVideo* and its improved version - *NoteVideo+* - as compared to others.

Author Keywords

NoteVideo, NoteVideo+, Video, Video Navigation, Online Streaming, Video summary, Blackboard-style videos.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Khan Academy's pre-recorded blackboard-style lecture videos attract more than 2 million online users every month [24]. Among other things, it enables a different style of education called flip teaching in which the student first studies the topic via videos by himself and uses the class time for additional learning-based interactive activities [5].

In this context, it is important to understand and support a variety of different student interactions with the videos. Such interactions include, among others: wanting summaries for orientation, finding where particular concepts are explained, and reviewing a specific portion of the video that has already been viewed.

In other words, the user may need to play and attend to large amounts of arbitrarily long videos in order to find the section of interest. We now turn to look to alternative proposals for automatically providing better navigation controls for video navigation.

RELATED WORK

Broadly, the related work consists of developing video browsing applications that rely on interaction similar to classical video players; applications that allow users to explore the video corpus using search; applications that visualize video content in unconventional ways; and systems that support direct manipulation in video browsing (for a comprehensive review, see [19]). In addition, some studies have indicated user preferences about navigation interfaces; for example, some work suggests that when it comes to video search, users prefer storyboards [25].

Video-Player Interfaces. There are many attempts to make video navigation easier by quickly moving forward and backward in a video ("scrubbing"). These include 'fast forward' that maintains audio pitch [13], zooming to different timescales [9], variations on improved timelines and random access [16], variations on 'overviews' (or table of contents) based on visual (e.g., keyframes) or audio (e.g., applause, cheering) features that make it possible to segment the video [1, 13]. For educational video based on slides, the separate slides (and the textual content) have been used to automatically provide table-of-content overviews to support navigation [13].

Search-Based Interfaces. Another area of active research involves search-based video retrieval, supporting queries based on textual keywords, visual examples, and concepts [20]. For textual search, some work on news videos is able to make use of the fact that there can be close-captioned text [14, 22]. For visual/example-based search, some approaches make use of low-level feature-extraction [2]. Concept-based search involves correlating some low-level feature(s) (e.g. an equals sign) with some concept

CHI'13, April 27 – May 2, 2013, Paris, France.

(e.g. ‘equations’)’ typical examples involve concepts based on visual features that can be identified, such as outdoor/indoor [21] and cityscape/landscape [26].

Unconventional Video Representations. Many of the solutions in this category show most, if not all, of a video in a way that allows users to preview and then quickly select some portion of the video, for example, ‘fisheye’ representations [3] and "video trees" [10]. Similarly, there are a number of techniques based on ‘spreading out’ the video spatially to support user navigation; for example, there is work on ‘salient stills’, which summarize the temporal changes that occur in a moving-image sequence with the salient features of individual frames preserved [23], and the Video Summagator - a volume-based interface for summarizing static and dynamic content as a space-time cube [18].

Direct Manipulation in Video Browsing. Perhaps, the most relevant to navigating blackboard-style lectures is work on systems that support in-scene navigation and directly jumping to a video frame instead of navigating linearly through time. The Chronicle system [7] involves video capture of an entire document history, indexed by document revisions and UI events. Users are then able to browse graphical representations of a document's revisions by manipulating an interactive timeline that provides a visual scheme of state and event histories. Another kind of system supports object-centric scrubbing [4, 8, 6, 11], where objects in videos are detected based on the history of movement. Once these objects are detected, the system supports scrubbing by selecting an object and dragging it backward and forward in time, manipulating the timeline. It is still linear search of specific parts of video where the object is, as you are only able to move and ‘scrub’ using the object instead of the scrubber, across a specific part of the timeline. Although it does not involve video, there has also been relevant work on augmented whiteboard systems. For example, the Flatland system [17] automatically segments augmented whiteboard diagrams and notations into meaningful clusters, which can transformed (i.e., moved, squashed, flipped, copied/pasted), as well as removed and then later retrieved (i.e., re-projected on the white board).

RESEARCH FOCUS

There is very little work on the specific problem of automatically generating an interface that allows students to quickly find a particular concept in a blackboard-style lecture video.

None of the existing work fully addresses the concerns and needs of students reviewing blackboard-style video lectures. Asking students to use traditional video interface controls (play, linear navigation, pause) are obviously limited. Search based systems that make use of textual transcriptions can make it easier to quickly locate certain parts of a video (e.g., when a sought concept is mentioned by name), but this approach suffers from a number of limitations, such as a concept being mentioned frequently. The existing ‘unconventional’ visualization interfaces tend to support quickly finding moments in the video when there are significant visual changes, something largely absent from blackboard-style videos.

Beyond that, existing work on direct manipulation in video browsing suffers from a number of significant limitations when it comes to navigating blackboard-style videos. The Chronicle system [7] relies on being able to capture the entire video creation history; it will not work for videos that are already completed. Object-centric scrubbing systems [4, 8, 6, 11] will not work for the static objects of a blackboard style lecture videos. The Flatland system [17] supports the typical activities of collaborative, augmented whiteboard usage rather than addressing the navigation issues that arise for a timeline-based medium such as video; furthermore, the system does not record or make use of any of the discussion that happens during the use of the whiteboard, so this is a significant limitation for lecture videos.

This paper reports on the development and evaluation of a novel video navigation system for blackboard-style videos called *NoteVideo* and its improved version called *NoteVideo+*. This system addresses the limitations of existing video-navigation systems by automatically generating a control interface that provides an overview as well as the ability to quickly navigate and jump to specific and appropriate points in the video.

METHOD

The research consisted of implementing the *NoteVideo* system and then having users perform comparative navigation tasks using the Youtube video player, a search-based interface, and the *NoteVideo* interface in a controlled experiment. The goal of the study was to determine which interface makes it easier to find different parts of a blackboard-style video.

The remainder of this section describes the structure of blackboard-style video lectures, relevant teacher activities, the *NoteVideo* system, the study participants, the study protocol, and the data-gathering and analysis techniques used in the study and the development of an improved version called *NoteVideo+* at the end.

Blackboard-style Lecturer Activities

The lecturer of blackboard-style videos typically performs only the following four types of activities: *drawing* symbols or graphs on the board, *erasing* or *scrolling* the blackboard to create more drawing space, and *explaining* concepts (see Figure 1). The first three types of activities usually do not overlap with each other while the voice-over explanation can either happen alone or concurrently with any of the other three activities. Successful recognition of these activities is essential in order to process these videos.

For each type of activity, we need to identify its start and end points, and understand the actions that happened in-between.

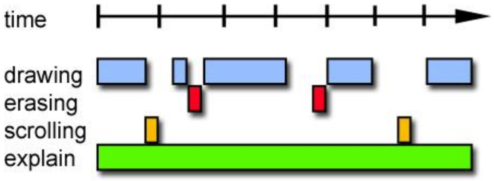


Figure 1. The distribution of the four types of activities (drawing, erasing, scrolling, and explaining) in a typical Khan Academy video.

Audio explanation: This activity happens throughout the lecture and can be regarded as the default or background activity, so no additional detection is necessary.

Drawing: Intuitively speaking, drawing in the physical world means a small amount of material being released onto the background medium, leaving a visible mark. In the digital world, the physical material is replaced with non-background-color pixels on a digital canvas. In a stable video, when the activity involves audio explanation, the visual content detected in each frame should be roughly constant until the lecturer starts to draw additional content on the blackboard. To detect the start of such drawings, one can simply continuously compare the amount of non-background pixels between consecutive frames until a noticeable increase is detected. As the lecturer continues to draw, such a noticeable increase in non-background pixels between consecutive frames should continue until the lecturer stops the drawing. This marks the end of this particular drawing activity (see Figure 2).

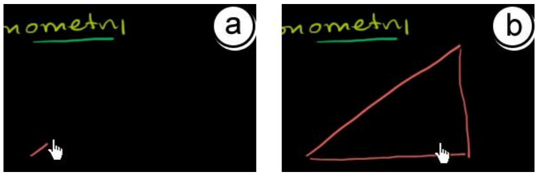


Figure 2. The lecturer draws a triangle. The start of drawing is detected after there's a change in the frame (a) and ends when there's no change in the frame (b).

Erasing: Erasing is simply the opposite of drawing. Instead of adding non-background pixels between frames, they are subtracted. The start and end of an erasing activity can be detected using the same approach as described earlier, except instead of detecting noticeable additions, the system detects noticeable subtractions.

Scrolling: Scrolling in digital world literally means sliding visual content across a display without changing the relative layout relationship among the individual elements in the display. In the implementation, it means two things: 1) the on-screen coordinates for all visual objects in the scrolling viewport will shift consistently by a certain offset; 2) certain objects can disappear from the frame. To detect scrolling, the system continuously compares the coordinates of all visual objects between consecutive frames. If a systematic offset is detected, scrolling has happened. Note that both erasing and scrolling can cause visual content to disappear from video frames. In erasing, the relative on-screen coordinates of the visual objects do not change, whereas in scrolling, they change systematically.

The *NoteVideo* System

NoteVideo automatically parses blackboard-style lecture videos into simultaneously displayed overview notes, where each element in the overview is linked to its corresponding video segment and is directly playable (see Figure 3 for an overview of how it works). We now describe the *NoteVideo video processor* and *interface.*

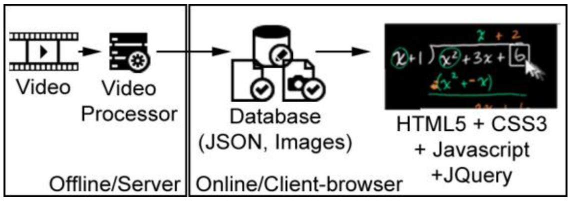


Figure 3. Overview diagram of how *NoteVideo* works.

NoteVideo Video Processor

The video processor is responsible for recognizing and retrieving all the visual objects in the video via 4 steps: pre-processing, object extraction and tagging, erasure- and scrolling-handling, and layout management.

Pre-processing: Before object extraction, visual elements that are not part of drawing notes (such as logo, mouse pointers) are identified and removed from the video frames by applying a template matching function.

Object extraction and tagging: In object extraction, we are most interested in the starting time at which an object is introduced, since that's most likely the time when the lecturer will start discussing about it.

As previously discussed, *NoteVideo* can identify all the drawing activities. To get the visual objects drawn from each drawing activity, we take the starting and ending frames from a drawing activity and perform a subtraction. The difference between the two frames is now an image of newly drawn objects.

Each visual element detected (saved as a separate image) will be processed and stored with its meta-data that include the x, y coordinates relative to the top-left screen position, length, height of the bounding box, timestamp of the start of the drawing activity, etc.

Erasure and scrolling-handling: In addition to object extraction, erasing and scrolling need to be handled since they affect the layout of the objects. There are three types of erasure activities: *typo erasure*, *update erasure*, and *clearance erasure*. Each of them has a different purpose and affects the navigation interface differently.

1) *Typo erasure* happens when a typo is detected. This type of erase typically happens in small scale (only affecting one or a few visual objects), and happens immediately after the objects are drawn.

In *typo erasure*, the ‘incorrect’ visual objects are mistakes, and need to be removed from the visual object collection so that they will not appear in the *NoteVideo* web interface. This type of erasure can be detected using the duration the erased object ‘lives’ (or the time span between its first appearance and disappearance) in the video, and removing it if it is less than a threshold (current threshold is at 3 seconds, after doing a trial-and-error procedure on several videos).

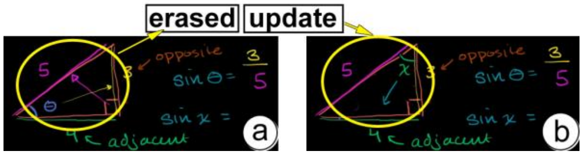


Figure 4. (a) The objects inside the triangle (yellow circle) will be erased and updated to draw a new angle (b) and get the ratio of the sine of the new angle.

2) *Update erasure* happens when the lecturer wants to update a piece of visual content in order to illustrate a related concept. For example, when lecturer explains the concept of getting the ratio of the SIN function given an angle in trigonometry, he first explains it using a right triangle with an angle ‘theta’ as an example, and later re-explains it using the same triangle but with an angle ‘x’. In this case, much content on the screen (the textual explanation of the SIN function) is still relevant, except the diagram needs to be update (see Figure 4)

For *update erasure*, both the erased and newly added visual objects need to be referenced so that they all appear in the web interface. In the current implementation, the new and older visual objects reside in different layers and overlap each other.

Note that if a lecturer performs a quick animation within a small region, our current implementation can mis-classify it as a typo. To solve this problem, one can check the frequency and coherence of the update events. Typos will happen most likely once or twice in a short time, but an animation requires a sequence of update events.

3) *Clearance erasure* is used to clear all the content on the board so that new content can be added.

For *clearance erasure* and *scrolling*, the canvas is cleared or expanded so that new visual objects can be added. In such cases, the newly added visual objects are indexed with new on-screen coordinates so that they will be placed on a new virtual blackboard adjacent to the previous blackboard.

Layout management: Layout is managed using the initial positions of the objects in the video and mapping them as their initial positions in the board interface. If the visual elements appear right after a clearance erasure or scrolling, an additional data called a board number is added and used as a multiplier to the video’s width and added to their ‘x’ coordinate. This will move visual elements of the new board to the right of the previous objects.

Once the visual objects are retrieved, they are displayed in the *NoteVideo* web interface.

NoteVideo Interface

The *NoteVideo* interface consists of the video board for watching and interacting with the video content. The visual objects in the main video board are spatially laid out on rectangular blackboards. The objects position typically replicates their positions in the video with a few exceptions (detailed in Layout management).

NoteVideo provides two ways to interact with video.

1) Point and click on objects: both visual objects for the video segments are directly clickable. Once clicked, the video will be navigated to the starting timestamp of the corresponding object and start to play the video inside the main video board (see Figure 5).

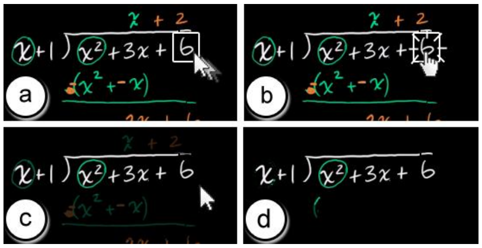


Figure 5. Interaction sequences with *NoteVideo* interface (a) mouse over on any interactive visual objects highlights it; (b) clicking on the object starts the video from its associated timestamp; (c) during video play, visual objects not visible from the current clip appear “faded” as the mouse enters the board; (d) they disappear as the mouse exits the board.

2) Click and drag on visual objects. Users can adjust the playing of the video by click and drag on any object either to the left (for rewinding) or to the right (for fast forwarding). The speed of rewind or fast-forward is proportional to the distance of the drag.

The web-based video navigation interface is implemented using the three main web technologies: HTML5, Cascading Style Sheets (CSS), and JavaScript (with the JQuery 1.8 library and JSON), which is responsible for the structure, style, and interactivity of the web interface, respectively.

All extracted visual objects (along with related meta-information) are stored in a text-based, human-readable file using the JavaScript Object Notation (JSON) language-independent data format. To display the visual object at its corresponding on-screen location from the original video, its CSS absolute x and y position attributes are defined with its original on-screen location.

The JQuery Javascript library handles the interactions of the visual elements or groups, taking advantage of the newly introduced video tag of HTML5, which allows MP4 videos to be directly embedded in the webpage.

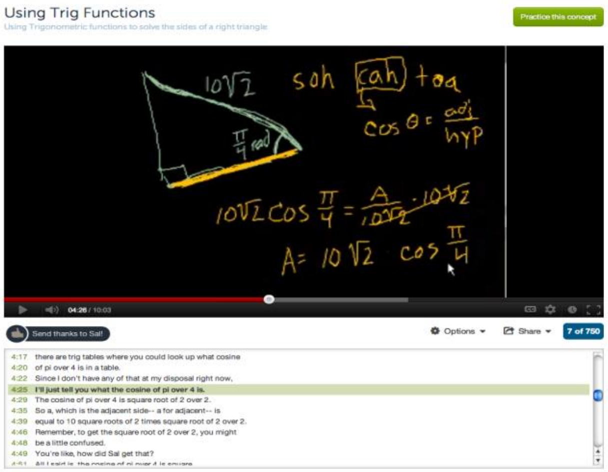
Evaluation Study

To measure the actual performance of the *NoteVideo* navigation interface, a user-study was conducted, involving 15 participants, 3 different interfaces (*Scrubber*, *Transcript*, *NoteVideo*), with 4 video navigational tasks per interface. Each session with a participant lasted approximately 1.5 hours (including breaks).

Participants

The participants who took part in the study included 5 females and 10 males between the ages 18 to 36 (Mean: 23.93, SD: 4.62), volunteered for the study.

All had previous experience with the Youtube scrubbing interface. None of the participants had prior experience with the transcription-based interface introduced by the Khan Academy, and none had previous experience with *NoteVideo*. As the prior experiences with the three interfaces were different, practice tasks were introduced to familiarize everyone with all the interfaces.



Using Trig Functions

Using Trigonometric functions to solve the sides of a right triangle

4:17 there are trig tables where you could look up what cosine
4:20 of pi over 4 is in a table.
4:22 Since I don't have any of that at my disposal right now,
4:25 I'll just tell you what the cosine of pi over 4 is.
4:29 The cosine of pi over 4 is square root of 2 over 2.
4:35 So a, which is the adjacent side-- a for adjacent-- is
4:39 equal to 10 square roots of 2 times square root of 2 over 2.
4:46 Remember, to get the square root of 2 over 2, you might
4:48 be a little confused.
4:49 You're like, how did Sal get that?
4:51 All I said is, the cosine of pi over 4 is square

Figure 6. *Transcript*-based navigation interface. It also includes a *scrubbing*-based navigation interface.

Materials and Apparatus

For the study, participants compared using *NoteVideo* with the timeline-based scrubbing interface (*Scrubber*) and a transcript-based navigation interface (*Transcript*) (see Figure 6). The *scrubber* is the current and most popular approach and is included as the baseline comparison. Video navigation from its text transcript is another well-known approach to enhance the video navigation experience, especially for classroom lecture videos [27, 8]. In *Transcript*, text search is supported by the web browser as an in-built functionality.

Although it would have been possible to use other video navigation interfaces (e.g., related work), the study was conducted with interfaces that are commonly used for blackboard style lecture videos – and which allow comparison of the interfaces that use a browser to navigate the videos (closely mimicking real-world scenarios of users browsing blackboard-style lecture videos on the web).

Key-frame based navigation technique was also considered. However, as it takes a large amount of screen real estate, it will make it an unfair comparison. If one chooses to place the frames in a narrow scrollable region, it becomes similar to the scrubber interface and unlikely to provide a great advantage.

The study was conducted on a Dell Optiplex 990 model workstation, with an Intel® Core™ i5-2500 CPU @ 3.30 GHz and 8GB RAM running on Windows 7 OS, with a 27” display. A 3-button, optical laser mouse and a standard QWERTY keyboard were used for input. The *NoteVideo* interfaces runs on the XAMPP webserver in localhost mode. The *scrubber* and *transcript* interfaces were provided by Khan Academy’s website. All videos were pre-loaded in the local computer to eliminate video loading latency. The video playing area for all three interfaces had the same standard size of 480 x 360. The 6 videos were randomly selected from the Khan Academy website. Four are about linear equation and two are on trigonometry.

Study Protocol

Each participant was briefed about the purpose of the study and was given practice tasks for each of the interfaces so that they could familiarize themselves with each one. For each interface, participants were asked to complete 4 tasks. Upon finishing the tasks, the participants were interviewed about the experience of using each interface.

A within-participant design was used. The order of the three interfaces was counter-balanced using Latin Square. (i.e., participants were randomly assigned to three groups of 5 participants each). Note that when the participant used the *Transcript*-based interface, they were still allowed to use the *Scrubbing* technique that is inherently a part of the interface to retain the real-world use of the interface.

Task and Stimuli

It is possible to classify video navigation as *sequential*, *ordered*, or *random* [15]. While this classification is useful, it is not specific to the type of information-seeking tasks performed by learners when watching the blackboard-style lecture videos. To better understand the type of tasks users perform when watching blackboard-style lecture videos, an informal interview with 5 users was conducted. Three broad types of tasks emerged as a result of this interview.

1) Visual or audio search for a known cue: User tries to find the start of a discussion point or concept based on certain audio or visual cues. For example, if users want to find the discussion of the term ‘trigonometry’ in the video, they will navigate the video to the part of the video when the writing of ‘trigonometry’ appears. This is also the basic behavior for the other categories.

2) Visual or audio search for a specific cue: The key difference between this search task and the previous one is that users will recognize the cue when she sees or hears it, but she does not know it beforehand. However, in the last case, the user knows exactly what she is looking for. This task is somewhat similar to the ordered search task described in [15] except that the user may have never watched the video before. A common example of this case happens when a user wants to skip a particular portion of the video to jump to the next concept, i.e., when watching the trigonometry video, the user has already known the definition of it, and wanted to skip to the next concept. The user does not know how long the lecture will talk about the definition, but she can recognize when the video is on a new topic.

3) Visual or audio search for a roughly known cue: User tries to find the start of a discussion point or concept based on certain audio or visual cues. However, the user may not know the exact spelling or the exact visual image of the cue, but rather may have only a rough idea of what it is. For example, if a user wants to find the discussion of the term ‘trigonometry’ in the video, but does not remember the exact spelling of the word, instead, they recognize it as ‘tri something’.

The tasks in the study were based on the above categorization. A total of 4 tasks were used for each video. These tasks belong to the following categories.

1) ***Overview***: This is a useful task for the user to obtain an overview of the video before watching, e.g., “*Without playing the video, write in three sentences, the main points or main topic that describes what the video is all about.”* Note that this task was included in order to determine whether it would be helpful for them to understand what the video is about just by looking at the interface (i.e., without actually playing the video).

2) Finding a specific answer (***Find***): In this task, the user has to find an answer to a specified question or equation in the video (visual or audio search for a known cue), e.g.: “*From where you are in the video, go to the point in the video where the value of ‘h’ has been revealed for a given sample triangle.*”

3) ***Play and Skip***: Here, the user is given a specific part to go to, skipping on parts that are known or unnecessary for the user (Visual or audio search for a specific cue), e.g., “*From where you are in the video, play the video for 20 seconds, and then go to the point in the video where another triangle with a new set of lengths for the sides of the triangle has been revealed.*”

4) Review and Explain (***Review***): the user must review a part of the video that explains a certain topic (Visual or audio search for a roughly known cue), eg: “*From where you are in the video, go to the point in the video where the lecturer explains the Pytha-something theorem being used to find the hypotenuse.*” (Note: the actual name of this theorem is Pythagorean theorem).

The order of the tasks in the experiment was same as the order of the description above.

Data-gathering

The data gathered during the study included performance measures as well as some self-reported data about the experience of using the different interfaces.

Dependent Measures: Navigation performance was measured in terms of response time and error distance. Response time was measured from the moment that participants finished reading the instructions and indicated that they are ready to start – up until participants explicitly indicate that they have found the answer. Error distance is measured as the time difference (in seconds) between the actual (logged) timestamp and the reported (intended) timestamp.

Repeated measures of analyses of variance were used to assess the effects of interface (*NoteVideo* vs. *Scrubber* vs. *Transcript*) on error distance and response time.

Post-task Measures. After completing all the tasks, participants were asked to answer multi-choice and open-ended questions that focused on comparing the experience of using the three interfaces.

RESULTS

Overall, there is a significant difference between interfaces in terms of response time, while there is no significant difference among the interfaces for error distance. In terms of self-reported measures, participants preferred *NoteVideo* over the other two interfaces.

Response Time

There is a significant main effect for interface ($F_{2,28}$ = 4.23, p = 0.034). Pairwise t-Tests showed that the response time of *NoteVideo* (32.8 s) is significantly faster than both Scrubber (43.9 s) (*p* = .024) and the *Transcript* (49.3 s) (*p* = .029).

This shows the *NoteVideo* has performance advantages over both the *Scrubber* and the *Transcript* interfaces, while there was no significant difference between the *Scrubber* and *Transcript* interfaces. While it was expected that *NoteVideo* would be faster than the *Scrubber*, it was a surprise to discover that the *Transcript* interface is not faster than the *Scrubber*.

There was also a significant main effect on task ($F_{3,42}$ = 24.58, p < .001). This result is not surprising since both the nature and difficult of the four types of tasks used in the study are different. The average time for each task type is: *overview* (25.25 s), *find* (35.68 s), *play and skip* (42.45 s), and *review* (65.53 s).

To further understand how the performance of the three interfaces compares to each other on the task level, a repeated measure ANOVA analysis was performed separately on the four tasks. There were significant main effects of interface for three of the four tasks: *overview*, *find*, and *play and skip* (all $F_{2,28}$ >4.8, all *p* < .02) except for the *review* task (*p* > .05).

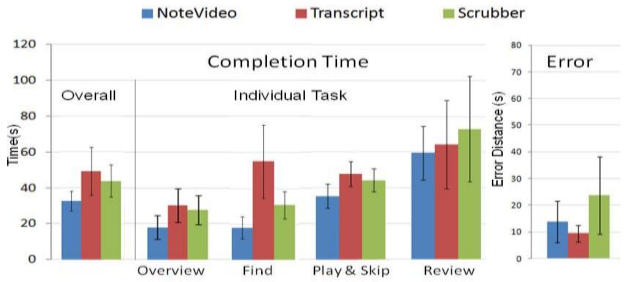


Figure 7. *Left*: Overall response time across the three interfaces and individual response times of the interfaces in (from left to right) describing the overview, finding, play and skip, and review tasks; *Right*: Error distance across three interfaces.

Pairwise t-Tests on the overview task showed that the performance of *NoteVideo* (17.94s) is significantly faster than both the *Scrubber* (27.66 s) and *Transcript* (30.16 s) interfaces, while there was no significant difference between the *Scrubber* and *Transcript* interfaces.

Pairwise t-Tests on the find task showed that *NoteVideo* (18.27 s) is fastest interface (*p* < .01), followed by the *Scrubber* (30.74 s) (*p* = .01), followed by the *Transcript* (55 s).

Pairwise t-Tests on the play and skip task showed that the *NoteVideo* (35.34 s) is significantly faster than both the *Scrubber* (p < .05) and the *Transcript* (*p* < .05), while there was no significant different between the *Scrubber* and *Transcript* interfaces.

Based on the above results, *NoteVideo* outperforms both the Scrubber and *Transcript* interfaces in three out of the four tasks (overview, find, play and skip) without sacrificing accuracy. The one exception is the review task, for which all three interfaces have comparable performance in terms of speed and accuracy.

Error Distance

There were no significant differences in error distance among the different interfaces for interface, task, or their interaction effects. The average error distance for *NoteVideo* is 11.32 s, Script is 24.85 s, and for scrubber is 9.6 s (see Figure 7, *right*).

Post-experiment measures

Participants preferred *NoteVideo* over the other two interfaces because it allowed them to already have an idea of the content and the relationships of the objects just by looking at the video layout without watching it (11/15 for overview task). It also allowed them to interact with the visual elements and start the video from there just by clicking at them (13/15 for finding task, 9/15 for both play and skip, and review tasks). Some preferred *Scrubbing* because of the freedom of movement that timeline gives in playing and skipping (3/15), while the others preferred using the *Transcript* because it allows them to read and know the content and its flow in textual words (4/15 for overview, 2/15 for finding, 3/15 for play and skip and 6/15 for review tasks).

Participants found it hard to use *Scrubbing* because they didn't know the content beforehand, making them rely on playing it or *Scrubbing* through it just to have an idea of the content (11/15 for overview and 9/15 for review and explain tasks). It was also hard for them to use the thumbnails because of their small size, and as a result, they often overshot their target information (8/15 for finding and 7/15 for play and skip tasks). Other participants found the *Transcript* hard to use because they still had to read several lines just to get to their target information or get a grasp of the content (4/15 both the overview and review and explain tasks.) They also found it hard to use the search dialog box because they didn't know the exact word the lecturer used. Thus, some of them still opted to use the scrubber to navigate and the *Transcript* to verify if they got the answer (7/15 for finding and 5/15 for play and skip). Lastly, participants found that sometimes they didn't have any idea what visual element maps to which explanation of a certain topic (2/15 for review and explain).

DISCUSSION

While the result of the experiment clearly indicated that *NoteVideo* is superior to both the *Scrubber* and *Transcript* interfaces, the more important questions are why it is so, and what lessons can we learn from this result? To answer these questions, the characteristics of the information cues offered by the three interfaces were analyzed.

When users are looking for a particular information source, such as the definition of the "Pytha-*something"* theorem or the sine ratio given an angle, they search the visual image and audio sound of keywords, symbols, and diagrams related to the video segments that explain these concepts.

In the scrubber interface, the *visual cues* that users rely on are the thumbnails that pop up as the user hovers its mouse over the timeline. In the *Transcript* interface, the cues come from both the thumbnails and the textual transcript. In the *NoteVideo* interface, users use the spatially laid out visual objects on the blackboard to navigate to the information source.

Among the three types of visual cues, the thumbnails have the lowest quality both due to its small size and the linear search users have to perform in order to find the answer. Transcripts are better than thumbnails in terms of clarity and the ability to allow direct search using the browser's search function – this can eliminate linear scanning through the transcript text. However, text reading is not a pre-attentive task and can result in significant cognitive load as compared to visual scan. This is one possible reason why many participants preferred to use the scrubber even when the transcript text was provided.

The *NoteVideo* interface provides the best quality visual cues, as the visual objects are both clear and directly accessible, making them the easiest to recognize and interact with.

Another factor is the number of choices the user has to face to make a decision. Using a Khan Academy video as an example, the total number of frames in a 9.5-minute video can be 8355 (15 frames per second). Even if the sample rate is reduced to 1 frame/sec, it will result in 570 frames. *Scrubbing* through these frames will take significant amount of time and effort. The transcript has about 1515 words distributed in 183 lines, with at most 11 lines visible without scrolling. *NoteVideo* has 217 detected objects laid out spatially, making it the easiest to go through.

An alternative for lowering the number of choices the user has to sort through in the transcript is to use the search dialog box given by the browser. But this only works when the user knows the exact term used in the transcript. In addition, the frequency of the term needs to be scarce. If the user wants to find a common term, such as 'X=', too many results will be shown which require additional time for the user to process.

Thus, most users opt to scan through all the lines in the transcript as the cost of reading just the first few words of each line is much easier than finding alternative phrases or words to be put in the search box.

Drawbacks of the *NoteVideo* interface

While *NoteVideo* shows clear advantages over its competitors in the first 3 tasks, the comparison result in the fourth task reveals a weakness of the *NoteVideo* interface. If the answer the user is looking for is not clearly represented by the visual objects on the screen, *NoteVideo* does not provide any advantage over the other two interfaces.

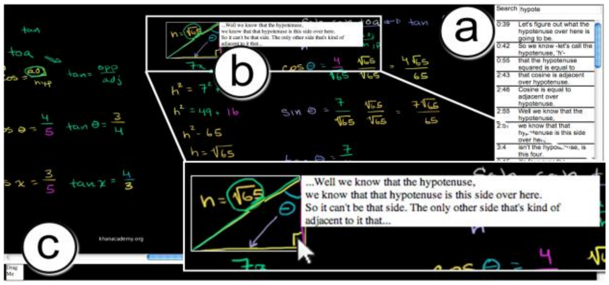
For example, during the study, participants were asked to find the explanation about how the "Pytha-*something"* theorem is used to find the hypotenuse. The participants' mental model for this task is to find a word "Pytha-*something*" in the video; however, no visual objects on the screen matches this term (this term is actually explained in the voice-over), so it took the same amount of time to do this with each interface. Although one might expect that this task would be fastest using the *Transcript* interface, in fact, participants still scanned through the transcript or scrubbed the timeline first before using the search box.

The second problem of the *NoteVideo* interface is it does not show a clear time sequence of the visual objects on the blackboard. Although in most cases this time sequence can be guessed as the order of writing typically goes from top to bottom and left to right, this guideline is not always followed. For example, when the lecturer performs an update erasure, it's no longer clear which object appears first in the video if we only examine their spatial relationship. Therefore, the *Scrubber* interface is also useful for users to discover the sequence of playing through interaction.

Since both transcript- and scrubber-based interfaces have the potential to further improve the performance of *NoteVideo*, we developed a new interface – *NoteVideo+* – that combines the three interfaces together.

NOTEVIDEO+

The new *NoteVideo+* interface (see Figure 8) adds a search box for the transcript (a), a hovering transcript text over the visual elements (b), and scrubber (c)



0:39	Let's figure out what the hypotenuse over here is going to be.
0:42	So we know -let's call the hypotenuse, 'h'-
0:55	that the hypotenuse squared is equal to
2:43	that cosine is adjacent over hypotenuse.
2:46	Cosine is equal to adjacent over hypotenuse.
2:55	Well we know that the hypotenuse,
2:5	we know that that hy tenuse is this side over he
3:4	isn't the hypo se, is this four

...Well we know that the hypotenuse,
we know that that hypotenuse is this side over here.
So it can't be that side. The only other side that's kind of
adjacent to it that...

Figure 8. *NoteVideo*+ Interface

The new *NoteVideo+* interface is intended to address two limitations of the previous interface.

1) While it is very convenient to obtain an overview of the video and access the individual video segments linked by each visual object, the order of appearance of these object is not necessarily self-evident simply by looking at the final layout. The Scrubber interface allows the user to discover this order through scrubbing.

2) Although the visual objects provide a good overview of the lecture content, it's not always detailed enough to answer specific questions a user might have, so it is useful to also link the transcript (if available) with each visual object in *NoteVideo*.

In *NoteVideo*+ (see Figure 8), a timeline is provided to allow the user to scrub through the video frames (see Figure 8 c). Unlike the small thumbnails provided by the Scrubber used in the user study, each frame is displayed full size in the interface. Note that providing this additional feature in the implementation does not incur any additional cost for bandwidth as all the information of the visual objects and their timestamps are already downloaded to the client's machine.

To access the transcript, users can either search for it using the search box, which then highlights the visual elements that are connected to a particular transcript, or just simply hover over a visual object, and the transcribed sentence overlapping with that time period will be displayed (see Figure 8 b). To avoid the potential distraction of constant transcript popups, the popups are only visible on demand (when the ALT key is pressed together with the mouse hover).

Once the improved interface was developed, a second study was performed with 14 new participants in the local university community (age range 21 to 27 years old, M= 24.14, SD= 2.38) to compare the performance of *NoteVideo* and *NoteVideo*+ on the same four tasks described in the previous experiment.

NoteVideo vs. NoteVideo+ Results

Overall, there was no significant difference between either the error distance and completion time for the *NoteVideo* (time: 36.89 s, error: 25.2 s) and *NoteVideo*+ (time: 27.55 s, error:24.8 s) interfaces.

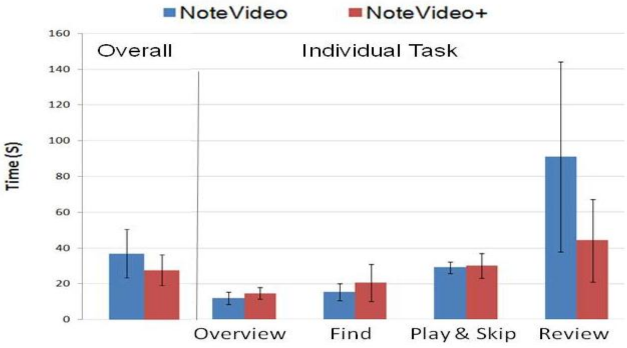


Figure 9. Overall response time (*left*) and average response time of the *NoteVideo* and *NoteVideo*+ across all four tasks (*right*)

However, there is a significant interface x task interaction effect on completion time ($F_{3,51}$ = 3.01, p = .042), indicating the performance difference among different tasks differ between the two interfaces.

Further analysis showed that *NoteVideo*+'s completion time has comparable performance with that of *NoteVideo* for the the *overview*, *find*, and *play and skip* tasks. However, *NoteVideo*+ (44.21 s) is significantly faster than *NoteVideo* (91 s) for the *review* task (t_{13} = 2.18, p < .05) (see Figure 9)

This shows that the integration of scrubber and transcript with the *NoteVideo* interface significantly improves the review task while still retaining the advantages of the original *NoteVideo* interface for the first three tasks. Note that the transcript feature of *NoteVideo*+ is not always possible if no audio transcription is provided for a video. However, in such cases, it is still possible to use the NoteVideo+ without the transcript feature and benefit from its other features.

DISCUSSION AND DESIGN IMPLICATIONS

Interface	Advantages	Disadvantages
Scrubber	Shows sequence / flow of visual action	- Cannot determine information by random access - Small thumbnail - bigger thumbnail = bigger bandwidth
Transcript	Allow search of text not easily identifiable in visual objects	- Only highlights hits and still shows unrelated transcript - Mapping between text and visual object can not retrieved in a glance
NoteVideo	Spatial layout of visual objects that facilitates random access	- Sequence of play not always clear - Difficult to find information if there is no clear visual cue

Table 1. Summary of Advantages and Disadvantages of *Scrubber, Transcript,* and *NoteVideo* interfaces

Table 1 summarizes the advantages and disadvantages of each of the interfaces in navigating blackboard-style lecture videos. Each advantage of one interface compensates for the disadvantage of the other interface. This breakdown also suggests why NoteVideo+ is successful: in addition to leveraging the different advantages of other interfaces, *NoteVideo+*'s interface also better leverages the user's mental model searching videos for specific information.

LIMITATIONS

The current *NoteVideo* implementations handle the blackboard-style video provided by Khan Academy. Although the *NoteVideo* approach is generalizable to other blackboard-style lecture videos, other real world blackboard-style videos will have additional issues (such as occlusion by humans, uneven lighting conditions, etc.). More sophisticated visual object recognition may need to be developed to achieve the results described here.

CONCLUSION AND FUTURE WORK

This paper has presented a novel way of navigating blackboard-style videos using *NoteVideo*, and its improved version, *NoteVideo+*. The evaluation studies with users indicate that *NoteVideo* provides an intuitive and effective way to navigate streaming, blackboard-style educational video content. Using a summarized image of the video as an in-scene navigation interface and interactive visual elements to jump to appropriate and specific points in the video that are closer to users' mental models allows them to target information more efficiently than thumbnails and transcripts alone and combined.

Future work will involve the development and evaluation of more sophisticated refinements of the *NoteVideo* interface. It will also be interesting to study whether *NoteVideo* can function effectively as a tool for other tasks performed by teachers and students who create and use blackboard-style videos.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their valuable feedbacks to improve the paper. We also thank members of the NUS-HCI lab for their generous help and support. This project is funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA) under Centre of Social Media Innovations for Communities (COSMIC).

REFERENCES

1. Arman, F., Depommier, R., Hsu, A., and Chiu, M., Content-based browsing of video sequences, in *Proc*. MULTIMEDIA '94, (1994), 97–103
2. Deng, Y., and Manjunath, B. S., Content-based search of video using color, texture, and motion, in *Proc*. Image Processing ‘97, (1997). 534–537
3. Divakaran, A., Forlines, C., Lanning, T., Shipman, S., and Wittenburg, K., Augmenting fast- forward and rewind for personal digital video recorders, in *Proc.* ICCE ‘05, (2005). 43–44
4. Dragicevic, P., Ramos, G., Bibliowitcz, J., Nowrouzezahrai, D., Balakrishnan, R., and Singh, K. Video Browsing by Direct Manipulation, in *Proc.* CHI '08, (2008). 237-246
5. Flipping the classroom: Hopes that the internet can improve teaching may at last be bearing fruit. http://www.economist.com/node/21529062
6. Goldman, D.B., Gonterman, C., Curless, B., Salesin, D. and Seitz, S.M. Video Object Annotation, Navigation, and Composition, in *Proc.* UIST '08, (2008). 3–12.
7. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories, in *Proc.* UIST '10 (2010). 143-152.
8. Hopfgartner, F., Urruty T., Hannah D., Elliott D., Jose J. M. Aspect-based video browsing – a user study, in *Proc.* ICME’09, (2009). 946–949
9. Hurst, W., and Jarvers, P. Interactive, Dynamic Video Browsing with the ZoomSlider Interface, in *Proc*. ICME ‘05, (2005). 558–561
10. Jansen, M., Heeren, W., and Dijk, B. V. Videotrees: Improving video surrogate presentation using hierarchy, in *Proc*, CBMI ‘08, (2008). 560–567
11. Karrer, T., M. Weiss, Lee, E., Borchers, J. DRAGON: A Direct Manipulation Interface for Frame-Accurate In-Scene Video Navigation, in *Proc.* CHI '08 (2008). 247-250.
12. Kimber, D., Dunnigan, A., Girgensohn, A., Shipman, F., Turner, T., TaoYang. Trailblazing: Video Playback Control by Direct Object Manipulation, in *Proc.* ICME '07, (2007). 675-678
13. Li, F., Gupta, A., Sanocki, E., He, L., and Rui, Y. Browsing digital video, in *Proc*. CHI ‘00, (2000). 169–176
14. Liu, J., He, Y., and Peng, M.. NewsBR: a content-based news video browsing and re-trieval system, in *Proc.* ICCIT ‘04, (2004). 857–862
15. Matejka, J., Grossman, T., and Fitzmaurice, G., Swift: reducing the effects of latency in online video scrubbing, in *Proc.* CHI ‘12, (2012). 637-646
16. Moraveji, N. Improving video browsing with an eye-tracking evaluation of feature-based color bars, in *Proc*. JCDL '04, (2004). 49–50
17. Mynatt, E.D., Igarashi, T., Edwards, W.K., and LaMarca, A. Flatland: new dimensions in office whiteboards. in *Proc.* CHI '99 (1999).346-353.
18. Nguyen, C., Niu, Y., and Liu, F., Video summagator: an interface for video summarization and navigation, in *Proc.* CHI ‘12, (2012). 647-650
19. Schoeffmann, K., Hopfgartner, F., Marques O., Boeszoermenyi L., and Jose J. M., Video browsing interfaces and applications: a review, in SPIE Reviews, Vol. 1, No. 1, (2010). 1-35
20. Snoek, C. G. M., Worring M., Koelma D. C., and Smeulders A. W. M., A learned lexicon-driven paradigm for interactive video retrieval, IEEE Trans. Multimedia 9, (2007). 280–292
21. Szummer, M., and Picard, R. W., Indoor-outdoor image classification. In *Proc.* CAIVD ’98, (1998). 42–51
22. Tang, X., Gao, X., and Wong, C., NewsEye: a news video browsing and retrieval system, in *Proc*. IMVSP ‘01 (2001). 150–153
23. Teodosio, L. and Bender, W., Salient stills, TOMCCAP ‘05, (2005) 16-36.
24. Thompson, C., How Khan Academy Is Changing the Rules of Education. http://www.wired.com/magazine/2011/07/ff_khan/all/1
25. Tse, T., Marchionini, G., Ding, W., Slaughter, L., and Komlodi, A., Dynamic key frame presentation techniques for augmenting video browsing, in *Proc*, AVI ‘98: (1998) 185–194
26. Vailaya, A., Fiqueiredo, M. A. T., Jain, A. K., and Zhang, H.-J., Image classification for content-based indexing, in *Proc*. Image Processing ‘01, (2001). 117–130
27. Villa, R., Gildea, N., Jose, J.. FacetBrowser: a user interface for complex search tasks, in *Proc*. MULTIMEDIA 2008. (2008) 489–498