석 사 학 위 논 문
Master's Thesis

DynamicLecture: 슬라이드 기반 동영상 강의를 위한 객체 기반 재생 및 편집 시스템

DynamicLecture: Enabling Direct Revision of Slide-based Lecture Videos

2019

정 형 식 (鄭 亨 植 Jung, Hyeungshik)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

DynamicLecture: 슬라이드 기반 동영상 강의를 위한 객체 기반 재생 및 편집 시스템

2019

정 형 식

한 국 과 학 기 술 원

전산학부

DynamicLecture: 슬라이드 기반 동영상 강의를 위한 객체 기반 재생 및 편집 시스템

정 형 식

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2018년 12월 17일

심사위원장 김 주 호 (인)

심 사 위 원 오 혜 연 (인)

심 사 위 원 Mik Fanguy (인)

DynamicLecture: Enabling Direct Revision of Slide-based Lecture Videos

Hyeungshik Jung

Advisor: Juho Kim

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Daejeon, Korea
December 17, 2018

Approved by

Juho Kim
Professor of Computer Science

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MCS
20173553

정형식. DynamicLecture: 슬라이드 기반 동영상 강의를 위한 객체 기반 재생 및 편집 시스템. 전산학부 . 2019년. 25+iv 쪽. 지도교수: 김주호. (영문 논문)

Hyeungshik Jung. DynamicLecture: Enabling Direct Revision of Slide-based Lecture Videos. School of Computing . 2019. 25+iv pages. Advisor: Juho Kim. (Text in English)

초 록

효과적인 강의 비디오를 제작하기 위해서는 오류 수정, 내용 검토, 피드백 반영 등 촬영 이후에도 여러 차례의 수정이 필수적이다. 하지만 강의자들은 비디오를 수정하기 위한 시간과 비디오 편집 기술을 충분히 가지지 못하는 경우가 많다. 강의 비디오의 보다 쉬운 수정을 가능케 하기 위하여 본 논문에서는 강의자가 비디오의 내용을 바로 수정할 수 있는 객체 기반의 비디오 편집 시스템인 DynamicLecture를 제안한다. 본 편집 시스템은 원본 비디오에서 슬라이드와 슬라이드 내에 있는 글, 그림 및 스크립트를 추출한다. 사용자 실험에서 강의자와 학생들은 먼저 강의자의 비디오에서 개선할 점을 작성했다. 강의자들은 DynamicLecture을 이용해서 31%의 항목을 수정할 수 있었고, 27%의 항목을 시간이 더 주어질 경우 가능할 것으로 응답했다. 강의자와 전문 비디오 편집자 모두 DynamicLecture을 이용해서 강의 비디오를 빠르고 쉽게 수정할 수 있었다.

핵 심 낱 말 교육 비디오, 강의 제작, 비디오 편집

Abstract

Producing high-quality lecture videos requires multiple rounds of edits to the original recording, for example, fixing errors, revising content, or incorporating feedback. However, instructors often lack the time and skills to apply the edits to the videos. To facilitate easier editing of lecture videos, we developed DynamicLecture, an object-oriented video editing interface that allows users to directly update the content of slide-based lectures within the video. The editing interface is powered by a computational pipeline which extracts slides, texts, images, and transcribed speech from the raw video. In a user study, students and instructors identified a list of items to edit from the instructors' lecture video. Using DynamicLecture, instructors successfully edited 31% of the items and marked an additional 27% of the items as achievable with more time. Both instructors and professional video editors could edit lecture videos faster and easier using DynamicLecture compared to existing tools.

Keywords Educational videos; Lecture production; Video editing;

Contents

List of Tables

List of Figures

Chapter 1. Introduction

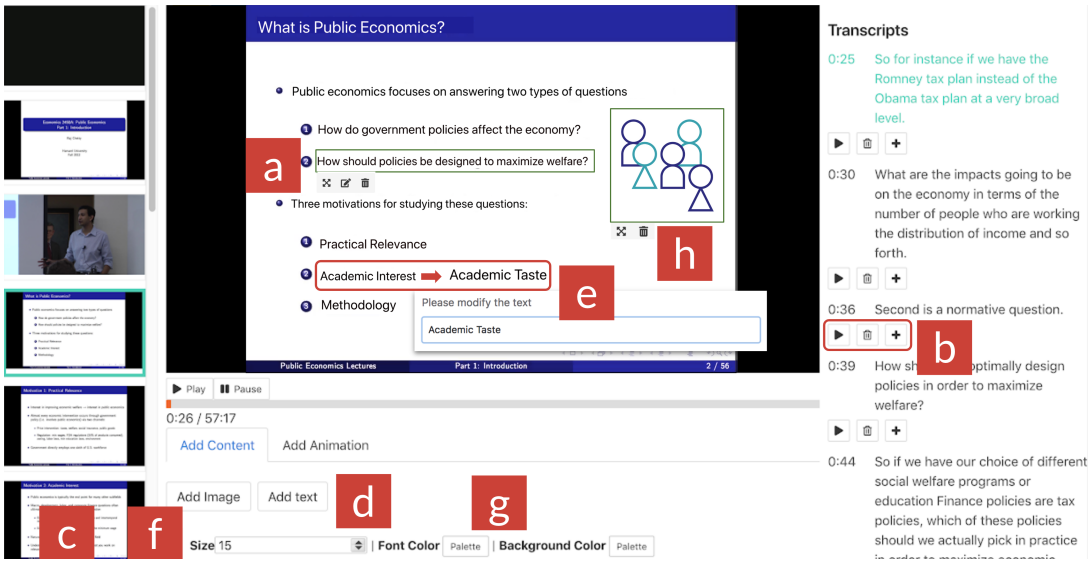

Figure 1.1. DynamicLecture enables revision of slide-based lecture videos without having to import video, overwrite existing content with new content, and re-render. Main features of DynamicLectureare: (a) Users can directly delete or (e) edit the content of text as well as style using (f) and (g); (b) Each sentence of the lecture has play and delete button as well as insert button for recording additional explanation (Figure 4.1); (c) Users can navigate to the beginning of each slide in the video; (d) Users can add a new image or text to the current slide, as well as emphasize the element in sync with explanation using animation (Figure 4.2); (h) Users can change the location and size of figure as well as delete it. To support these features, DynamicLecture detects text and image objects from the embedded slides of a slide-based lecture video with transcripts.

In online learning, video is a popular medium to deliver contents at scale in an engaging way. Producing high-quality lecture videos and keeping them up-to-date takes a lot of effort. Rather than publishing the initial recording from the lecture as-is, instructors often prefer to polish the video before distribution, for example, in order to fix errors or to clarify explanations. Similar considerations arise after the publication of a video. Learners leave feedback, instructors find errors, or the mere passage of time calls for updated explanations or examples, which require post-production and editing.

However, editing lecture videos remains a tedious and difficult task. For example, consider the simple case where an instructor wants to fix a typo from a lecture slide. First, the instructor needs to create a new visual with the corrected text, for example, by editing the original slide deck. Then, using a video editing software, the instructor needs to overlay the new visual on top of the original one, adjusting its placement and timing carefully to sync with the rest of the visuals as well as the audio recording. Finally, the video must be encoded, uploaded, and published again. The task becomes more complicated if both visuals and the related audio narration need editing. Regardless of one's proficiency with video editing software, the time and effort required to edit videos is a huge barrier for instructors

to update their videos. As a result, many videos are published with minimum polishing and remain without any updates to the content. Error corrections, learner feedback, and up-to-date examples are not incorporated, despite their potential to benefit future learners.

To remedy this situation by enabling efficient editing of lecture videos, we present DynamicLecture, an object-oriented video editing interface that enables in-video editing of slide-based lecture videos. We chose slide-based videos as a target format of the video as they are widely used and suitable to explore the feasibility and benefit of object-oriented editing. Based on the findings from interviews with instructors and professional editors, DynamicLecture supports users to directly manipulate and edit text and image objects in the lecture slide. Users can also edit the audio recording by deleting or re-recording sentences in the narration. In addition, users can add animation effects to emphasize visual objects in the slide in sync with the corresponding audio narration. DynamicLecture supports editing the video within the web browser, allowing users to edit the component of their lecture videos seamlessly with less effort.

Providing these object-level interactions requires parsing the original pixel-based video frames and audio into discrete visual and audio objects. To achieve this goal, we present a computational pipeline that segments the lecture video into slides, retrieves text and image objects from each slide, and transcribes the audio narration into sentences with start and end timestamps. For accurate parsing, we employ a human-machine hybrid workflow such that users can correct automatic parsing errors on-demand using a custom error correction interface (Figure 5.2). We test our pipeline against a database of 10 videos and confirm that our pipeline finds 77% of boundaries between slides, 72% of text segments in slides, and 93% of figures in slides.

In order to assess to what extent instructors can revise their own videos using DynamicLecture, we had video revision sessions with instructors where instructors listed up the revision items and revise their videos using the system. Results from the study demonstrate that DynamicLecture supports common revision scenarios (covers 58% of suggested revision items) and is easy to use (Figure 6.2). A second study with professional video editors also reveals that editing lecture videos using DynamicLecture takes less time than using existing video editing tools. The result suggests that DynamicLecture's object-oriented editing interface lowers the bar for video editing, which can potentially encourage the update and improvement of video lectures.

This paper makes the following contributions:

- **Design goals** for systems that aim to support efficient editing of lecture videos. These are identified from interviews with professional instructors and lecture video editors for online learning platforms.
- **DynamicLecture, an object-oriented video editing interface** that enables efficient editing of slide-based lecture videos.
- **A computational pipeline** that parses slide-based lecture videos into discrete visual and audio objects. To improve parsing accuracy, we use a human-machine hybrid workflow and provide a custom error correction interface.
- **Qualitative user evaluation** showing how object-level editing feature of DynamicLecture covers various cases of revising lecture video by instructors who are not familiar with video editing software.

Chapter 2. Related Work

Our work is based on previous research on interfaces that leverage the content of the informational video, systems that support MOOC instructors, and non-linear multimedia editing tools.

2.1 Content-level understanding and browsing support for informational video

Recent works proposed systems that extract metadata (text, image, script) from informational videos for application scenarios such as generating an alternative video format, helping video navigation, or adjusting the content of the video. Video Digests [1], SceneSkim [2], VideoDoc [3], and Visual Transcripts [4] utilizes the transcript and the visual content of the lecture to provide a more readable and structured video format. A thread of research utilized the content of the video to support non-linear navigation. NoteVideo [5], ViZig [6], and Codemotion [7] propose video browsing using objects in video, such as text segments [5], visual elements like charts and tables [6], or source code in the screencast [7]. MMToC [8], Yang et al. [9], and Zhao et al. [10] introduce techniques for video navigation using various level of cues, such as slide boundaries, topics, and keywords. The closest to our work is adjusting the content of the video, such as improving the legibility of text [11], highlighting the currently explained part of a video [12], or important parts of a video [13]. Beyond improving the watching experience, DynamicLecture uses video content for improving editing experience by converting the raw video into a set of objects that can be directly manipulated.

2.2 Instructor-supporting systems in online education

Building systems to help instructors overcome challenges and leverage opportunities in online education has been an active research topic. A line of works facilitated collecting feedback and comments from the students. MudSlide [14] collects confusing points of the online lecture by asking students to mark muddy parts of embedded slides. TrACE [15] and Video Collaboratory [16] provides asynchronous discussion space around in-video anchor points that can be used as an implicit source of feedback. In online learning, learners generate a massive amount of learning activity logs. These logs collected at scale can provide insights for instructors [17]. PeakVizor [18], VisMOOC [19], and DropoutSeer [20] suggest information visualization could help instructors easily analyze learner activities. DynamicProblem [21] enables instructors to easily create, deploy, and analyze randomized experiments on components of digital problems such as hints or feedback messages. While all these tools are built to help instructors collect feedback and understand the behavior of learners, DynamicLecture aims to lower the barrier for updating lecture videos so that instructors can easily incorporate insights from learners into their existing video.

2.3 Non-linear multimedia editing

In recent years, researchers have developed systems that leverage the discrete representation of audio and video content to facilitate editing, instead of conventional linear editing. Rubin et al. [22] and Shin et al. [23] presents a text-based editing workflow for audio editing. With the transcript, users can navigate and delete the segments in the audio recordings. Berthouzoz et al. [24] presents tools for editing interview videos using transcripts, and QuickCut [25] supports the production of a narrated video by matching video shots with the relevant part of narration scripts. Upon these lines of research, DynamicLecture extracts discrete structure of contents from the slide-based video to support efficient editing, especially for those who are not familiar with video editing.

Chapter 3. Current Practice for Editing Lecture Videos

We conducted a series of interviews with instructors and professional video editors to understand how they produce and edit online lecture videos as well as the challenges they face with existing tools and practices.

3.1 Formative interviews

We interviewed four university professors and three part-time instructors, all of whom had experience recording lectures for e-learning platforms. The university professors collaborated with professional editors working at the university to produce their lecture videos, while the part-time instructors recorded and edited the videos on their own. Each interview lasted about 30 minutes and was centered around the overall process of producing lecture videos, especially focusing on the editing practice.

In addition, we conducted a group interview with three professional video editors from a content production team at a university. The team was in charge of editing about 30 lecture videos per month. The group interview was structured similarly to the individual interviews and lasted for an hour.

3.2 Findings

Instructors want to edit the videos before and after publication. Before publishing their videos online, instructors wanted to polish them (Table 3.1). A common need was adding emphasis. While instructors can certainly add emphasis at the recording time (e.g., with digital ink, a physical pointer, or animated effects on the slides), they also wanted to apply emphasis after the recording, for instance by magnifying a visual item on the slide or by changing its color. Other scenarios included fixing typos in a slide, inserting subtitles to supplement the explanation, revising a verbal explanation, and adding sound effects (e.g., clapping) or visual effects (e.g., an exclamation point or question mark above the instructor's head) to make the video more engaging. Instructors expressed similar needs for editing their videos after publication.

Editing cost is prohibitive. Even when support was available from dedicated video editing professionals, the time and effort to edit videos was deemed considerable by both the instructors and the video editors. Communication between the instructor and the editor took a lot of time and several iterations. First, the editors produce a draft version of the video after recording the lecture. Then, the instructor or a TA watch the draft video and makes a list of items to be edited. Since editors are not as familiar with the flow or content of the lecture, instructors need to mark the precise location of the video and describe the edit in detail, for example, by using screenshots or extensive annotations. Even then, editors reported they often had follow-up questions to fully understand the instructors' intentions (e.g., having trouble referencing a specific part of a formula).

Instructors who did not have support from professional editors had to make their own edits. Instructors were not as proficient with video editing software and often did not know how to achieve the effect they wanted. Therefore, for editing tasks that go beyond simple cutting and pasting, they had to record that part of the lecture video again, which takes significant time.

	Slide	Audio	Organization
Add	- Add a table summarizing the content - Add caption to supplement explanation - Add icons (e.g., smile, question mark)	- Add sound effects (e.g., clapping sound)	- Add a virtual character - Add intro / outro clips - Add transition effects
Update	- Fix a typo - Change or resize a figure - Visually highlight the concepts	- Give another version of explanation - Fix mistake by re-recording	- Replace a set of slides
Delete	- Delete noise from the camera	- Delete sounds like 'ahem' - Trim unnecessary explanations	- Delete a set of slides

Table 3.1: Major editing operations for lecture videos noted from the interviews. We classified operations according to (1) where they are made and (2) how they change the original content. While 'Slide' and 'Audio' columns refer to changes made to individual slides and the audio respectively, 'Organization' shows editing operations that modify multiple slides or add effects throughout the video.

In general, the substantial time and effort required to make the edits meant that only severe errors were corrected and other revisions or polishing were left unattended. This was especially true when editing required additional recording.

Editing often involves substituting the original (slide) contents. Many of the editing operations we identified from our interviews (Table 3.1) involved updating a part of the original slide content, for instance, replacing text or a diagram from the slide with new content. Take the common scenario of replacing a text item in the slide to fix typos. This seemingly simple task can be very tedious in the current workflow. First, the editor creates an image of the new text, usually with the presentation authoring software used to create the original slides. Then, in the video editing software, the editor imports the new image and overlays it on top of the video frame showing the original slide. This involves adjusting the size and position of the image to cover the old text, while also matching a consistent style (e.g., font-size) with the rest of the slide. The start and end timing for the image also needs to be adjusted precisely, especially when animation effects are involved. Since conspicuous edits can draw the viewers attention unnecessarily, editors and instructors put a lot of effort into making the edited content look seamless with the original content. This kind of tuning requires editors to go back and forth between the slide authoring tool and the video editing software and takes up a significant portion of the total editing time.

3.3 Design goals

The interview results emphasize the need to provide users with an intuitive interface to directly edit the slide content and verbal explanations seamlessly within the video. Based on the findings above, we identified three high-level design goals for the design of DynamicLecture.

D1. Provide an intuitive interface for instructors to directly edit their videos. Currently, there is a significant gap between the edits that instructors *want* to make and the edits that are actually implemented. This is due to both the instructors' lack of time and proficiency with video editing software. Even when professional editors are available to help, the communication cost between the instructor and the editors is burdensome. To help instructors edit their own videos directly with ease, we designed DynamicLecture's interface to resemble existing tools (e.g., Microsoft PowerPoint, Google Slides) that instructors are already familiar with.

D2. Enable users to update elements of the lecture slide within the video. Current workflow for updating parts of the original lecture slides involves creating a new slide element, overlaying it on top of the video frame, adjusting its position, adjusting its timing, and re-rendering the whole video. These steps take a lot of time and often involve tedious fine-tuning. Instead, inspired by prior work on direct object manipulation in videos [26, 27], we enable users to update the content of the original slides directly within the video.

D3. Facilitate synchronization of visual and audio events in the video. In lectures, the timing of visual events is tightly linked to the timing of corresponding verbal explanations. For example, a bullet point in the slide is emphasized by changing its color when the instructor explains that point or a subtitle is displayed while the instructor explains a specific concept. Usually, precise time-alignment between visual and audio events require tedious fine-tuning. In DynamicLecture, we facilitate the synchronization by providing a time-stamped transcript and allowing users to link the timing of visual events to specific points in the transcript.

Chapter 4. DynamicLecture Interface

DynamicLecture's editing interface is built on three key features: (1) updating visual objects, (2) updating verbal explanations, and (3) emphasizing visual objects through animation effects.

4.1 Updating visual objects in the slide

Text and figures within the slides are presented as individual objects within the video frame, which users can directly update (**D2**). Users can add and delete text elements, modify its content, location, or style (font size, background color, and foreground color) (Figure 1.1(e)). For non-text elements such as figures, users can move, re-size, or delete existing figures (Figure 1.1(h)) as well as add a new figure (Figure 1.1(d)).

4.2 Updating verbal explanations

Users can also update verbal explanations using the sentence-by-sentence transcriptions provided by DynamicLecture (**D2**, Figure 1.1(b)). Users can delete a sentence in the transcript which deletes the corresponding audio. Users can also insert new sentences between existing sentences by making a new recording (Figure 4.1) which is automatically transcribed and inserted into the transcript. We chose sentence as the atomic unit for manipulating narrations since recording an entire sentence creates a more natural-sounding result and it is a more natural workflow for users than recording a few words within an existing sentence. The transcript allows users to interact with the audio track easily by manipulating the text.

Modifying the audio track requires updating the corresponding video frames as well. DynamicLecture treats videos as a set of audio segments (corresponding to each sentence) and their corresponding video frames. When an audio segment is deleted (Figure 4.3(a)), the system deletes corresponding video frames. (Figure 4.3(b)). When a new audio segment is inserted, the last frame from the previous segment is displayed during the newly inserted segment (Figure 4.3(c)).

4.3 Emphasizing visual objects through animation effects

Finally, users can add animation effects to emphasize objects in the slide. The timing of the emphasis or animation effect is closely linked to the corresponding verbal explanation (i.e., an object should be emphasized when it is being explained) (**D3**). This type of synchronization between audio and visual events requires tedious fine-tuning in traditional video editing tools. To facilitate the process, DynamicLecture allows users to specify the timing of animation effects in the transcript. To apply an animation effect, users first select an object, choose the animation type (Appear, Magnify or Highlight), and then specifies the start and end time by clicking on the buttons between two words in the transcript (Figure 4.2). The animation timing is linked to the corresponding part of the transcript, such that the object appears emphasized while that part of the transcript is played in the video. This link is maintained even if users modify the transcript (and thus the audio and the corresponding video), for

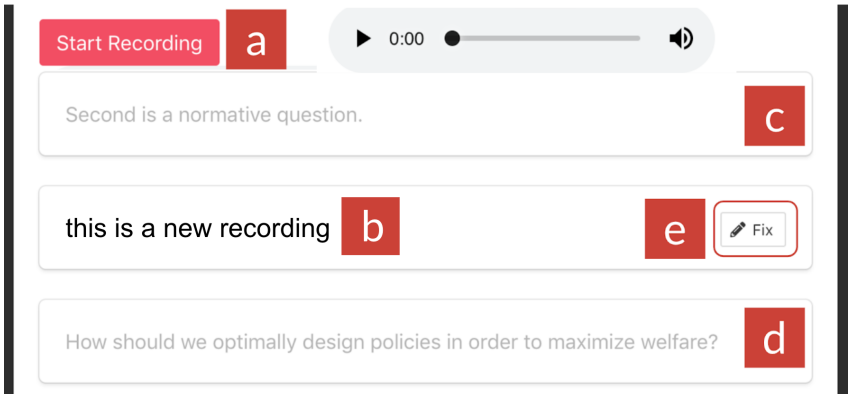

Figure 4.1: Users can record and insert new verbal explanation to the video using the audio recording interface of DynamicLecture. The script for the previous (c) and the next (d) sentences relative to the user-specified location is displayed to provide context. After the user records the new explanation (a), it is transcribed and displayed (b). The user can correct the transcription result (e).

example, by adding or deleting a sentence around it. If the users delete the linked part of the transcript, the animation effect is deleted together.

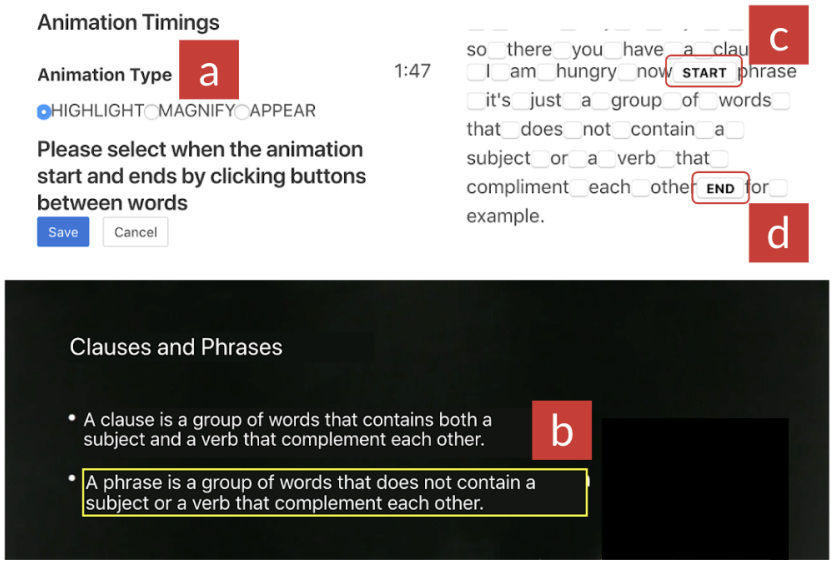

Figure 4.2: Users can emphasize objects in a slide in sync with the selected verbal explanation. Users can select the type of emphasis effect (a), object to be emphasized (b), and specify the timing of the emphasis from the transcript (c) (d).

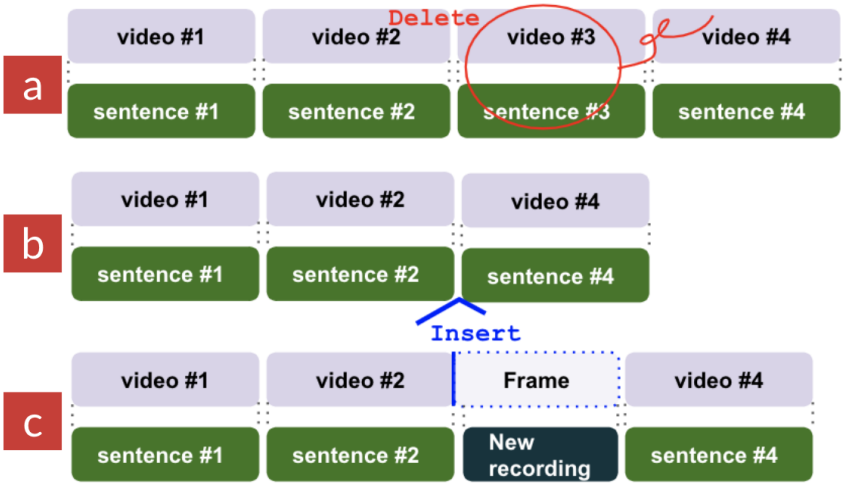

Figure 4.3: DynamicLecture manages videos as a set of audio segments for each sentence and their corresponding video segments. (a) When the user deletes a sentence , (b) the corresponding video segment is removed as well. When the user deletes ⊑ a new audio segment, the last frame of the previous video segment is displayed for the new recording (c).

Chapter 5. Computational Pipeline

To support the object-oriented interactions described above, DynamicLecture's computational pipeline parses an input video and extracts an object-level representation. This allows users to edit existing videos without relying on the original slide document. Specifically, DynamicLecture takes a slide-based video as an input and returns a set of unique slides, a set of text items and figures in each slide, and the timing and text representation of the audio narration. The pipeline consists of three main parts: (1) slide boundary detection, (2) text segmentation within slides, and (3) object (figure, table) detection within slides. We also obtain sentence-level text representations of the audio narration (Figure 5.1).

5.1 Stage 1: Slide boundary detection

The goal of this stage is to find the boundaries between video frames that represent unique slides. This allows us to extract the set of slides used in the lecture, along with the start and end time for each slide. We measure the frame difference using two criteria: (1) the edge-based frame difference [28], and (2) the Levenshtein distance [29] between the all the words extracted from each frame using Google Cloud Vision API [30]. If the frame difference is above a threshold, we identify it as a slide boundary.

5.2 Stage 2: Text segmentation within slides

The goal of this stage is to group words in a slide into a set of semantic units, such as a phrase, a sentence, or a single item in a bullet point list (Figure 5.2(b)). For this, we use the bottom-up slide layout analysis by Zhao et al. [10].

Using the text segments from the pipeline, DynamicLecture renders the text over the original text of lecture video. Instead of rendering only the modified text of the slide, we chose to re-render every text of the slide to achieve visual consistency between the modified text and the original one. We get the color and background color of the text by first taking the two most frequent color values inside the bounding box, and regard the color at the corner of the box as the background color of the text.

5.3 Stage 3: Object detection within slides

The goal of this stage is to detect non-text objects such as figures, tables, and charts (Figure 5.2(c)). We trained the Inception Resnet v2 network [31] using the Tensorflow Object Detection API [32]. For training, we collected 3,314 lecture slide PowerPoint slide from the web (a total of 84,444 slides). We used 80% of these as the training set and the remaining 20% as the test set. We obtained the object bounding box information from PowerPoint's XML data. DynamicLecture does not distinguish different types of visual objects but treats all objects as either text or figure.

Similar to the text objects, DynamicLecture re-renders non-text objects in the slide by rendering the cropped images of detected objects over the original objects.

5.4 Audio transcription

The goal of this stage is to parse the audio narration into sentence-level text representations and to obtain word-level timings. We use Google Cloud Speech-to-Text API [33] to obtain transcripts with word-level timing and punctuation. Then, we use NLTK's sentence tokenizer to parse the text into sentences.

5.5 Performance evaluation

We evaluate the accuracy of each stage of the pipeline on a data set of 10 slide-based lecture videos for which we manually created ground-truth labels. Please refer to the supplementary material for the URL of each video and the detailed test results.

For stage 1, we evaluated whether the pipeline correctly determined each frame as a slide boundary. Our pipeline achieved F1-score of 77% for this task.

For stage 2, we evaluated the recall, or how many text segments from the ground truth were detected by the pipeline. Our pipeline retrieves 480(72%) out of 694 text segments from the ground truth data set. [1]

For stage 3, our object detection model achieves mAP@ [.5:.95] [34] of 0.408. Applying an IoU threshold of 0.8 (intersection over union between the detected area and the true object area) and a confidence threshold of 0.5, the model achieves F1 score of 93% for finding figures in our test set.

Our computational pipeline is optimal for parsing lecture videos with full-screen digital slides without animation effects. Other types of video input may produce erroneous results. For instance, if the instructor appears in the frame, she could be parsed as a figure object. Or if animation effects create enough difference within a single slide, the pipeline may detect multiple slide boundaries. Since such errors from the pipeline can hurt the users' editing experience significantly, we provide an easy-to-use interface where users can verify and fix the pipeline results.

5.6 Human correction of the automatic results

Figure 5.2 shows the error correction interface. Users can verify and correct results from each stage of our computational pipeline. For instance, users can change or remove a detected slide boundary, re-group text segments, adjust the size and location of the detected figures, or identify a new figure.

[1] We evaluate the recall rate rather than the F1-score because higher precision with less false positives (wrong results with few larger segments) does not necessarily mean better results than lower precision with more false positives. (wrong results with many small segments).

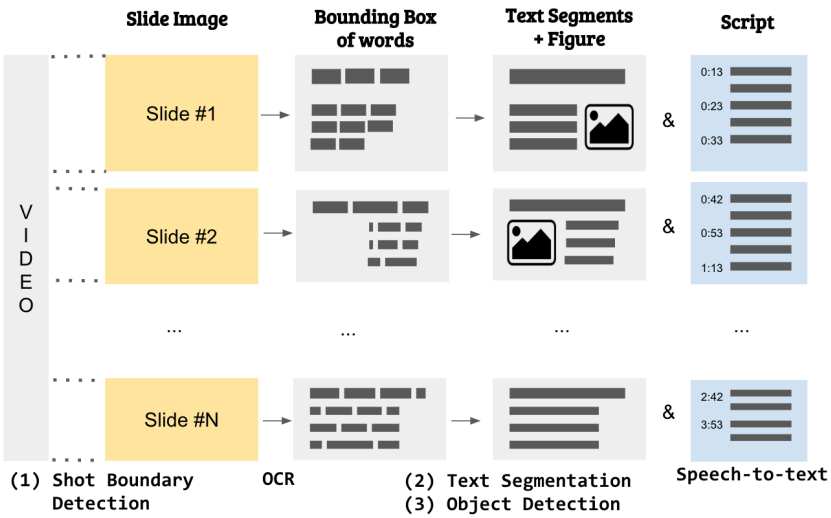

Figure 5.1: An overview of DynamicLecture's computational pipeline. Given an input video, stage 1 detects unique slides with a shot boundary detection algorithm, stage 2 finds text segments in each slide by grouping words together, and stage 3 finds figures in the slides.

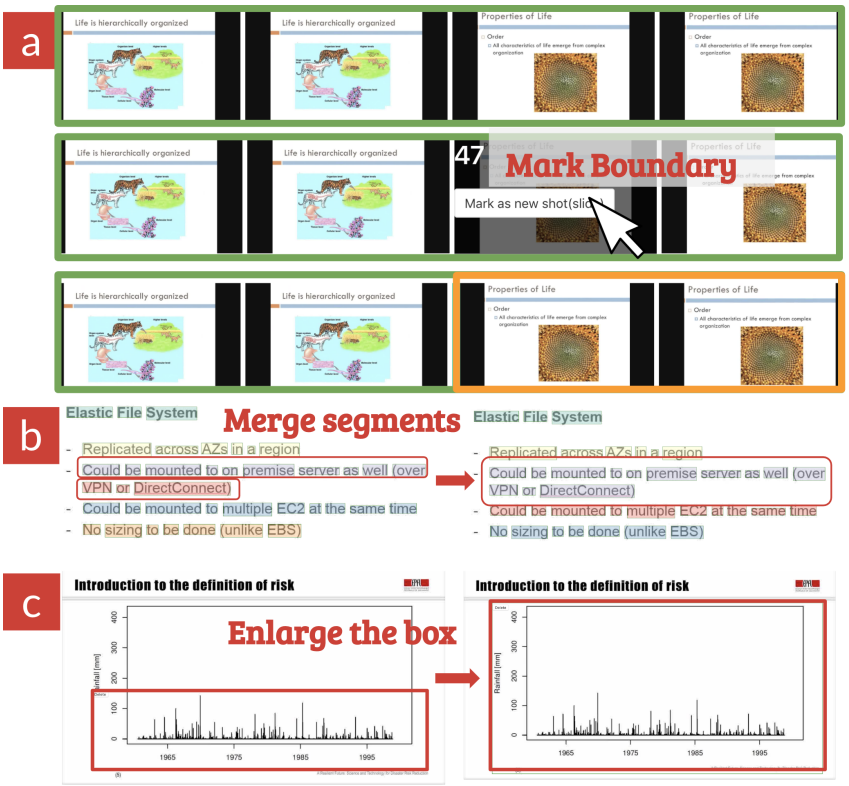

Figure 5.2: UI for checking and correcting the result of DynamicLecture's video processing pipeline. Users can adjust (a) the boundary frame between slides, (b) the text segmentation result, and (c) the figure detection result.

Chapter 6. Evaluation

To assess how DynamicLecturefacilitates revision of slide-based lecture videos, we conducted a video revision session with 6 instructors. We explored the following two research questions:

RQ1. To what extent can DynamicLecture's object-oriented editing features support instructors' various revision needs?

RQ2. What are the potential benefits of editing lecture videos with DynamicLecture compared to a conventional video editing workflow?

6.1 Participants

Six instructors took part in the study (5 males). 4 of them were university professors, one was a graduate student, and one was a software engineer who recorded lecture videos part-time (Table 6.1). We asked each instructor to share one of their lecture videos. All the videos used slides and were less than 20 minutes. Two of them also included instructors' headshot in the video. Two of the videos were in English and four were in Korean.

6.2 Study setup

In order to let instructors focus on the editing experience, one of the authors imported instructors' videos into DynamicLecture and corrected any errors from the computational pipeline, including the audio transcription results.

To simulate students' comments or feedback from an online learning platform, we recruited ten students for each video from a university bulletin board and asked them to give feedback about the video. Students were asked tag each feedback with a time-stamp (to the corresponding part in the video) and the type of editing operation involved ((Table 3.1: slide, audio or organization). Each student was asked to leave at least five comments for a video. We assigned a video to a student only if the student responded as having prior knowledge of the lecture topic.

6.3 Procedure

The evaluation session lasted for 60-80 minutes for each instructor. Four sessions were conducted online using video chat and screen sharing while two sessions were done in-person. Each session was composed of two main stages:

Instructors list revision items for their lecture videos. After filling out a background questionnaire, instructors watched their lecture video and listed up revision items that could improve the quality of their lecture. For each revision item, they were asked to specify the location in the video, how they wanted to revise, and why. Afterward, we showed instructors the feedback from the students and asked them to update their original revision list.

Instructors modify their video using DynamicLecture. After finalizing the revision list, we gave instructors a 10-minute tutorial of DynamicLecture. We then asked them to apply as many revision

	Lecture experience			Proficiency			Video Revised		
Participant	**Offline**	**Online**	**Editing experience**	**P[a]**	**V[b]**	**S[c]**	**Topic**	**Length**	**Type**
P1	12 years	7 courses	2 courses	7	6	3	Linguistics	8:36	Slide + Lecturer
P2	3 years	8 videos	8 videos	5	2	3	Blockchain	16:45	Slide + Voice
P3	half a year	10 videos	10 videos	7	2	7	Algorithm	11:39	Slide + Voice
P4	1 year	20 videos	2 videos	7	4	5	Programming	13:40	Slide + Voice
P5	20 years	200 videos	60 videos	7	5	5	English Writing	10:26	Slide + Lecturer
P6	20 years	100 videos	10 videos	5	4	5	Programming	15:37	Slide + Screencast

Table 6.1: Participants' demographics and the video they revised. Instructor reported their proficiency in using (a) Presentation, (b) Video editing, (c) Screen capture softwares with 7-point Likert scale (1-Cannot use, 7-Proficient).

items as possible within 20 minutes using DynamicLecture. Instructors marked each revision item with one of the following three labels: *done* (successfully applied the revision item using DynamicLecture), *possible* (seems possible with DynamicLecture but didn't execute due to time limit), and *impossible*. Afterward, instructors filled out a questionnaire to provide feedback about their experience.

6.4 Results

6.4.1 RQ1: DynamicLecture supported 58% of revision items.

Instructors listed a total of 85 revision items (mean=13.8, stddev=11.26, min=7, max=36), including 38 items that were added after reviewing the feedback from students. Out of the 85, instructors completed 27 (31%) items using DynamicLecture and marked an additional 23 (27%) items as possible with more time. Table 6.2 illustrates how instructors edited their videos and what operations DynamicLecture could not support.

Modifying the text and image within slides: The most common usage of DynamicLecture's editing feature was to supplement the slide content. For example, instructors added an new example image (P2, Figure 6.1(a)), added an additional line to an equation (P3), or added an extra line illustrating the computation of a function(P4, Figure 6.1(b)). Another common pattern was modifying the content, such as changing the title to convey the main message of a slide instead of just keywords (P3, P5, Figure 6.1(c),(d)). Instructors also fixed typos in their slide (P1), and made their handwritten text more visible by overlaying typed-text (P4). Instructors also used the animation feature to emphasize items in sync with the corresponding verbal explanation (P3, P5, P6).

For certain items, instructors expressed the need for more detailed editing features. For example, DynamicLecture supports updating the style of an entire sentence, but sometimes they wanted to change the style of a part of the sentence. Supporting revision of other types of objects, such as a table, an equation, and a handwritten text was another need that DynamicLecturedid not support.

Modifying the verbal explanation Four instructors modified their explanations using DynamicLecture. The goal was to remove unnecessary explanations (P3, P4), fix a mistake (P5), improve fluency (P1) and supplement existing explanation (P3). Instructors found the text transcription convenient for checking and revising their explanations.

While DynamicLecture supports modifying explanations at the sentence level, some instructors expressed the need for word-level editing of audio. Instructors also wanted to improve the quality of

	Slide	Audio	Organization
No. Done	21 (37.5%)	6 (27.3%)	0
No. Possible	13 (23.2%)	10 (45.4%)	0
No. Impossible	22 (39.3%)	6 (27.3%)	7 (100%)
Done or Possible items	- Change the title of slide - Remove unnecessary visuals - Overwrite handwriting with new text - Add subtitle to supplement verbal explanation	- Supplement explanation - Delete unnecessary explanation	
Impossible items	- Highlight a word (not the whole segment) - Underline a word - Zoom a part of slide - Create and insert a table - Edit handwritten text	- Delete a part of sound (murmuring, noise, silence) - Insert a new recording between sentences	- Add / delete a slide - Change the slide template - Split the video

Table 6.2: Editing operations supported and not supported by DynamicLecture

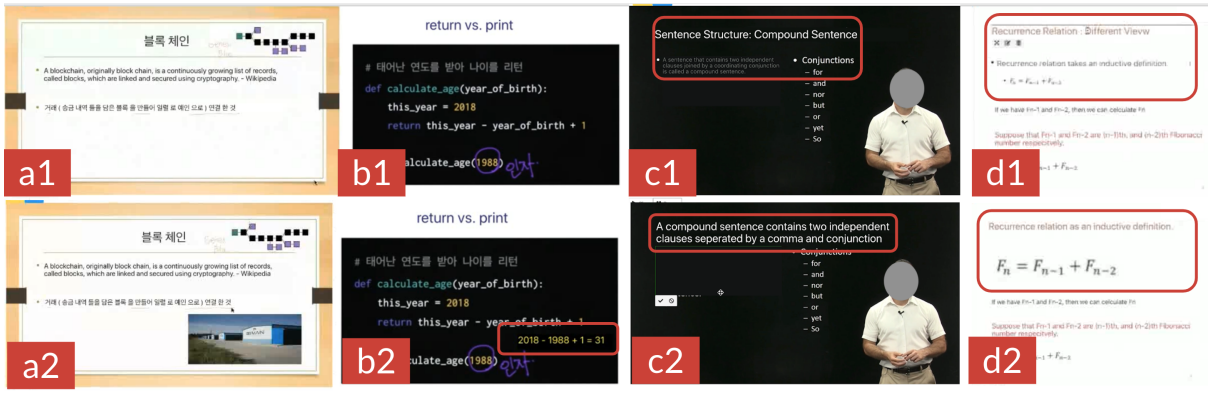

Figure 6.1: Examples of edited slides by instructors using DynamicLecture

the sound as well as the content, for example, by reducing noise or removing typing sounds from the keyboard.

6.4.2 RQ2: DynamicLecture empowers instructors to modify their lecture videos.

Overall, participants responded positively to the simplicity of DynamicLecture's editing features (Figure 6.2). Previously, some users were not familiar with the concept of revising lecture videos: *"When I watch my own video, it's not satisfactory but I had no way of updating it."* (P6). All appreciated the ease of modifying the slide content: *"I didn't dare to change the content of my lecture video because the only video editing I could do was trimming and inserting clips using iMovie. So I had to choose between recording the part of my lecture again or not"* (P3). Instructors found the DynamicLecture useful in their workflows: *"Role of the instructor in lecture video production has expanded"* (P1), *"It's hard to ask editors to fix minor issues in my video because I know they are already busy"* (P4).

Meanwhile, instructors were concerned about the detailed quality of the edited videos. One specific concern was whether their newly recorded voice would sound different from the original recording: *"Conceptually it's cool, but if the quality issue remains I will record the video again"* (P5).

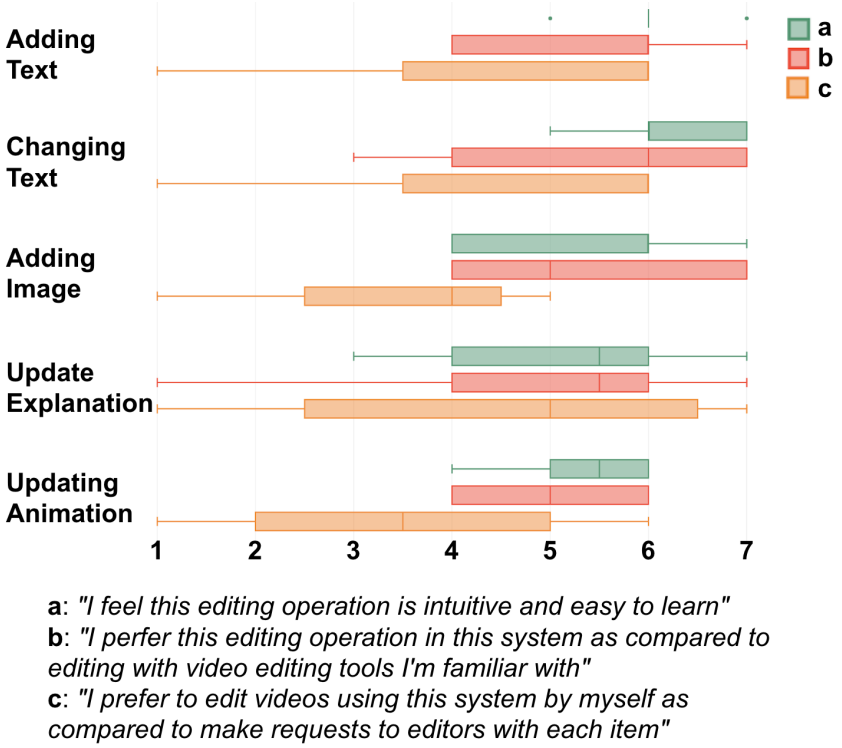

Figure 6.2: Responses to the three 7 likert-scale questions about each feature of DynamicLecture. (1: No, 7: Yes)

6.5 Comparison with the conventional video editing workflow

We conducted hands-on sessions of DynamicLecture with two professional editors, in order to compare the editing workflow with conventional video editing tools. Video editors were working in a content production team of a university and had also participated in the formative interview.

We prepared a sample video and six editing tasks composed of modifying the slide content (4 tasks), deleting a sentence in the lecture (1 task), and emphasizing a sentence in the slide in sync with its verbal explanation (1 task). For each item, we gave the location of video and a detailed description for editing (e.g., Please add this picture to the slide starting at 3:24), similar to instructions they would receive from instructors. We first asked the editors to do editing tasks using the tool they normally use (both used Adobe Premiere). Then we gave a 10-minute tutorial of DynamicLecture and asked them to the same task with our system. We measured the time taken for each task under two conditions. After finishing the editing tasks, we conducted a semi-structured interview with instructors. The session took an hour for each editor.

Among the six tasks, both editors finished four tasks faster using DynamicLecture, while P1 took more time with two tasks (Table 6.3) because of their unfamiliarity with our interface. During the interview, editors reacted positively to the ability to directly modify existing text or images in the slide. In conventional editing workflow, editors have to create the new text or images in order to put them on existing video, which takes a lot of time. For DynamicLecture's animation feature, two editors expressed opposite opinions. P2 preferred conventional timeline-based animation because of its fine-grained control, but P1 preferred DynamicLecture's emphasis feature with the convenient link to the timings in the transcript.

	Editor 1		**Editor2**	
Task	**Base**	**Ours**	**Base**	**Ours**
Deleting a sentence	0:48	**0:26**	0:57	**0:21**
Enlarge a figure	3:13	**0:15**	0:60	**0:26**
Enlarge a sentence during explanation	0:39	**0:33**	3:25	**3:11**
Change a title of slide	**0:21**	1:06	1:24	**0:20**
Change a color of text	1:06	**0:25**	1:57	**0:35**
Add a picture	**0:16**	0:40	0:15	0:15

Table 6.3: Comparison between time took for editing with conventional tool and ours

Chapter 7. Dicussion and Future Work

7.1 The editing capability of the system

The current prototype of DynamicLecture is best suited for modifying lecture videos composed of static slides. It is less suited for videos with frequent visual changes such as slide lectures with a lot of animation effects, videos where instructors make annotations over slides, or screencast videos. Detecting diverse kinds of objects other than text or figure (e.g., handwriting, instructor) will enable object-oriented editing for more various type of lecture videos.

In the evaluation study, instructors listed up some operations that require organizational, multi-slide changes and failed to apply them using DynamicLecture (Table 6.2, 'Organization' column). While our current design intentionally focuses on supporting direct edits that could be simply made on top of existing videos, in the next iterations of the system, we will investigate the feasibility of supporting instructors' diverse needs including making organizational changes.

7.2 The quality of edited video

The detailed quality of edited lecture video has room for improvements which can affect the video watching experience of learners. DynamicLecture renders the text object in the slide using a single type of font in a single color. Therefore text on a non-plain background is not rendered smoothly (e.g., background with patterns) and typography from the original video is lost. Incorporating font detection technology [35] and image in-painting technology [36] into DynamicLecture will enable more seamless object-oriented video editing.

Recordings with DynamicLecture also have similar smoothness issues because the new recording can sound dissimilar to the original recording. This problem could be relieved if we leverage the speech synthesis technology like Tacotron [37] or VoCo [38] by extracting and applying the acoustic characteristic of original recordings to the new recordings.

7.3 Availability of quality feedback on lecture videos

During user evaluation, instructors could list the revision items and revise their video by themselves. However, usually they don't have time to watch their lecture videos again, and detailed feedback about the lecture is not always available. Providing instructors with the computed complexity of lecture [39], or letting students participate in the update of the video can reduce the effort of instructors to find revision items.

Chapter 8. Conclusion

In contrast to the offline lecture where the instructor can adjust the content of the lecture each time, it takes a lot of effort to change the content of lecture videos online. From interviews with instructors and professional editors, we found common editing operations and challenges in the lecture video production. We built DynamicLecture, an object-oriented video editing interface that supports editing text and images in the lecture, modifying verbal explanations, and emphasizing the content of lecture in sync with the explanation. From the evaluation, instructors could apply revision items they found to their own video with DynamicLecture.

Bibliography

[1] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. Video digests. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*, pages 573–582, New York, New York, USA, 2014. ACM Press.

[2] Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. SceneSkim. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pages 181–190, New York, New York, USA, 2015. ACM Press.

[3] Rebecca P Krosnick, Electrical Engineering, Computer Science, Partial Fulfillment, Electrical Engineering, Computer Science, Technology September, Rebecca P Krosnick, Electrical Engineering, and Computer Science. VideoDoc : Combining Videos and Lecture Notes for a Better Learning Experience by. pages 1–93, 2015.

[4] Hijung Valentina Shin, Floraine Berthouzoz, Wilmot Li, and Frédo Durand. Visual transcripts. *ACM Transactions on Graphics*, 34(6):1–10, oct 2015.

[5] Toni-Jan Keith Palma Monserrat, Shengdong Zhao, Kevin McGee, and Anshul Vikram Pandey. NoteVideo. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 1139, New York, New York, USA, 2013. ACM Press.

[6] Kuldeep Yadav, Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Saurabh Srivastava, and Om Deshmukh. ViZig: Anchor Points based Navigation and Summarization in Educational Videos. In *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16*, pages 407–418, New York, New York, USA, 2016. ACM Press.

[7] Kandarp Khandwala and Philip J Guo. Codemotion. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale - L@S '18*, pages 1–10, New York, New York, USA, 2018. ACM Press.

[8] Arijit Biswas, Ankit Gandhi, and Om Deshmukh. MMToC. In *Proceedings of the 23rd ACM international conference on Multimedia - MM '15*, pages 621–630, New York, New York, USA, 2015. ACM Press.

[9] Haojin Yang and Christoph Meinel. Content Based Lecture Video Retrieval Using Speech and Video Text Information. *IEEE Transactions on Learning Technologies*, 7(2):142–154, apr 2014.

[10] Baoquan Zhao, Shujin Lin, Xiaonan Luo, Songhua Xu, and Ruomei Wang. A Novel System for Visual Navigation of Educational Videos Using Multimodal Cues. In *Proceedings of the 2017 ACM on Multimedia Conference - MM '17*, number July, pages 1680–1688, New York, New York, USA, 2017. ACM Press.

[11] Andrew Cross, Mydhili Bayyapunedi, Dilip Ravindran, Edward Cutrell, and William Thies. VidWiki. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14*, pages 1167–1175, New York, New York, USA, 2014. ACM Press.

[12] Shoko Tsujimura, Kazumasa Yamamoto, and Seiichi Nakagawa. Automatic Explanation Spot Estimation Method Targeted at Text and Figures in Lecture Slides. In *Interspeech 2017*, volume 2017-Augus, pages 2764–2768, ISCA, aug 2017. ISCA.

[13] Xiaoyin Che, Haojin Yang, and Christoph Meinel. Automatic Online Lecture Highlighting Based on Multimedia Analysis. *IEEE Transactions on Learning Technologies*, 11(1):27–40, jan 2018.

[14] Elena L. Glassman, Juho Kim, Andrés Monroy-Hernández, and Meredith Ringel Morris. Mudslide. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 1555–1564, New York, New York, USA, 2015. ACM Press.

[15] Brian Dorn, Larissa B Schroeder, and Adam Stankiewicz. Piloting TrACE. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*, pages 393–403, New York, New York, USA, 2015. ACM Press.

[16] Vikash Singh, Sarah Abdellahi, Mary Lou Maher, and Celine Latulipe. The Video Collaboratory as a Learning Environment. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16*, pages 352–357, New York, New York, USA, 2016. ACM Press.

[17] Juho Kim, Phu Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. pages 4017–4026, 2014.

[18] Qing Chen, Yuanzhe Chen, Dongyu Liu, Conglei Shi, Yingcai Wu, and Huamin Qu. PeakVizor: Visual Analytics of Peaks in Video Clickstreams from Massive Open Online Courses. *IEEE Transactions on Visualization and Computer Graphics*, 22(10):2315–2330, oct 2016.

[19] Conglei Shi, Siwei Fu, Qing Chen, and Huamin Qu. VisMOOC: Visualizing video clickstream data from Massive Open Online Courses. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 159–166. IEEE, apr 2015.

[20] Yuanzhe Chen, Qing Chen, Mingqian Zhao, Sebastien Boyer, Kalyan Veeramachaneni, and Huamin Qu. DropoutSeer: Visualizing learning patterns in Massive Open Online Courses for dropout reasoning and prediction. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 111–120. IEEE, oct 2016.

[21] Joseph Jay Williams, Anna N. Rafferty, Dustin Tingley, Andrew Ang, Walter S. Lasecki, and Juho Kim. Enhancing Online Problems Through Instructor-Centered Tools for Randomized Experiments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, New York, New York, USA, 2018. ACM Press.

[22] Steve Rubin, Floraine Berthouzoz, Gautham J Mysore, Wilmot Li, and Maneesh Agrawala. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, pages 113–122, New York, New York, USA, 2013. ACM Press.

[23] Hijung Valentina Shin, Wilmot Li, and Frédo Durand. Dynamic Authoring of Audio with Linked Scripts. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, pages 509–516, New York, New York, USA, 2016. ACM Press.

[24] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics*, 31(4):1–8, jul 2012.

[25] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. QuickCut: An Interactive Tool for Editing Narrated Video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, pages 497–507, New York, New York, USA, 2016. ACM Press.

[26] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, page 237, New York, New York, USA, 2008. ACM Press.

[27] Laurent Denoue, Scott Carter, Matthew Cooper, and John Adcock. Real-time direct manipulation of screen-based videos. *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion - IUI '13 Companion*, page 43, 2013.

[28] Don Adjeroh, M C Lee, N Banda, and Uma Kandaswamy. Adaptive Edge-Oriented Shot Boundary Detection. *EURASIP Journal on Image and Video Processing*, 2009:1–13, 2009.

[29] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[30] Google. Google cloud vision api, 2018. https://cloud.google.com/vision.

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. feb 2016.

[32] Tensorflow. Tensorflow object detection api, 2018. https://github.com/tensorflow/models/tree/master/research/object_detection.

[33] Google. Google cloud speech-to-text api, 2018. https://cloud.google.com/speech-to-text/.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, may 2014.

[35] Guang Chen, Jianchao Yang, Hailin Jin, Jonathan Brandt, Eli Shechtman, Aseem Agarwala, and Tony X. Han. Large-Scale Visual Font Recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3598–3605. IEEE, jun 2014.

[36] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image Inpainting for Irregular Holes Using Partial Convolutions. pages 1–23, apr 2018.

[37] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards End-to-End Speech Synthesis. In *Interspeech 2017*, volume 2017-Augus, pages 4006–4010, ISCA, aug 2017. ISCA.

[38] Zeyu Jin, Gautham J Mysore, Stephen Diverdi, Jingwan Lu, and Adam Finkelstein. VoCo. *ACM Transactions on Graphics*, 36(4):1–13, jul 2017.

[39] Frans Van der Sluis, Jasper Ginn, and Tim Van der Zee. Explaining Student Behavior at Scale. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale - L@S '16*, pages 51–60, New York, New York, USA, 2016. ACM Press.

Curriculum Vitae in Korean

이 름: 정 형 식

생 년 월 일: 1991년 03월 27일

전 자 주 소: hyungshike@gmail.com

학 력

2007. 3. – 2010. 2. 용인한국외국어대학교부설고등학교 (3년 수료)

2010. 3. – 2017. 2. 서울대학교 자유전공학부 (컴퓨터공학, 정보문화학 전공) (학사)

2017. 3. – 2019. 2. 한국과학기술원 전산학부 (석사)

경 력

2017. 3. – 2019. 2. 한국과학기술원 전산학부 조교

연 구 업 적

1. **Jung, Hyeungshik**, Hijung Valentina Shin, and Juho Kim. "DynamicSlide: Reference-based Interaction Techniques for Slide-based Lecture Videos." In The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings, pp. 23-25. ACM, 2018.
2. Chang, Minsuk, Léonore V. Guillain, **Hyeungshik Jung**, Vivian M. Hare, Juho Kim, and Maneesh Agrawala. "RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale." In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, p. 451. ACM, 2018.