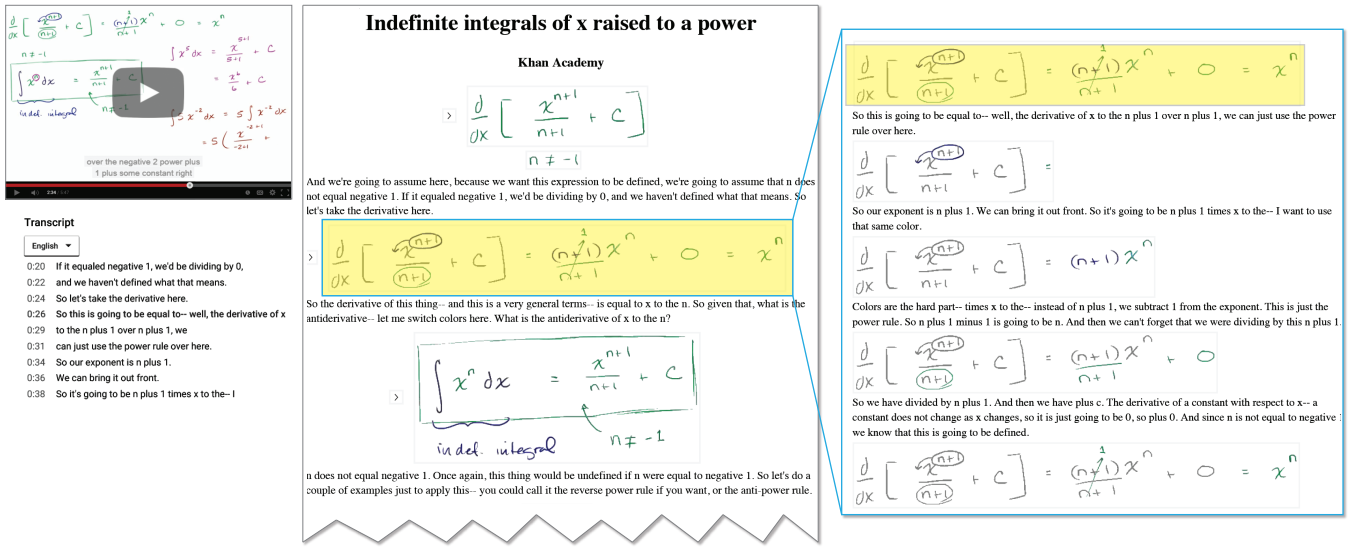
Visual Transcripts: Lecture Notes from Blackboard-Style Lecture Videos

Hijung Valentina Shin* MIT CSAIL
Floraine Berthouzoz Adobe Research
Wilmot Li Adobe Research
Frédo Durand MIT CSAIL



(a) Input video and transcript (b) Visual transcript (c) Visual transcript: detailed view

Figure 1: *Our system transforms contents of a blackboard-style lecture video (a) into a readable interactive lecture note (b) which interleaves visual content with corresponding text. The visual transcript can be read by itself, or used with a standard video-interface as an interactive transcript. Our output shows a compact representation of figures and hides redundant information. Users can click on a figure to see its step-by-step detailed derivation (c).*

Abstract

Blackboard-style lecture videos are popular, but learning using existing video player interfaces can be challenging. Viewers cannot consume the lecture material at their own pace, and the content is also difficult to search or skim. For these reasons, some people prefer lecture notes to videos. To address these limitations, we present *Visual Transcripts*, a readable representation of lecture videos that combines visual information with transcript text. To generate a Visual Transcript, we first segment the visual content of a lecture into discrete visual entities that correspond to equations, figures, or lines of text. Then, we analyze the temporal correspondence between the transcript and visuals to determine how sentences relate to visual entities. Finally, we arrange the text and visuals in a linear layout based on these relationships. We compare our result with a standard video player, and a state-of-the-art interface designed specifically for blackboard-style lecture videos. User evaluation suggests that users prefer our interface for learning and that our interface is effective in helping them browse or search through lecture videos.

CR Categories: I.3.8 [Computer Graphics]: Applications K.3.0 [Computers and Education]: General;

Keywords: Video summarization, Video navigation, Lecture videos, Blackboard-style lectures

1 Introduction

Despite the increasingly important and broad role of lecture videos in education, learning from such videos poses some challenges. It is difficult for viewers to consume video content at their own pace [Chi et al. 2012]. To skip quickly through familiar concepts or slowly review more difficult material, the viewer must interrupt playback and scrub back-and-forth in the timeline. It can also be difficult to find specific information in a video. While scrubbing allows users to browse the visual information in the lecture, it is not effective for skimming the audio content, which often includes critical explanations and context that accompany the visuals. As an alternative, some platforms (e.g., Khan Academy and YouTube) provide synchronized transcripts that allow users to click on a phrase and play the video at that location. However, skimming the transcript for relevant content can also be challenging since the text is not structured, and viewers must click on various parts of the text

*e-mail:hishin@mit.edu

to see the corresponding visuals. Finally, it is hard to get a quick overview of the lecture content without watching the entire video. For these and other reasons, some people prefer static learning materials such as textbooks or printed lecture notes over videos.

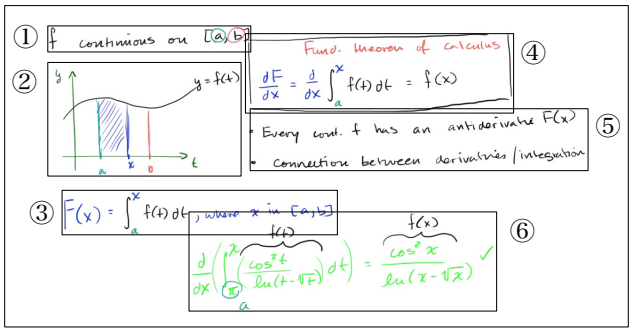
Inspired by lecture notes, we present *Visual Transcripts*, a readable representation of both the visual and audio content of a lecture video that facilitates reviewing, browsing and navigation. We focus on blackboard-style lectures that show a (possibly infinite) blackboard where the instructor writes down by hand the content of the lecture. Visual Transcripts aggregate the full lecture content in a structured format where visual information is segmented and grouped with the corresponding narration text. For example, Figure 1(b) shows our automatically generated output for a math lecture that interleaves verbal explanations with the corresponding equations written on the board. By default, Visual Transcripts hide redundant information to show a compact representation of the content that viewers can expand interactively to show relevant details (Figure 1(c)). Presenting video content in this manner allows users to review the lecture at their own pace while getting both the visual and textual information in a readable, skimmable format. Visual Transcripts can also be linked with the video such that clicking on text or visuals plays the video from the corresponding location. In this respect, Visual Transcripts offer many of the benefits of traditional static media, such as textbooks and lecture notes, while also giving viewers direct access to the video content.

There are two main challenges in transforming a video and its transcribed audio into a Visual Transcript: (1) visuals, which are drawn progressively on the board, must be discretized into meaningful static entities, and (2) visual entities and audio (text) must be organized into a compact, structured format that emphasizes the relationships between the two channels of information. To segment the visuals into meaningful entities, we propose a dynamic programming approach that takes into account both the spatial layout of strokes and the time when they were drawn. We further time-align the transcript with the audio and use this alignment to establish correspondences between the visuals and the text. Finally, we use the visual-text correspondence to detect redundant information and arrange the content in a compact, sequential layout where the text is organized into readable paragraphs.

We evaluate our approach with a user study that compares Visual Transcripts with a baseline transcript-based video player, and an existing, state-of-the-art visual-based video player, NoteVideo [Monserrat et al. 2013]. We measure performance on summarization and search tasks, and observe how the participants interact with the interfaces. We find that Visual Transcripts are an effective medium for studying lecture videos. Specifically, users performed best using Visual Transcripts for search tasks involving text. Users noted that Visual Transcripts helped them to get a quick overview of the video including the details conveyed only through the text, and to efficiently focus in on parts of interest. They also found the structured text easier to read and connect to relevant visuals than the baseline text-only transcript. In a post-study survey, users strongly preferred our interface for learning over the baseline and NoteVideo.

2 Related Work

Video Visualization: There is a large body of work that aims to automatically summarize videos to facilitate navigation and browsing, but most research focuses on live action footage which is very different from educational videos. Recent survey papers [Truong and Venkatesh 2007; Borgo et al. 2011] comprehensively review these techniques, which can be broadly divided into two classes according to their output: *video skims* and *still-image abstracts*. Video



- Depictive sentence for ①: *"Let's say I have some function f that is continuous on an interval between a and b."*
- Explanatory sentence between ② & ③: "*Well, how do we denote the area under the curve between two end points? Well, we just use our definite integral."*

Figure 2: *(top) Lectures convey concepts progressively. Here, the labels (1 through 6) show the order in which concepts were presented. They also organize visuals into discrete entities (outlined in this visualization with bounding boxes). (bottom) Verbal explanations during lectures can either be explanatory or dedicative.*

skims [He et al. 1999; Ekin et al. 2003; Ngo et al. 2005; Lu and Grauman 2013] summarize a longer video with a shorter video, usually consisting of segments extracted from the original video. These skims retain audio and motion elements and are especially useful for understanding dynamic scenes, but they are less suitable for conveying the dense, static information of blackboard-style lectures. Still-image based methods [Uchihashi et al. 1999; Barnes et al. 2010; Hwang et al. 2006; Boreczky et al. 2000] primarily focus on conveying the visual content of a video in static form through a collection of salient images extracted from the video. [Christel et al. 2002] and [Pickering et al. 2003] developed a still-image based method specific to news stories that combines text and images into summaries. Most relevant to our work is [Choudary and Liu 2007], which summarizes blackboard-style lectures by creating a panoramic frame of the board. Our work combines the audio content with the visuals and therefore maintains the sequence of the lecture and makes textual content directly accessible.

Tools for Online Lecture Videos: [Kim et al. 2014a] uses interaction data collected from MOOC platforms to introduce a set of techniques that augment existing video interface widgets. For lecture videos based on slides, [Li et al. 2000] use separate slides to automatically generate table-of-content overviews. These works *annotate* the original video with useful data to facilitate navigation, but do not reformat the video content. [Pavel et al. 2014] provides a tool to create *video digests*, structured summaries of informational videos organized into chapters and sections. They use only the transcript to segment and summarize the video, whereas we leverage both the visual and audio content. Most closely related to our work is NoteVideo [Monserrat et al. 2013], which presents a summary image of blackboard-style lecture videos. Their image is composed of click-able visual links to support spatial and temporal navigation. Although they provide a search box for the transcript, text is not included as part of their summary.

3 Visual Transcript Design

The design of our interactive Visual Transcripts and our approach for generating them from input videos are informed by the following key characteristics of blackboard-style lectures:

- **Lectures present information progressively.** Most lectures convey concepts in a progressive manner where each new

piece of information builds on the previously presented content. For example, Figure 2 (*top*) shows a panoramic image of the board for an entire lecture, where the labels show the order in which things were presented. Understanding the lecture often requires knowing this order. To emphasize presentation order, our Visual Transcript arranges all the content within the video in a top-to-bottom linear format.

- **Visuals are organized into discrete entities.** The visual content of a lecture is typically organized into well-defined entities (e.g., a line of text, an equation, an explanatory figure) that correspond to the set of presented concepts. For example, Figure 2 (*top*) shows visual entities in a calculus lecture. Each visual entity consists of strokes that are close together in both space and time. Moreover, since people are accustomed to parsing visual information line-by-line, from top to bottom, and left to right, visual entities are often laid out in the same manner. Building on this observation, our system segments drawings on the board into visual entities based on their spatial alignment and temporal proximity.

- **Audio content complements visuals.** In our analysis of lecture videos, we found that verbal explanations tend to serve one of two broad objectives. Explanations given while the instructor is not drawing are often *explanatory*, providing additional information not directly represented in the visuals or making connections between drawings. On the other hand, explanations given while the instructor is drawing are typically more *depictive*, repeating or reading aloud the visual information (Figure 2, *bottom*). While depictive explanations can help viewers follow along with the video, they often result in long, repetitive transcript text that is cumbersome to read or skim through. This problem is exacerbated by the fact that most spoken explanations are somewhat colloquial. Our system automatically categorizes transcript text as explanatory or depictive, and in our output, we hide depictive sentences and show explanatory text interspersed with the set of visual entities extracted from the video. [Large et al. 1995] and [Christel and Warmack 2001] have shown that such combinations of pictures and captions aid recall and comprehension as well as navigation of video material. Our design gives the viewer relevant context for understanding the visual information without cluttering the output with redundant text.

4 Method

Our method consists of three main stages. We first segment the visual content of a lecture into visual entities using a dynamic programming approach (Section 4.1). We then structure the transcript content by computing temporal correspondences between visual entities and transcript sentences (Section 4.2). Finally, we generate a Visual Transcript by interleaving visual entities with transcript text (Section 4.3). The rest of this section describes these steps in detail.

Pre-processing. The visual content in blackboard-style lectures consists of *strokes*, the set of foreground pixels generated during one continuous drawing action. In the context of a graphics tablet, a stroke corresponds to the continuous path of a pen while maintaining contact with the writing surface. As a pre-processing step, we extract individual strokes from the input video using a method similar to [Monserrat et al. 2013]. We detect the start and end time of each drawing action by comparing the number of foreground pixels in consecutive frames. A large increase marks the start of an action, while no change marks the end. The difference image between the end and start frames gives an image of the stroke drawn during that period. The manual steps involved in this process are (1) identifying the cursor image, which is automatically removed from all frames,

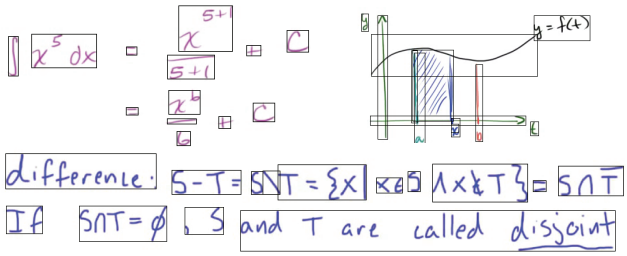


Figure 3: *Examples of strokes (marked by black bounding boxes) extracted from video frames.*

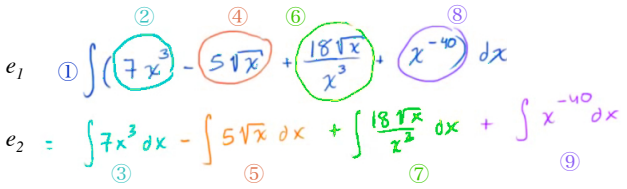


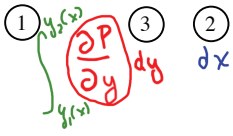
Figure 4: *The instructor goes back and forth between writing two lines, e_1 and e_2. The order of strokes 1-9 is as indicated.*

(2) setting a threshold for foreground/background separation, and (3) setting a smoothing window to get rid of the noise in the foreground pixel count. Depending on the instructor's writing speed, a typical stroke comprises several characters to several words, or it can also be a part of an illustration or a graph (Figure 3).

In addition to the visuals, lecture videos include an audio track with the instructor's spoken explanations. Several on-line video lecture platforms (e.g. Khan Academy, YouTube) provide transcripts of the audio. We assume such transcripts and we use an online audio transcription service (castingwords.com) if they are not available.

4.1 Segmenting Visual Content

One straightforward strategy for grouping strokes into visual entities is to process strokes in the order they are drawn and decide whether each stroke represents the start of a new visual entity or is part of an existing visual entity formed by previous strokes [Mynatt et al. 1999]. While this simple, greedy approach works in some cases, there are many scenarios where it leads to poor segmentations. For example, in the inset figure, there is a large space between the first stroke ($-\int_b^a$, ①) and the second stroke (dx, ②). Without considering the semantics of these symbols, they appear to be separate equations. However, once we consider the subsequent set of red strokes(③) it becomes clear that this is not the best segmentation. In general, computing good stroke segmentations requires considering the global configuration of strokes in both space and time.



In this respect, the problem of segmenting strokes into visual entities (Section 3) is analogous to the *line-breaking* problem, i.e., arranging the words of a paragraph into lines. In both cases, we want to segment a sequence of elements (strokes or words) into an optimal set of groups (visual entities or lines) defined by some scoring function over candidate entities or lines. An important difference is that in the traditional line-breaking problem, only a contiguous set of words can be put on the same line. In our case, strokes in one visual entity can be interspersed by strokes in a different visual entity. For example, the instructor may go back and forth between two lines of equations, or between a graph and an equation (Fig-

Algorithm: Stage 1 - Segmenting Visual Content

Input : list of strokes, S = $\{s_0, \ldots, s_n\}$
Output: optimal set of visual entities, V_n

for *each* $s_i \in S$ **do**
  //Compute V_i: optimal set of visual entities for all strokes up to s_i
  $E_i = +\infty$ //minimum segmentation score up to s_i
  for *each* $j < i$ **do**
    $S_{ji} = \{s_{j+1}, \ldots s_i\}$
    //Compute V_{ji}: optimal set of visual entities from grouping S_{ji} with V_j
    //(1) Consider merging with previous entity in V_j
    $E_{merge,j} = +\infty$ //minimum score to merge to S_{ji} to V_j
    e_j //best entity in V_j to merge S_{ji}
    for *each visual entity* $e \in V_j$ **do**
      $E_{merge,j,e} \leftarrow$ score to merge S_{ji} with e
      if $E_{merge,j,e} < E_{merge,j}$ **then**
        $E_{merge,j} = E_{merge,j,e}$
        $e_j = e$
    //(2) or forming a new entity in addition to V_j
    $E_{new,j} \leftarrow$ score to form new entity S_{ji}
    //take minimum of (1) and (2)
    if $E_{merge,j,} < E_{new,j}$ **then**
      $E_{ji} = E_{merge,j}$
      $V_{ji} \leftarrow$ merge S_{ji} with $e_j \in V_j$
    else
      $E_{ji} = E_{new,j}$
      $V_{ji} \leftarrow$ add new entity S_{ji} to V_j

    //take minimum over all j<i
    if $E_{ji} < E_i$ **then**
      $E_i = E_{ji}$
      $V_i = V_{ji}$

Figure 5: *We use dynamic programming to segment strokes into an optimal set of visual entities. For each stroke, s_i, the algorithm considers all previous partial solutions, $V_{j<i}$ and $S_{ji} = \{s_{j+1}, ..., s_i\}$. For each V_j, it considers two possibilities: merging S_{ji} with an existing entity or forming a new entity.*

ure 4). Given these observations, we propose a dynamic programming approach for stroke segmentation based on the classic optimal line-breaking algorithm [Knuth and Plass 1981] that handles non-contiguous grouping. We first explain the high-level structure of the algorithm before describing the scoring function in detail.

Algorithm Overview

Given a sequence of n strokes $S = \{s_o, \ldots, s_n\}$ ordered by when they appear in the video, we find the optimal set of inter-stroke boundaries that segment the strokes into visual entities. We refer to the boundary between s_i and s_{i+1} as b_i. Our algorithm processes the strokes in order and for each s_i computes and records the optimal set of visual entities V_i formed by all strokes up to b_i, along with the total score $E(V_i)$ of this partial solution. To determine the optimal partial solution for stroke s_i, we consider each previous boundary b_j where $j<i$, and evaluate two possible ways of grouping the set of strokes $S_{ji} = \{s_{j+1}, \ldots, s_i\}$: 1) merging S_{ji} with one of the existing entities in V_j, or 2) forming a new entity with S_{ji}. Allowing S_{ji} to be merged with existing entities enables our algorithm to support non-contiguous stroke groupings. We take the better (lower) of the two scores for S_{ji} and add it to $E(V_j)$ to obtain the total score for the proposed segmentation. After considering all candidate boundaries b_j, we identify the partial solution with the minimum segmentation score and record the corresponding set of entities as V_i and the score as $E(V_i)$. Once the algorithm iterates through all strokes, V_n gives the optimal set of visual entities for the entire lecture. Figure 5 gives detailed pseudo-code of our segmentation algorithm.

Scoring Function

The dynamic programming algorithm described above requires a scoring function that evaluates the goodness of candidate visual entities formed by sets of strokes. We define this scoring function based on several observations: Strokes within a visual entity are

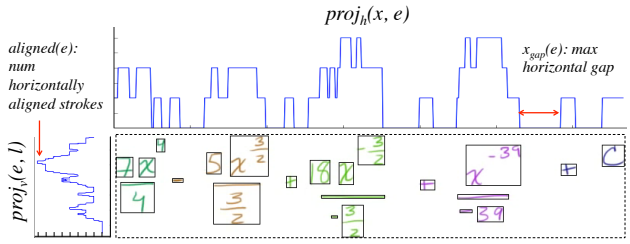


Figure 6: *Horizontal ($proj_h$) and vertical ($proj_v$) projection functions of strokes in a line. In this example, $y_{gap}(e) = 0$.*

(1) compactly arranged (2) and horizontally aligned. In addition, separate visual entities are (3) spatio-temporally distant from each other.

(1) Visual entities are compact. Strokes that belong together in the same visual entity are typically arranged in a compact way. We consider two measures of compactness for a visual line: horizontal and vertical.

- Intuitively, horizontal compactness is related to the horizontal *gap* between strokes within a visual entity. Figure 6 shows an illustration of how gaps between strokes are measured. First, we define a horizontal projection function for a set of strokes, S, as

$$proj_h(x, S) = \left|\{s \in S | x_{\min}(s) \leq x \leq x_{\max}(s)\}\right| \quad (1)$$

  where $x_{\min}(s)$, and $x_{\max}(s)$ are the minimum and maximum x-coordinates of the bounding box of stroke s respectively. Then, the maximum horizontal gap of a visual entity e is

$$x_{\text{gap}}(e) = \underset{x_i, x_{i+1}}{\text{argmax}}(x_{i+1} - x_i) \quad (2)$$

  where x_i and x_{i+1} are distinct consecutive elements in the ordered set $X = \{\, x \mid proj_h(x, e) \neq 0 \,\}$. We observed that the horizontal gap between different visual entities is usually around 100 pixels or more, so we define a horizontal compactness term C_h that imposes harsher penalties when the maximum horizontal gap exceeds this distance.

$$C_h(e) = \left(\frac{x_{\text{gap}}(e)}{100}\right)^2 \quad (3)$$

- Vertical compactness is defined similarly in terms of a vertical projection function, $proj_v$, the maximum vertical gap, $y_{\text{gap}}(e)$, and a typical vertical gap of 40 pixels between different visual entities.

$$C_v(e) = \left(\frac{y_{\text{gap}}(e)}{40}\right)^2 \quad (4)$$

(2) Strokes within a visual entity are aligned horizontally. With the exception of certain illustrations such as graphs, the strokes in most visual entities are horizontally aligned (e.g., equations, lines of text). Thus, we prefer to group horizontally aligned strokes into a single entity. The number of horizontally aligned strokes in each visual entity is computed by taking the mode of its vertical projection function (Figure 6).

$$aligned(e) = \underset{y_{\min}(e) \leq y \leq y_{\max}(e)}{\text{argmax}} proj_v(y) \quad (5)$$

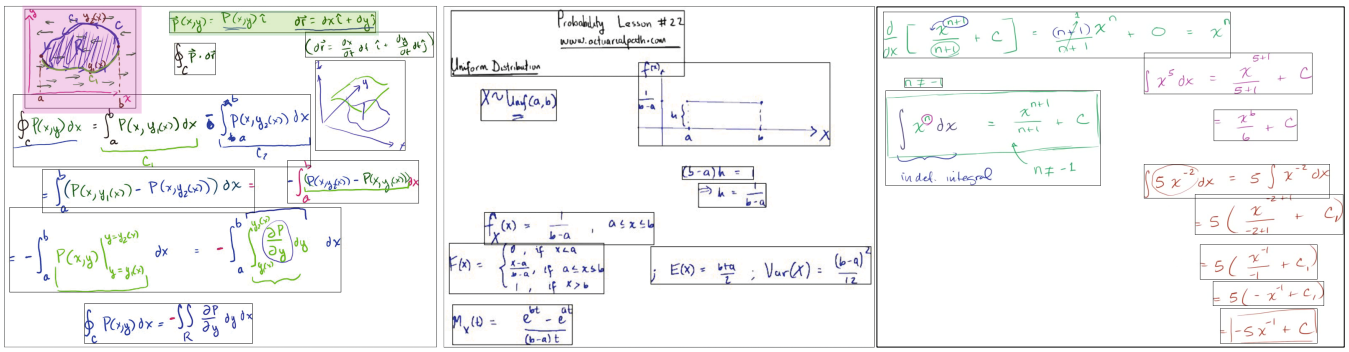


Figure 7: *Examples of visual ~~lines~~entities output from our line-breaking algorithm. Our algorithm successfully identifies meaningful groups even from complex layouts with a mix of equations, figures and graphs.*

We then define an alignment term C_a whose contribution gradually diminishes with the total number of aligned strokes.

$$C_a(e) = aligned(e) - \frac{1}{aligned(e)+1} \tag{6}$$

(3) Visual entities are spatio-temporally distant from each other. This observation is complementary to the first observation, i.e. visual entities are compact. Whereas strokes that belong together are written close together, instructors usually leave some space on the board, for example, between lines of equations or separate illustrations. We express this property by penalizing any overlap between distinct visual entities, measured by the overlapping area between their bounding boxes. In particular, we define the overlap penalty term

$$P_o(V) = \sum_{e_i, e_j \in V, i \neq j} \left(\frac{area(e_i \cap e_j)}{min(area(e_i), area(e_j))}\right) \tag{7}$$

A similar property holds in the temporal domain. For example, after writing a single line of an equation and before going on to the next line, there is a brief pause while the instructor moves the cursor to the next position or provides some verbal explanation. We compute the temporal distance between two consecutive strokes across visual entity boundaries.

$$t_{\text{dist}}(s_i, s_{\text{i+1}}) = \begin{cases} 0, \text{if } s_i, s_{\text{i+1}} \text{ belong to the same visual entity} \\ start(s_{\text{i+1}}) - end(s_i), \text{otherwise} \end{cases}$$

where $start(\cdot)$ and $end(\cdot)$ are the start and end times of when a stroke is drawn in the video. We penalize visual entity boundaries with a small temporal gap.

$$P_t(V) = \sum_{i=0}^{n-1} \frac{1}{t_{\text{dist}}(s_i, s_{i+1})} \tag{8}$$

where n is the total number of strokes.

Combining scoring terms. So far, we have defined terms that measure the compactness (C_h, C_v) and horizontal alignment (C_a) of an individual visual entity e, as well as the spatio-temporal distance (P_o, P_t) between a set of candidate entities V. We combine all these terms into a single scoring function F as follows.

$$F(V) = \sum_{e \in V} [C_v(e) + 0.5C_h(e) - C_a(e)] \tag{9}$$

$$+ P_o(V) + P_t(V) \tag{10}$$

The factor of 0.5 puts a smaller weight on horizontal versus vertical gaps. Higher values of C_a indicate more horizontally aligned strokes and better segmentation, so we put a minus in front.

The final output of our algorithm is a grouping of all the strokes on the board into a set of meaningful visual entities (Figure 7). To test the robustness of our segmentation algorithm, we applied it to 20 video lectures from 10 different authors, using the same set of parameters as described above. The lectures included non-linear layouts of visual content and examples of complex diagrams with several layers of information. In all cases, the algorithm produced reasonable segmentations which generated comprehensible Visual Transcripts. There were few cases ($\approx 5\%$) where the output segmentation was less than ideal, but these did not affect the overall quality of the Visual Transcripts. Please see Limitations for more details. We also test the importance of each of our scoring terms. The full set of results are included in the supplementary material.

4.2 Structuring Transcript Content

Once we have segmented the visual content of the lecture, the next step is to organize the transcript text with respect to the extracted visual entities. We leverage temporal correspondences between the transcript and visuals to distinguish between explanatory and depictive sentences (Section 3) and to break long text descriptions into shorter, more readable paragraphs.

Aligning transcript to video. To obtain the temporal alignment between the transcript and video, we use an automatic algorithm by [Rubin et al. 2013] which extends the Penn Phonetics Lab Forced Aligner (P2FA) built on the HTK speech recognition software. This aligner takes a verbatim transcript and an audio file as inputs and outputs a time-stamped transcript, where each word is annotated with a start and end time.

Detecting explanatory versus depictive sentences. As discussed in Section 3, depictive sentences typically coincide with drawing actions while explanatory sentences do not. Using the time-aligned transcript, we compute correspondences between transcript sentences and visual entities. A sentence is matched to a single visual entity if most of its utterance time ($\geq 75\%$) overlaps with the drawing time of the visual entity. If a sentence does not coincide with any entity, we refer to it as an unmatched sentence. We classify all matched sentences as depictive text (associated with the corresponding visual entities) and all unmatched sentences as explanatory text. Note that while this is a heuristic, we found it to work well in practice. We use this information in the layout stage to reduce clutter and make the text more readable.

Breaking up long text descriptions. In some cases, complex vi-

sual entities that contain a lot of information may get matched with large blocks of depictive text. When reading such text blocks, it can be hard to identify and follow all the correspondences between the individual sentences and the relevant parts of the figure. We address this problem by breaking up complex visual entities into sub-entities, each of which has a shorter, more readable block of depictive text.

In particular, we use a variant of the stroke segmentation algorithm described in the previous section to further segment a complex visual entity e. In this case, we use the following scoring function F_{sub} to evaluate a set of candidate sub-entities, V_{sub}:

$$F_{\text{sub}}(V_{\text{sub}}) = \sum_{e_{\text{sub}} \in V_{\text{sub}}} \lambda_1 |n_{\text{words}}(e_{\text{sub}}) - w| + P_{\text{o}}(V_{\text{sub}}) \qquad (11)$$

where $n_{\text{words}}(e_{\text{sub}})$ is the number of words in the depictive text associated with sub-entity, e_{sub}; P_{o} is the overlap between bounding boxes of sub-entities in V_{sub} (defined in Equation 7); w is the target number of words in the depictive text for each sub-entity; and λ_1 determines the relative importance of the word count and overlap terms. We set $w = 50$ (about 2-4 sentences) and $\lambda_1 = 1/25$. Using this scoring function, we apply the same dynamic programming procedure described in Section 4.1 to segment e into sub-entities. In this variant, we only allow consecutive strokes to be grouped together since our goal is to obtain temporally sequential sub-entities. Figure 1c shows an example output from this optimization.

4.3 Layout and Formatting

We organize the visual and audio content into a static, sequential format by interleaving visual entities with blocks of transcript text in the order of their appearance in the video. As we point out in section 4.1, a single visual entity can be composed of non-contiguous groups of strokes. For example, in Figure 4, e_1 and e_2 each consist of 4 separate groups of strokes, (1&2, 4, 6, 8) and (3, 5, 7, 9) respectively. In this case, we show each contiguous group of strokes at its associated time, together with previous strokes in the same visual entity which are shown for context. For example Figure 4 would be presented as: 1&2, 3, (1&2)&4, (3)&5 etc., where the parentheses indicate previous strokes. The new group of strokes is highlighted with color on top of the previous strokes (Figure 8).

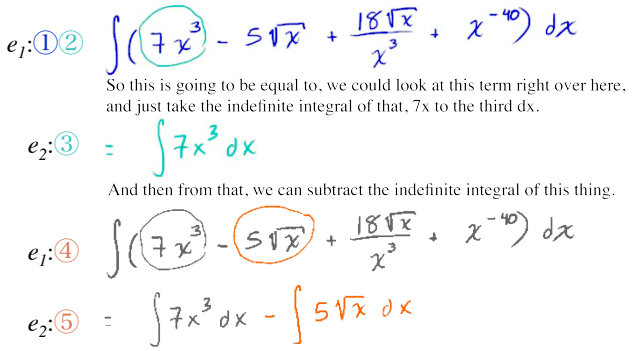


Figure 8: *Visual Transcript presentation of strokes 1-5 of Figure 4. Each contiguous group of strokes is shown together with previous strokes in the same visual entity.*

By default, all visual entities and explanatory sentences are shown; the depictive text associated with visual entities is hidden to reduce clutter. Users can click on the *expand* buttons next to individual visual entities to display the corresponding depictive sentences. For complex visual entities, the expanded view shows the decomposed sub-entities with their associated depictive sentences (Figure 1c).

5 Results

The final output of our method is a Visual Transcript, a readable and printable representation of both the visual and audio content of a lecture video. Since a Visual Transcript contains all of the visual and audio information from the input video, it can be used by itself to study the content. Alternatively, it can be linked to the original lecture video to function as an interactive navigation aid. Similar to NoteVideo [2013], clicking on a visual entity or transcript sentence plays the video at that point in time. As the video is played, the corresponding visual entity and/or transcript sentence is highlighted.

We have used our system to generate 20 Visual Transcripts based on math and physics. 10 of those videos were taken from Khan Academy, and the others were from 10 different instructors on YouTube. Figure 9 shows a subset of our results. Please view the supplementary material for additional examples. The rest of this section highlights some of the key features of Visual Transcripts.

Linear format highlights lecture progression. The layout of text and visual entities in Visual Transcripts often emphasizes the instructor's thought process and clarifies the intermediate steps that lead to a result. Figure 10 (left) compares equations in the final view of the blackboard at the end of the lecture to our Visual Transcript. Although the blackboard view shows the same set of equations, it is difficult to infer how the equations relate to and build upon each other. Our Visual Transcript shows a step-by-step progression of the visual content.

Interspersing text with visuals clarifies connections. A purely visual summary of the video omits verbal explanations, whereas a purely textual summary (i.e., standard transcript) can be confusing without the corresponding visuals. Instead, Visual Transcripts interleave explanatory text and visual entities. This makes it easy to see the connection between illustrations, or the context of an illustration. For instance, compare the leftmost example in Figure 7, which shows a final view of the blackboard and Figure 10 (right) the Visual Transcript for the same video. In the former, it is difficult to see the connection between the illustration (pink highlight) and the equation to its right (green highlight) without listening to the lecture. In the latter, the text in-between explains clearly that the equation represents the vector field depicted in the illustration.

Different levels of detail. By default, visual transcripts hide redundant depictive text that just describes the corresponding visuals. If a reader wants to see more details, she can reveal the hidden text by clicking on the visual entity. In the case of a long equation or a complicated illustration, the expanded view breaks up the visual and textual information into easy-to-read blocks (Figure 1c).

6 User Evaluation

We performed a comparative study to test the hypothesis that Visual Transcripts facilitate learning. We compared three interfaces to study video lectures: a standard YouTube player with an interactive text transcript (Baseline), the NoteVideo interface [Monserrat et al. 2013], and our Visual Transcript interface linked to the video (Figure 11). The YouTube video player is currently the most common viewing interface for online lectures, and NoteVideo while less established, was specifically designed to facilitate navigation of blackboard-style lecture videos. In NoteVideo, a panoramic image of the board with strokes from the entire lecture serves as an in-scene navigation interface. Users can click on any stroke to play the video at that point in time.

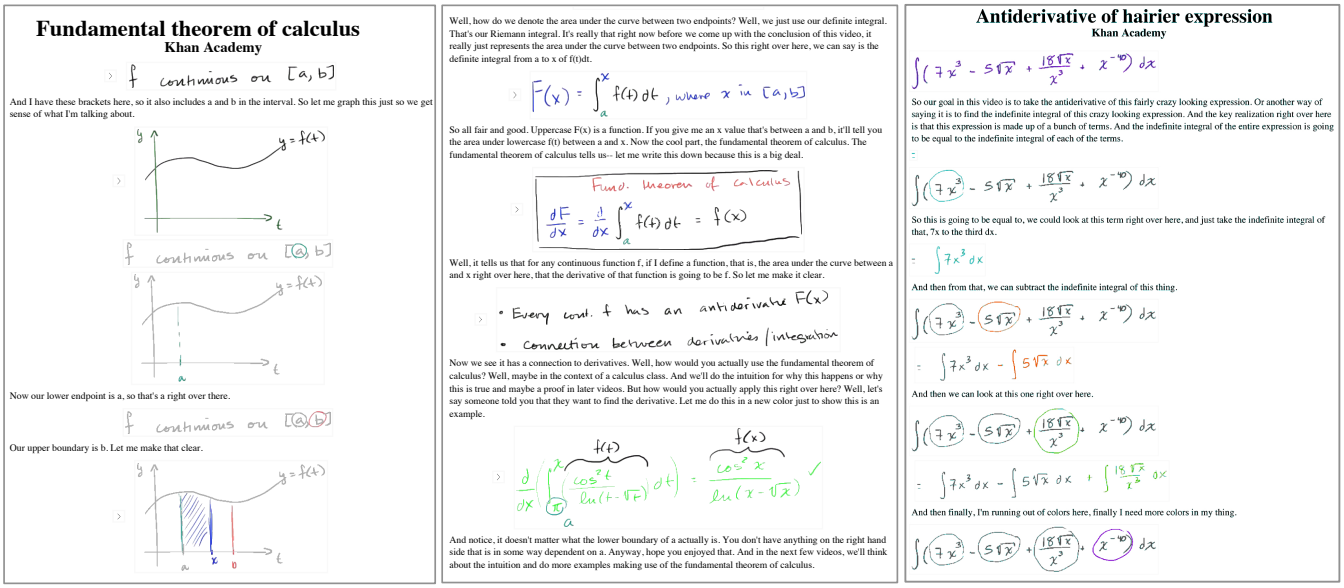


Figure 9: *Examples of Visual Transcripts from two different lectures. Our output interleaves verbal explanations with corresponding visual contents written on the board. For example, the sequence of the visual contents, which is ambiguous in Figure 2, becomes clear in our output (left and middle).*

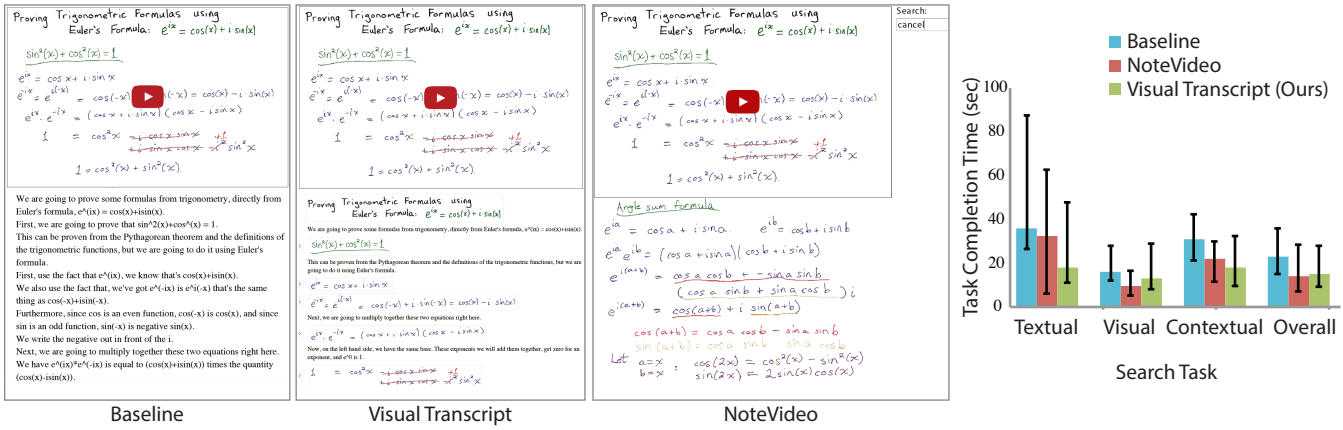


Figure 11: *(left) We compared three interfaces to study video lectures: A standard YouTube player with an interactive text transcript, our Visual Transcript, and NoteVideo. (right) Graph shows median search task completion time, where error bar represents interquartile range.*

Tasks. Our study includes two tasks: (1) summarization, to get a quick and comprehensive overview of the lecture without watching the entire video, and (2) search, to quickly locate specific information. Although not a direct measure of learning, these tasks are inherent activities in learning and also match common evaluation tasks used in the literature on tools for lecture videos [Kim et al. 2014a; Pavel et al. 2014; Monserrat et al. 2013].

- In the **summarization** task, users have to quickly provide an overview of the lecture without watching it entirely. We gave users only 3 minutes to view and summarize 7-8 minute long lectures. We purposely did not give enough time to watch the entire video so as to motivate the users to quickly scan through its content. Before the task, users watched a sample lecture video and read a sample summary comprised of main points and detail points. Users were encouraged to try to write down at least all the main points of the lecture, and as many of the detail points as possible. We compared the summaries written by users to a gold standard list of main/detail points manually created by two referees. The user summaries were scored by the number of points they covered.

- The **search** task emulates scenarios when the user wants to quickly find a specific piece of information in the video (e.g. to solve a question in a problem set, or to look up a specific formula). We differentiate three different types of search problems depending on whether the information is in the visuals, the transcript or a combination of both. The *visual search* reproduces situations when a user remembers something visually and wants to find where it appeared. (E.g., *Find the point in the lecture where the instructor strikes out part of an equation, where terms add up to eliminate each other.*) For the *textual search*, the cue is often a word or a phrase that could be found in the transcript either directly or indirectly (E.g., *Find the point in the lecture where the property that every continuous function has an antiderivative is stated.*) For the *contextual search*, the information is neither in the text nor visuals alone, but rather in the context between the two.

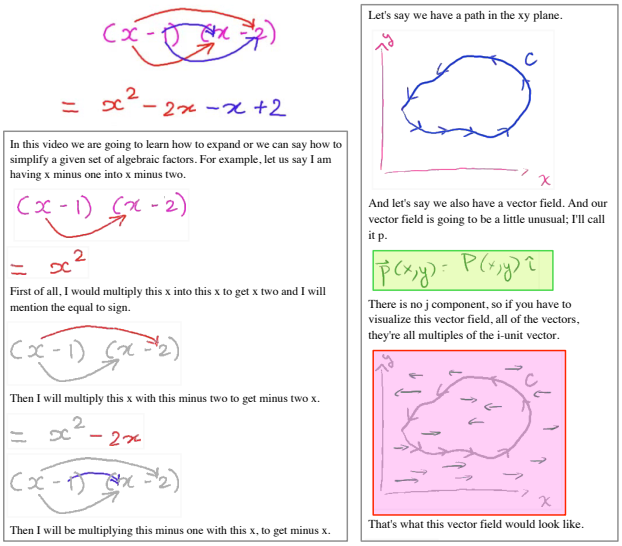


Figure 10: *(Left, bottom) Visual Transcript shows the step-by-step progression of an equation which is not apparent in the (left, top) final visual of the board. (Right) Interspersing text with visuals clarifies the connection between the illustration of a vector field and its equation. Compare with the leftmost example in Figure 7.*

	Baseline	NoteVideo	Ours
Main points	0.83±0.12	0.81±0.21	0.87±0.18
Detail points	0.50±0.22	0.56±0.18	0.58±0.15

Table 1: *Percentage of points covered by user summaries compared to the golden standard list.*

(E.g., *Find the point in the lecture where the instructor writes an integral expression for a bounded area under some curve.*) User performance was assessed by the task completion time as well as correctness.

Protocol. Nine participants (2 female, 7 males), ages 20 to 35 took part in our study. All of them were familiar with the general subject matter of the lectures, although they had not seen the particular lectures before. We chose three college-level math lectures for our study: *Fundamental Theorem of Calculus* by Salman Khan (8 minutes), *Proving Trigonometry Formulas Using Euler's Formula* by Lee Stemkoski (7.2 minutes), and *Uniform Distribution* by Actuarialpath (8 minutes).

We used a within-participant design, where each participant performed tasks on each interface. We counter-balanced the order of the interfaces and the assignment of videos to interfaces using a Latin Square. Before using each interface, participants were briefed about their features and given time to familiarize themselves. After each task, they answered questions about their interaction with the interface. After completing all tasks, participants completed a questionnaire on their preference and the usability of each interface. Please refer to the supplementary material for the full set of tasks and post-task questionnaires.

6.1 Findings and Discussion

There are several notable findings from our user study:

1. Users write more comprehensive summaries with Visual Transcripts.
Users listed the most number of main and detail points using our interface, although differences across the interfaces were not statistically significant according to the one-way analysis of variance (ANOVA) (main points: $F_{2,24} = 0.23$, $p = 0.79$, detail points: $F_{2,24} = 0.48$, $p = 0.62$). Table 1 shows the percentage of main/detail points covered by user summaries with each interface. Note that while on average, there may not seem to be a significant difference between ours and the two alternatives, summary quality varied significantly depending on the video. In particular, when the sequence of lecture was not clear in the panoramic image, NoteVideo users mixed the order of points or missed a main point entirely. For example, in the lecture on *Fundamental Theorem of Calculus* (Figure 2), NoteVideo users immediately clicked on the "Fundamental Theorem" (④) skipping the first third of the lecture about the graph of a continuous function and the area under it (①-③). While users performed comparably with Ours or the baseline, when asked which interface they preferred for the summary task, they preferred NoteVideo (5/9) and ours (4/9).

2. Users find information involving text faster with Visual Transcripts than with NoteVideo or the baseline.
For the *text search* and the *contextual search* users performed fastest with Visual Transcript followed by NoteVideo and then the baseline (Figure 11, *right*), although the differences across the interfaces were not statistically significant according to the non-parametric Kruskal-Wallis Test ($\chi^2 = 0.82, p = 0.676$). For these tasks, users either had to find relevant text or find a visual and also look at the text around it (or listen to the audio). Visual Transcripts naturally support such tasks by interleaving text and figures. NoteVideo does not provide a text to skim through, but users could search for key words or phrases (a feature also provided in Visual Transcripts and the baseline). Alternatively, they could click on a visual and listen. Interestingly, the baseline performed worst on these tasks, despite the fact that it is most text-centered and provides the exact same text as Visual Transcripts. This is likely because the text in the baseline was unstructured and difficult to read (see finding 3).
For the *visual search* users performed fastest with NoteVideo. For all videos we tested, NoteVideo had the advantage of presenting all the visuals in one screen, which made it easier for users to scan the entire visual content without having to scroll. On average, participants' performance on the search task was comparable on ours and NoteVideo, which was better than the baseline. The difference between ours and the baseline was statistically significant with the Mann-Whitney U (MWU) test ($Z = -2.1$, $p = 0.02$), whereas the difference between ours and NoteVideo was not ($Z = 1.2$, $p = 0.12$). Occasionally, users missed the information in their first search attempt and then tried to scan the entire lecture, contributing to a large variance.
In terms of accuracy, users were most successful in locating the correct information with Visual Transcripts (average error rate $e = 0.06$) compared to NoteVideo ($e = 0.07$) or the baseline (0.15), although the differences were not statistically significant (ANOVA, $F_{2,24} = 1.04, p = 0.15$).

3. Visual Transcripts make the transcript text easy to read and skim through.
Both the baseline and Visual Transcripts include the entire transcript text. However, the usefulness of their transcripts is rated very differently. On a 1-7 usefulness scale, Visual Transcript scored 6.3 (range: 5 to 7), whereas baseline scored 4.7 (range: 1 to 7). With the baseline, participants mostly scrubbed through the timeline to complete the tasks. Several users (3/9) mentioned that the baseline transcript text was difficult to skim through or find correspondences with the video. In contrast, with the Visual Transcript, users primarily relied on the text and visuals to solve the tasks (rather than the video). One user commented that the layout was *"similar to a textbook"* and *"easy to read"*. Another user said that *"the paragraph structure corresponds to the main points"* which facilitates skim-

ming.

4. Users prefer Visual Transcripts for learning.
The post-task survey showed that, for learning in general, most users (7/9) preferred our interface over NoteVideo (2/9) or the baseline. The reasons for preferring Visual Transcript included *"having the whole script and equations [visuals]"* and *"a good balance between getting the overview and some more detail."* Those who preferred NoteVideo appreciated having all visual content presented *at once* without having to scroll, but also noted that the sequence was not apparent (e.g., many users asked where to click to get to the beginning of the lecture) and that *"there's a risk of missing something that's not written on the board."*

7 Limitations

Our implementations focused on blackboard-style lectures, but the key ideas behind the design of Visual Transcript (e.g., presenting discrete visual entities next to its corresponding narrative in a linear layout) are generalizable to other style of lecture videos. Different styles of lecture videos would require different or more sophisticated visual entity recognition and layout techniques. For example, a classroom recording might have human occlusion, and a slide-based presentation could have animation or other multimedia effects.

The pre-processing step for stroke extraction works especially well with digital blackboard-style lectures with constant background and minimal occlusion. However, videos with more noise (e.g., from lighting change or occlusion by hand) may require a more sophisticated method. We also do not handle special cases such as partial erasure or copy-and-paste. For instance, if the instructor updates a part of visual content in order to correct a mistake, the current implementation only shows the *newest* stroke.

Although in all of our examples the segmentation algorithm outputs results that produce comprehensible Visual Transcripts, we also observed 2 types of failure cases: (1) *under-segmentation*, where too many strokes are grouped into a single visual entity, and (2) *over-segmentation*, where related strokes are separated into different visual entities. Our scoring function assumes a layout where distinct visual entities are more or less spatially separate from each other. A different method may be required to handle videos that violate this assumption, for example a history lecture where most of the writing is on top of a map, or where figures are overlayed on top of each other (Figure 13). An editing or annotation mechanism, including crowdsourcing, to aid segmentation would be useful and a potential area for future work.

In placing temporally aligned visuals and sentences next to each other, we assume that instructors talk about what they are drawing at the same time. This assumption holds in most cases, but fails to resolve other types of references. For example, in Figure12, the pronoun 'here' is used twice, each time referring to a different part of the visual. Whereas in the video these references become clear with the cursor movement, they remain ambiguous in our static output.

8 Conclusion

This paper introduced Visual Transcripts, a readable and interactive representation of blackboard-style lecture videos, which interleaves visual content with corresponding text. We use a variant of the classic line-breaking algorithm to segment the visual content of a lecture video into discrete figures. Then, we leverage the temporal correspondence between the figures and transcript sentences to structure the transcript text. Finally, we interleave the figures with corresponding text in an easy-to-read format. Our small user evaluation suggests that compared to a standard video player and a state-of-the-art interface for watching blackboard-style lectures, users prefer our interface for learning. It also suggests that our interface is effective in helping users browse or search through lecture videos.

Over *here* you'll get positive fluxes.
[…]
But *here*, you'll get a negative flux.

Figure 12: *Layout using only temporal correspondence fails to resolve some references. In this example, 'here' in the first sentence refers to the top right portion of the boundary R, whereas the second 'here' refers to the bottom left portion. In the video, these references are clarified by pointing with a cursor.*

Figure 13: *Our segmentation algorithm assumes that distinct visual entities are more or less separate from each other. For example, a history lecture where most of the writing is on top of a map, or where figures are overlayed on top of each other (Figure 13) may require a different method.*

Acknowledgements

We would like to thank Sylvain Paris, Gaurav Chaurasia, Yichang Shi, Fadel Adib and Michael Gharbi for their helpful feedback and discussions. We also thank the authors of NoteVideo for making their implementations available for our comparative user study. This project is partially funded by Quanta Computer, Inc., Royal Dutch Shell, and Samsung Scholarship.

References

AGNIHOTRI, L., DEVARA, K. V., MCGEE, T., AND DIMITROVA, N. 2001. Summarization of video programs based on closed captions. In *Photonics West 2001-Electronic Imaging*, International Society for Optics and Photonics, 599–607.

BARNES, C., GOLDMAN, D. B., SHECHTMAN, E., AND FINKELSTEIN, A. 2010. Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics (TOG) 29*, 4, 89.

BORECZKY, J., GIRGENSOHN, A., GOLOVCHINSKY, G., AND UCHIHASHI, S. 2000. An interactive comic book presentation for exploring video. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 185–192.

BORGO, R., CHEN, M., DAUBNEY, B., GRUNDY, E., JANICKE, H., HEIDEMANN, G., HOFERLIN, B., HOFERLIN, M., WEISKOPF, D., AND XIE, X. 2011. A survey on video-based graphics and video visualization. In *Proc. of the EuroGraphics conf., State of the Art Report*, Citeseer, 1–23.

CHI, P.-Y., AHN, S., REN, A., DONTCHEVA, M., LI, W., AND HARTMANN, B. 2012. MixT: automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th an-*

nual ACM symposium on User interface software and technology, ACM, 93–102.

CHI, P.-Y., LIU, J., LINDER, J., DONTCHEVA, M., LI, W., AND HARTMANN, B. 2013. DemoCut: generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM, 141–150.

CHOUDARY, C., AND LIU, T. 2007. Summarization of visual content in instructional videos. *Multimedia, IEEE Transactions on 9*, 7, 1443–1455.

CHRISTEL, M. G., AND WARMACK, A. S. 2001. The effect of text in storyboards for video navigation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 3, IEEE, 1409–1412.

CHRISTEL, M. G., HAUPTMANN, A. G., WACTLAR, H. D., AND NG, T. D. 2002. Collages as dynamic summaries for news video. In *Proceedings of the tenth ACM international conference on Multimedia*, ACM, 561–569.

CHUN, B.-K., RYU, D.-S., HWANG, W.-I., AND CHO, H.-G. 2006. An automated procedure for word balloon placement in cinema comics. In *Advances in Visual Computing*. Springer, 576–585.

EKIN, A., TEKALP, A. M., AND MEHROTRA, R. 2003. Automatic soccer video analysis and summarization. *Image Processing, IEEE Transactions on 12*, 7, 796–807.

HE, L., SANOCKI, E., GUPTA, A., AND GRUDIN, J. 1999. Auto-summarization of audio-video presentations. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, ACM, 489–498.

HU, Y., KAUTZ, J., YU, Y., AND WANG, W. 2015. Speaker-following video subtitles. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 11*, 2, 32.

HWANG, W.-I., LEE, P.-J., CHUN, B.-K., RYU, D.-S., AND CHO, H.-G. 2006. Cinema comics: Cartoon generation from video stream. In *GRAPP*, 299–304.

JACKSON, D., NICHOLSON, J., STOECKIGT, G., WROBEL, R., THIEME, A., AND OLIVIER, P. 2013. Panopticon: A parallel video overview system. In *proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM, 123–130.

KIM, J., GUO, P. J., CAI, C. J., LI, S.-W. D., GAJOS, K. Z., AND MILLER, R. C. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM, 563–572.

KIM, J., NGUYEN, P. T., WEIR, S., GUO, P. J., MILLER, R. C., AND GAJOS, K. Z. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM, 4017–4026.

KNUTH, D. E., AND PLASS, M. F. 1981. Breaking paragraphs into lines. *Software: Practice and Experience 11*, 11, 1119–1184.

KURLANDER, D., SKELLY, T., AND SALESIN, D. 1996. Comic chat. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, 225–236.

LARGE, A., BEHESHTI, J., BREULEUX, A., AND RENAUD, A. 1995. Multimedia and comprehension: The relationship among text, animation, and captions. *Journal of the American Society for Information Science 46*, 5, 340–347.

LI, F. C., GUPTA, A., SANOCKI, E., HE, L.-W., AND RUI, Y. 2000. Browsing digital video. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 169–176.

LU, Z., AND GRAUMAN, K. 2013. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, IEEE, 2714–2721.

MONSERRAT, T.-J. K. P., ZHAO, S., MCGEE, K., AND PANDEY, A. V. 2013. NoteVideo: Facilitating navigation of blackboard-style lecture videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1139–1148.

MYNATT, E. D., IGARASHI, T., EDWARDS, W. K., AND LAMARCA, A. 1999. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 346–353.

NGO, C.-W., MA, Y.-F., AND ZHANG, H.-J. 2005. Video summarization and scene detection by graph modeling. *Circuits and Systems for Video Technology, IEEE Transactions on 15*, 2, 296–305.

PAVEL, A., HARTMANN, B., AND AGRAWALA, M. 2014. Video digests: a browsable, skimmable format for informational lecture videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM, 573–582.

PICKERING, M. J., WONG, L., AND RÜGER, S. M. 2003. ANSES: Summarisation of news video. In *Image and Video Retrieval*. Springer, 425–434.

RUBIN, S., BERTHOUZOZ, F., MYSORE, G. J., LI, W., AND AGRAWALA, M. 2013. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM, 113–122.

SHAH, D., 2014. "MOOCs in 2014: Breaking down the numbers (edsurge news)".

SHAHRARAY, B., AND GIBBON, D. C. 1995. Automatic generation of pictorial transcripts of video programs. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, International Society for Optics and Photonics, 512–518.

SHAHRARAY, B., AND GIBBON, D. C. 1997. Pictorial transcripts: Multimedia processing applied to digital library creation. In *Multimedia Signal Processing, 1997., IEEE First Workshop on*, IEEE, 581–586.

SMITH, M. A., AND KANADE, T. 1998. Video skimming and characterization through the combination of image and language understanding. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, IEEE, 61–70.

TRUONG, B. T., AND VENKATESH, S. 2007. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 3*, 1, 3.

UCHIHASHI, S., FOOTE, J., GIRGENSOHN, A., AND BORECZKY, J. 1999. Video manga: generating semantically meaningful video summaries. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, ACM, 383–392.