VidAdapter: Adapting Blackboard-Style Videos for Ubiquitous Viewing

ASHWIN RAM, NUS-HCI Lab, Department of Computer Science, National University of Singapore, Singapore
HAN XIAO, School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom
SHENGDONG ZHAO, NUS-HCI Lab, Department of Computer Science, National University of Singapore, Singapore and CNRS@CREATE LTD, Singapore
CHI-WING FU, Institute of Medical Intelligence and XR, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong

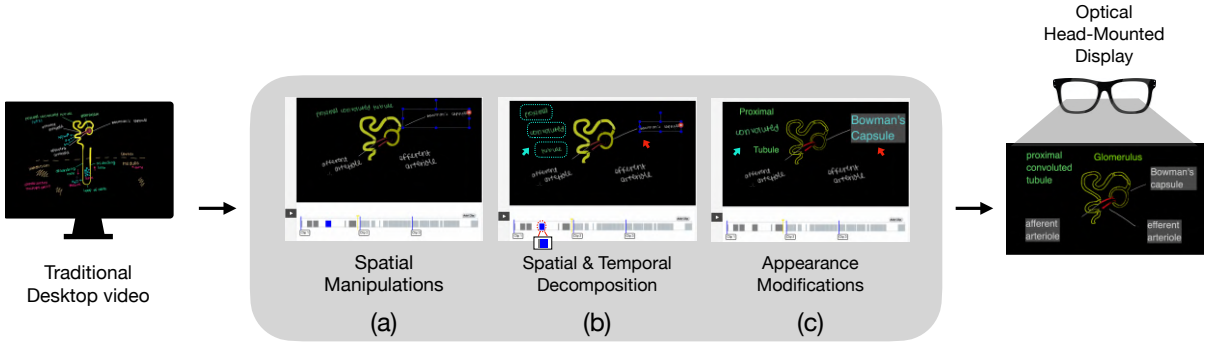


Fig. 1. The VidAdapter tool facilitates the adaptation of blackboard lecture videos for ubiquitous viewing on various mobile and wearable platforms by performing (a) direct spatial manipulations on automatically extracted video elements to reorganise their layout and reduce clutter (b) spatial (content indicated by the cyan arrow is broken down into smaller units) and temporal (content indicated by the red arrow is further segmented in time) decomposition for fine-grained adaptation of elements, and (c) modifications to the visual style of the extracted text and image elements.

Video lectures are increasingly being used by learners in a ubiquitous manner. However, existing video designs are not optimised for ubiquitous use, creating the need to adapt the style of these videos to meet the constraints of the learning platform and context of use. Our formative study with experienced video editing users, however, found that performing

Authors' addresses: Ashwin Ram, ashwinram@u.nus.edu, NUS-HCI Lab, Department of Computer Science, National University of Singapore, 15 Computing Dr., Singapore, 117418; Han Xiao, hanxiao701@gmail.com, School of Electronic Engineering and Computer Science, Queen Mary University of London, Peter Landin Building, Mile End Rd, Bethnal Green, London, United Kingdom, E1 4FZ; Shengdong Zhao, zhaosd@comp.nus.edu.sg, NUS-HCI Lab, Department of Computer Science, National University of Singapore, 15 Computing Dr., Singapore, 117418 and CNRS@CREATE LTD, 1 Create Way, 08-01 CREATE Tower, Singapore, 138602; Chi-Wing Fu, cwfu@cse.cuhk.edu.hk, Institute of Medical Intelligence and XR, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

2474-9567/2023/9-ART119
https://doi.org/10.1145/3610928

these adaptations using traditional video editors can be a challenging and time-consuming task. We developed VidAdapter, a tool that facilitates lecture video adaptation by allowing direct manipulation of the video content. For this, VidAdapter automatically extracts meaningful elements from the video, enables spatial and temporal reorganisation of the elements, and streamlines the modification of an element's visual appearance. We demonstrate the capabilities and specific use cases of VidAdapter within the domain of adapting existing blackboard lecture videos for on-the-go learning on Optical Head-Mounted Displays. Our evaluation of the tool with experienced video editing users revealed that VidAdapter was strongly preferred over traditional approaches and can improve the efficiency of the adaptation process by over 53% on average.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Content Adaptation; Ubiquitous Learning; Video Lectures; Smart Glasses

ACM Reference Format:
Ashwin Ram, Han Xiao, Shengdong Zhao, and Chi-Wing Fu. 2023. VidAdapter: Adapting Blackboard-Style Videos for Ubiquitous Viewing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 119 (September 2023), 19 pages. https://doi.org/10.1145/3610928

1 INTRODUCTION

Video has become a popular medium for gaining knowledge and learning new skills, with as many as two million users learning through MOOC platforms like Khan Academy every month [31]. With the mobility offered by mobile platforms, an increasing number of users have taken their learning sessions on the go [26, 30], creating the need to support this new ubiquitous learning paradigm.

However, many existing videos are created for devices with larger displays and used in stationary scenarios with dedicated attention. Such videos, when viewed on smaller-screen devices and on-the-go multitasking situations, become less understandable, significantly reducing the learning outcome [9, 16, 38]. Recent works have shown that such a reduction of learning outcomes can be mitigated by altering the presentation style of videos to suit the viewing platform and context of use [15, 26]. For example, a recent work in UbiComp [26] proposed to adapt existing videos to a layered serial visual presentation (LSVP) style so that blackboard-style lecture videos can be more easily viewed on wearable platforms like Optical-Head Mounted Displays (OHMDs). In particular, they found the need to adapt the layout and visual appearance of the video content to take advantage of the see-through properties of the OHMDs. Similarly, Kim et al. proposed to adapt existing videos for improved viewability on mobile phones [16].

While adapting a video is possible given the capabilities of current professional video editors, our formative study with experienced video editing users ($n = 4$) suggests it is a tedious and challenging task. In particular, we asked these users to adapt a blackboard lecture video for the case of viewing on OHMDs using their preferred video editor. We found that adapting a video requires the collaborative effort of multiple external tools/platforms as no single tool offers all the required services. Furthermore, the existing tools are not designed to support video adaptation, making it tedious and time-consuming, taking experienced video editing users an average of 26 minutes to adapt even a 1.5-minute clip.

In this work, we present VidAdapter, a tool specifically designed to facilitate the workflow needed to adapt blackboard-style lecture videos into a format suitable for ubiquitous learning. VidAdapter achieves this by automatically extracting meaningful video content and their temporal location as directly manipulable elements. While element-based manipulation has been proposed previously for purposes such as video navigation [10, 21] and more recently to perform simple in-video modifications to element's size and position [15], VidAdapter extends their work by supporting more complex adaptation workflows in both spatial and temporal dimensions. In particular, VidAdapter allows users to respecify the constituents of the extracted elements, adapt their visual appearance, and realign their time and display duration to better support various ubiquitous viewing requirements in different platforms.

We evaluated VidAdapter with experienced video editing users ($n = 12$) who compared the tool's capabilities with their traditional approach for the task of adapting blackboard lecture videos for viewing on OHMDs. Note that we consider adaptations needed for OHMD-viewing as it is inclusive of adaptations needed for viewing on other ubiquitous devices such as phones (refer Table 1). Our results indicate that VidAdapter was strongly preferred by users and more efficient for the adaptation process than traditional methods, significantly reducing the time required to adapt a one-minute clip by as much as 53%. We also discuss how VidAdapter can inform the design of future video authoring tools, especially those that aim to create simplified video presentation styles for ubiquitous usage. Taken together, our work offers the following contributions:

- Insights regarding the challenges of adapting existing blackboard videos for on-the-go learning on OHMDs using traditional video editors as informed by the formative study
- The design and implementation of VidAdapter as a holistic tool for adapting blackboard videos for ubiquitous viewing
- An evaluation of VidAdapter to understand how it can facilitate the video adaptation for OHMDs

2 BACKGROUND AND RELATED WORK

In this section, we first seek to familiarise the reader with the need to adapt digital content for mobile and wearable platforms based on recent research. We then look into prior research on tools that manipulate videos using semantic in-video information that our work builds upon.

2.1 Adapting Digital Content for Mobile and Wearable Platforms

Prior work has largely focused on adapting static digital content such as text documents or websites for phones through the use of responsive designs [24]. Responsive designs have been found to improve mobile readability and have been used to facilitate various tasks such as web-based learning [4], learning from e-books [3, 28, 34], and for information visualisation [13, 37] on mobile devices.

More recently, there has been increasing interest to adapt more dynamic digital content such as educational videos for ubiquitous use. This is because existing online educational videos have been designed keeping in mind a traditional user seated at a large-screen desktop who can afford dedicated attention during learning. As a result, these videos are unsuitable for learning on other small screen platforms such as phones or OHMDs in ubiquitous situations where users experience divided attention [9, 16, 26, 38]. For instance, Kim et al [16] investigated the usability of existing lecture videos for mobile phone-based learning and found their readability to be severely compromised, highlighting the need to adapt the size and layout of the video content for a better learning experience.

Recent work has also explored how wearable platforms like OHMDs can facilitate a better on-the-go video learning experience over phones if the videos are adapted into a presentation style called Layered Serial Visual Presentation (LSVP) [26, 27]. In addition to size and layout adaptations, LSVP requires modifying (1) the visual style of the video content to take advantage of the see-through properties of OHMD, and (2) the information presentation style to reduce the effects of divided attention that learners experience during ubiquitous learning. We provide a more elaborate discussion of this in Section 2.4.

While it is possible to create these adapted videos from scratch using existing video editing tools, we noticed in our formative studies that the process can be highly inefficient and tedious, even for experienced users. We believe that the efficiency of the adaptation process can be improved by reusing the video content rather than recreating it from scratch, for which we need the video content in a manipulable form. Thus, in the following section, we look at prior works that manipulate videos using in-video information.

2.2 Supporting Video Manipulation Using Semantic In-Video Information

As opposed to using video frames, considerable work has looked at how videos can be manipulated more efficiently using semantic information extracted from videos. One set of works has explored the use of semantic categorisations or summarisations extracted from videos for supporting tasks such as video navigation. For instance, ViZig [42] extracts and represents clickable object categories (e.g., diagrams or tables) from lecture videos to navigate to their respective position in the video. Similarly, Biswas et al. [5] enabled topic-wise video navigation using an automatically created list of topics in a video by analyzing its audio-visual content.

A more closely related line of work has explored the use of in-video objects to support direct manipulation wherein users can directly interact with objects instead of using a seeker bar. A classic example of this technique is the work by Dragicevic et al [10], which enables video navigation by dragging in-video objects along their motion trajectories. Of particular relevance is the NoteVideo tool [21] that extracts meaningful content from blackboard lectures as directly clickable in-video objects to navigate to the initial occurrence of that object.

The above tools, however, do not support modifying the objects themselves to create adapted versions of the video, which is the focus of VidAdapter. In this regard, the most closely related work to ours is the recent one by Kim et al [15]. Their system automatically converts existing slide-based lecture videos into a mobile-phone-friendly design and also enables the lecture elements to be resized and repositioned according to the user's preferences. However, the kind of adaptations needed to support ubiquitous platforms like OHMDs is more sophisticated, requiring adaptations along the spatial, temporal, and visual appearance dimensions. VidAdapter aims to support these multiple adaptation workflows in an efficient and streamlined manner. Note that by supporting adaptation for OHMDs, we can naturally support adaptations for mobile phones or tablets as their requirements can largely be considered as a subset of OHMDs' adaptation requirements.

2.3 Extracting Semantic Information from Lecture Videos

Semantic-level manipulation of in-video elements requires detecting and extracting content from lecture videos. Prior works have explored this task in two ways. The first set of works assumes access to lecture slides (and their meta-data) or transcripts to extract lecture content from the video [7, 17, 23, 25, 32, 33]. However, these methods can be restrictive as the source slides of the existing online lecture videos are often not available.

The second set of works focuses on analysing pixels across video frames and grouping them into semantic elements [2, 14, 21, 29, 41, 43–46]. A classic example of this type of approach can be seen in Monserrat et al [21], where the pixel difference across frames is monitored to identify lecture elements. More recently, there have been efforts to translate the success of deep learning to this task. For instance, Yadav et al [42] and Kim et al [15] trained Convolutional Neural Networks on labeled slide-based lecture data to improve detection accuracy. Despite the improved performance offered by these deep learning models, their applicability is limited to the domain of the data they were trained on, i.e., a model trained on slide-based video lectures can be difficult to generalise to blackboard lecture videos without large amounts of domain-relevant data. Given the limited availability of such large-scale datasets for blackboard lecture videos, VidAdapter's element extraction pipeline relies on a classical approach similar to [21] to extract semantic information.

2.4 Background

Here, we discuss the Layered Serial Visual Presentation (LSVP) style [26, 27] in more detail and how it informs the adaptation of lecture videos for on-the-go learning on OHMDs. LSVP denotes a set of desirable properties that educational videos should possess to enable a better balance of visual attention between the learning and path navigation tasks. These properties were identified by prior work through controlled studies in various realistic navigation tasks, which showed that videos presented on OHMDs using LSVP style could improve users' recall

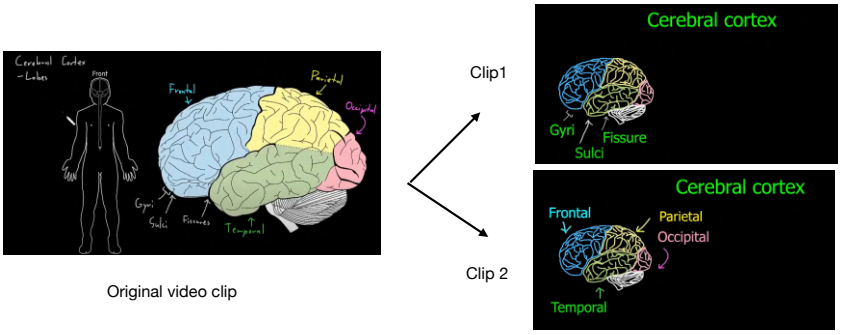


Fig. 2. Adapting a traditional blackboard lecture video from Khan Academy [1] (left) for on-the-go use on OHMDs by dividing it into shorter clips (right), each containing limited information and necessary adaptations described in section 2.4

scores with minimal impact on their walking abilities, as compared to using traditional lecture styles on phones or OHMDs.

One set of properties focuses on leveraging the see-through properties of OHMDs to facilitate navigation. These include (1) using a black video background that becomes transparent on an OHMD due to the additive nature of its display, (2) using typeface fonts of appropriate size instead of handwritten ones for better readability, (3) converting filled-images into an outline form to reduce environment occlusion, (4) reorganise the overall layout to reduce visual clutter, and (5) improving the color contrast of content for better clarity relative the background environment.

The other set of properties concerns presentation techniques that can reduce their cognitive load and improve their learning. In particular, information should (1) appear sequentially as small meaningful chunks to better guide the user's visual attention to relevant information, and (2) persist after it appears till the end of the video to reduce the temporal load on their working memory. However, persisting all information throughout the video can lead to visual clutter. To alleviate this, the video can be (3) segmented into shorter clips, each persisting only visual information relevant to its audio explanation. Future clips can include and persist information from previous clips on a need basis.

In addition to the above properties, traditional multimedia learning principles [20] such as aligning a content's appearance with the relevant part of the audio explanation (temporal contiguity) [22] should still be ensured for an effective learning experience. Together, these properties help convert a traditional lecture video into a format that is more suitable for on-the-go use on OHMDs as shown in Figure 2.

3 FORMATIVE STUDY

To guide our tool design, we conducted an observational study where we asked users with prior video editing experience to convert an existing lecture video into an OHMD-friendly style. Users converted the video using their preferred video editor, and we conducted formative interviews to understand the challenges they faced during the conversion process.

3.1 Participants & Apparatus

We recruited 4 users from the university community (3 female and 1 male, 26.25 ± 2.36 years) who self-identified as being experienced in the use of professional video editors such as Adobe Premiere Pro (3 users) and Final cut

pro (1 user). One of the users had five years of experience with Premiere Pro, while the remaining users had 2-3 years of experience creating tutorials or demos for various courses and projects.

Stimuli. We selected a blackboard-style lecture video from Khan Academy (3 min) as the material for adaptation. We focused on blackboard videos for two reasons. First, given their dynamic drawing style, they can be more challenging to adapt than static slide-based lecture videos. Second, it is a widely used lecture video style that is highly popular among e-learners [31].

Participants had to adapt the original video into smaller clips following the guidelines described in section 2.4. Since prior work does not provide concrete estimates regarding how short clips should be, we conducted pilots to find that a 1-1.5 minute long clip could provide a reasonable balance between on-screen visual clutter and conveying meaningful information to the user.

Hence, we asked users to adapt the original clip into two shorter clips of 1.5-minute duration consisting of an image element, 4-6 text elements, and any connecting lines/arrows as shown in Figure 2. Since it can be difficult to familiarise and apply these guidelines within the limited duration of the study, we created a sample version of the adapted clip and provided it as a reference during the task. For a successful adaptation, users had to recreate these sample clips from scratch, matching their versions as much as possible in terms of visual appearance, spatial organisation, and time of occurrence of the adapted content.

3.2 Procedure

After explaining the task to the participants, they were given time to skim through the original video and the sample adapted clips. Creating the first adapted clip was considered a training session, during which participants were free to explore and identify their preferred approach for adapting a video. They could use other tools (e.g., image editing tools like Adobe Photoshop) or platforms (e.g., iPads) in their conversion process as required.

After a short break, participants created the second clip using the working approach they identified earlier. We measured the time taken to complete the adaptation of the second clip. Finally, we conducted semi-structured interviews to understand the challenges associated with the adaptation task.

3.3 Findings

During the exploration stage, users considered several viable workflows before settling on a suitable one. Irrespective of the workflow, adapting a video's style consisted of several tedious and time-consuming steps, requiring on average (26.25 ± 3.5 minutes) for a 1.5-minute clip. In general, the steps performed by users to adapt a video can be divided into two broad categories: 1) content modification, which includes extracting content from the original video and modifying their visual appearance as necessary 2) content reorganisation, in which users adjusted the spatial position of the adapted content to reduce the visual clutter and its temporal properties to ensure a sequential and persistent display. Below we discuss three key issues that users faced while performing the above adaptation steps.

3.3.1 Content Modification is Cumbersome and Requires Multiple Tools/Platforms. Adapting a video for OHMDs requires fine-grained modification of the visual appearance of the original content. For instance, a crucial step in the adaptation process is converting solid-filled images into simple outline-based diagrams with transparent backgrounds [26]. However, existing video editors do not possess straightforward mechanisms to achieve such modifications. As a result, users had to use specialised image editing tools such as Adobe Photoshop where a complex array of color-based image segmentation and background removal techniques were used to create the required visual style (P2). Alternatively, most users (P1, P3, P4) preferred a less cumbersome yet time-consuming approach of tracing an outline diagram on a laptop or an iPad using a digital pen. The reliance on multiple external tools/platforms also creates the additional overhead of transferring content between tools or platforms.

Table 1. The adaptation workflows that our VidAdapter interface should support efficiently, as informed by guidelines for viewing on OHMDs [26, 27] and phones [16].

Video component	Issue	Platform	Adaptation to support	Code
Background	Video background occludes environment	OHMD	Black background (for transparency)	B1
Layout & Duration	High information density	OHMD, Phone	Remove unnecessary content	LD1
		OHMD, Phone	Reorganise content layout	LD2
		OHMD	Segment video into shorter clips	LD3
Text	Inappropriate font style	OHMD, Phone	Convert handwritten font style to typeface	T1
	Small text size	OHMD, Phone	Resize text	T2
	Low color contrast	OHMD, Phone	Color and background adjustment	T3
Image	Complex images	OHMD	Convert solid-filled image into outline diagram	I1
	Small image size	OHMD, Phone	Resize image	I2
Presentation style	Large chunks of information displayed at once	OHMD	Split into smaller chunks that appear sequentially	S1
	Transient information	OHMD	Persist the content in necessary clips	S2
	Information appearance not aligned with audio	OHMD, Phone	Align the appearance of content with audio (temporal contiguity)	S3

3.3.2 Realigning Adapted Content at the Right Time Points. Another challenge faced by users was during the content reorganisation process. In particular, setting each adapted content to appear sequentially at the right point in time matching the lecture audio was a tedious process. This is because performing this step requires users to identify the time points based on visual or auditory cues in the original video and instantiate the adapted content at that time point. Users located the time points by repeatedly browsing the original video with a series of coarse clicks and fine scrubs. Although some users eased this alignment process by labeling the time points after locating them, P3's comment sums up users' outlook regarding this step: "I need to play and scan the video repeatedly to find the right location [to add content]... it would be easier if I just knew where to add them ".

3.3.3 Frequent Alternation Between Modification and Reorganisation Steps Can Be Inefficient. We observed two different styles in which participants performed the adaptation steps. Most participants (P1, P2, P4) favored completing the adaptation of each content sequentially, leading to frequent switching between the modification and reorganisation steps. Alternatively, P3 adopted a more parallel approach by modifying all the content before reorganizing it. Our task completion time estimates with the limited group of users suggest that the parallel approach could be faster than the sequential one. While this could be due to individual differences, these findings support the extant literature on the attention costs of task switching and its detrimental effect on task performance [35]. However, current tools are not designed towards such a parallel approach – a shortcoming that we aim to resolve in our design.

4 DESIGN GOALS

We establish four design goals to guide the creation of the VidAdapter system based on the formative interview findings.

D1: Provide support for adaptation workflows that cater to ubiquitous viewing requirements To support ubiquitous learning, VidAdapter should allow a range of functionalities that can help adapt blackboard lecture videos to a variety of learning platforms. The required adaptations that need to be realized are listed in Table 1. Note that while phones and OHMDs have overlapping requirements, the adaptation specifics are different.

D2: Enable adaptation as a reuse rather than recreate a process Our formative study revealed that users spent the bulk of their time during video adaptation recreating the structure or content in the original video. For

instance, users had to recover the time points when the adapted content should appear based on the original video, which was tedious and time-consuming. Instead, VidAdapter should inherently enable direct reuse of the original video content, reducing the overhead of recreating content from scratch.

D3: Support efficient content adaptation VidAdapter should facilitate content modification and reorganisation operations using interaction metaphors that are efficient and intuitive for the adaptation process, removing the need for complex multi-step operations or multiple external/tools platforms.

D4: Implicitly streamline the adaptation steps Our observations of users' workflow indicated that current video editing tools do not naturally lead the user toward a more efficient adaptation workflow. VidAdapter should promote a seamless workflow by implicitly guiding users to perform adaptation operations that blend efficiently.

5 VIDADAPTER FRAMEWORK

To meet the above goals, we propose VidAdapter, a full stack framework that automatically extracts visual elements from existing lecture videos (D2) in the backend and allows users to directly manipulate and adapt the extracted elements in the frontend (D1, D3, D4). Computationally intensive operations such as modification of the extracted elements are routed to the backend server.

5.1 Visual Element Extraction

The input to the VidAdapter system is a blackboard lecture video. To extract the visual elements, we begin with an approach similar to Monserrat et al [21], preprocessing the video frames to remove mouse cursors. We then extract elements from the frames as images by (1) tracking the temporal changes in pixels across frames, and (2) grouping pixels spatially within a frame.

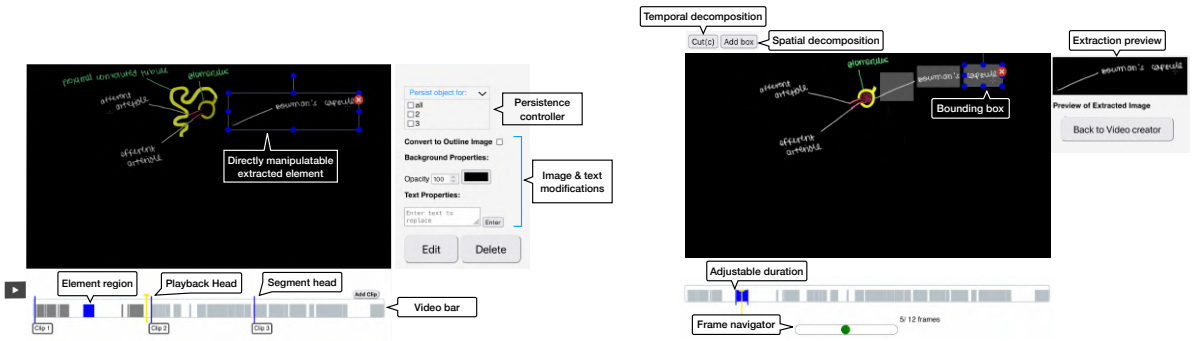
5.1.1 Tracking Temporal Changes in Pixels. In blackboard videos, content is progressively added to the board by the instructor as they explain a concept. To extract the newly added content as meaningful visual elements, we identify the start and end of the content by tracking the change in pixels across the frames. The difference image between the start and end frame provides an image of the element. Note that conceptually distinct content appearing in quick succession (e.g., when labeling a diagram) may be extracted as a single element in the difference frame due to temporal tracking.

5.1.2 Spatial Grouping of Pixels. Since the difference image may include multiple distinct elements, we further resolve it by grouping spatially contiguous pixels using connected component analysis [8]. This is based on the observation that lecture videos often display conceptually related content contiguously. Each group represents a visual element that is stored along with its metadata consisting of position coordinates, the width and height of its bounding box, and the start and end timestamps.

The accuracy of the above extraction pipeline can be affected by situations such as the complete erasure or scrolling of the board to create more space. We handle these situations by identifying the movement or disappearance of existing on-screen content using an object tracker.

5.2 VidAdapter Interface

The interface is designed to support multiple adaptation operations (D1) on the extracted visual elements in an efficient manner (D3). To perform these operations in a streamlined manner (D4), the VidAdapter is designed as a hierarchical interface consisting of two modes as seen in Fig 1, (1) Create mode and (2) Edit mode. Below we discuss the features and functions unique to each mode.



(a) Create mode (b) Edit Mode

Fig. 3. The two modes of the VidAdapter interface. The (a) Create mode, which allows direct manipulation of elements and their regions in the video bar for reorganisation and visual appearance modification, and (b) Edit Mode, which allows fine-grained spatial and temporal restructuring of the content extracted by the automatic pipeline.

5.3 Create Mode

This is the default mode of the interface where users can modify and reorganise the elements extracted from the video on a black video background (B1) as depicted in Figure 3(a).

5.3.1 Direct Manipulation on the Video Canvas. To facilitate a more intuitive adaptation (D3), the extracted elements are made directly manipulable as clickable images on the video canvas. Elements can be resized or repositioned on the canvas using the element's bounding box (LD2, T2, I2). Spurious elements extracted by the pipeline or unnecessary content can be deleted to reduce clutter (LD1), and consecutive elements can be grouped into a single element to be displayed together.

5.3.2 Modifying Visual Appearance of Elements. Handwritten text can be replaced with its typeface equivalent (using the text input bar) and its font color, background color, and opacity can be adjusted to make it more salient in OHMDs (T1, T3). Filled-image elements can be converted into their outline versions by applying the automatic outline filter (I1). The outline filter achieves this by creating an edge mask of the original image using a pre-trained deep edge detector model [40]. The pixels along the edge mask in the original image are extracted to serve as the final outline version of the filled image.

5.3.3 Region Highlighting in the Video Bar. The time in the video at which an element was extracted is highlighted in gray in the video bar as shown in Figure 3. The element region turns dark gray when the playback head passes it, indicating that the element is visible on the canvas. When an element is selected on the video canvas its region is highlighted in blue. Once selected, the element region can be dragged and dropped to a new position in the video bar, changing the time at which it will be displayed in the adapted video (S3). For more accurate repositioning, the element region can be automatically set to the playback head's location by dragging and dropping the region on it.

5.3.4 Managing Clutter Using Clip Segmentation. While directly deleting content can help reduce clutter, a more fine-grained approach to manage clutter is to split the video into shorter clips (LD3) and display only the required content in each clip. The segment heads help achieve this task by dividing the original video into shorter clips, with only the elements within the bounds of a clip being visible in that clip by default. By default, three segment

heads are randomly initialised in the video bar, and their positions can be manually adjusted to vary the elements that will be included in each clip.

5.3.5 Controlling the Persistence of Elements. Another adaptation workflow that needs to be realised is the ability to control the persistence of an element in clips (S2). An element is said to be persistent in a clip if it is visible for the entirety of that clip, after its initial appearance. We achieve this by using the segment heads in combination with the *persistence controller*, a dropdown box that allows users to specify which clips the element should persist in. We believe this can be an easy way to specify persistence because the segment heads are designed to limit an element's persistence to the clip where it is present for clutter management. Thus, the persistence controller acts as a simple way to indicate whether the element should be allowed to bypass the segment head and persist in other clips.

5.4 Edit Mode

To further modify an extracted element, users can enter the edit mode by clicking on the edit button.

Although similar in appearance, the edit mode differs from the create mode in two major ways. First, the video canvas now displays the original video along with the bounding box and element region created by the pipeline. Second, both the bounding box and the duration of the element region are editable, allowing users to redefine what will be extracted and transferred to the create mode. The extraction preview canvas provides real-time feedback of the extracted content. In addition, a frame navigator is provided to support frame-level navigation within the element regions.

Specifically, the edit mode provides functionalities to redefine what constitutes a meaningful element. For instance, consider the case in Figure 4 (first row), where the pipeline extracts the term "Bowman's capsule" and its connecting line as a viable element. Yet, this may not match the user's mental model. In such situations, users can perform the following operations.

5.4.1 Spatial Decomposition. Users can further decompose an extracted element into smaller chunks that display simultaneously. This can be done by adding multiple bounding boxes (using the add box button) and aligning each box to cover the required portion of the content as seen in Figure 4 (middle row). Reverting to the create mode allows users to reorganise each extracted chunk spatially independently of each other, allowing for finer layout management than existing tools (LD2).

5.4.2 Temporal Decomposition. To decompose an extracted element into smaller chunks that will display sequentially (S1), users can cut an element region temporally into two distinct regions and adjust the bounding box of each region. This situation is visualised in Figure 4 (last row), where the user chunks the connecting line and the term "Bowman's" as a unit followed by the term "capsule". Correspondingly, in the create mode the original element region appears split with the second region only associated with the term "capsule".

5.5 Implementation

VidAdapter interface is implemented as a javascript web application. The front-end direct manipulation interactions are developed with Fabric.js. The backend processing is handled by Python 3.8 running on a Node.js 16.17 server. We use the OpenCV 4.5.5 Python module for image processing operations and the deep edge detector for the outline image filter based on PyTorch [39]. We developed the system on a MacOS machine.

6 TECHNICAL EVALUATION

Before we evaluate our tool with users, we test the ability of our pipeline to extract meaningful content from the video. The evaluation is conducted by comparing the temporal segment boundaries of an extracted content (the start and end time points) against ground truth segmentations provided by humans.

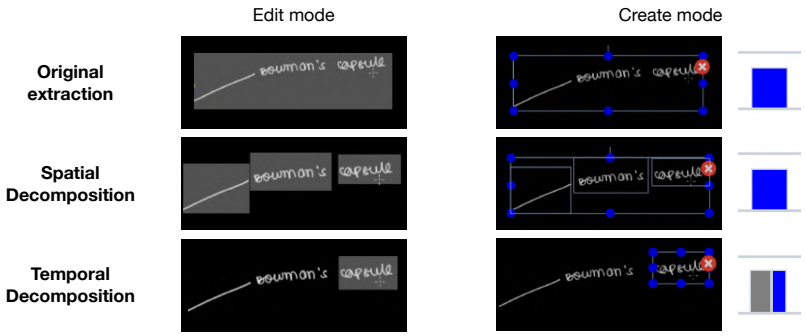


Fig. 4. A closer look at how the Edit mode enables spatial and temporal reconstruction of extracted elements. The first row indicates the element as extracted by the pipeline and how it is visualised in either mode. The second row shows how the original element can be decomposed into smaller chunks within the same element region using multiple boxes in the Edit mode. These sub-elements can then be spatially reorganised in the Create mode. The third row demonstrates the division of an element in Edit mode into two distinct elements that will appear sequentially. The temporal division is indicated by the split of the element region in the create mode.

6.1 Target Videos

To understand the generalisability of our approach we tested on five digital blackboard videos from varying subject domains sampled from two popular educational YouTube channels, Khan Academy and The Organic Chemistry Tutor. We chose videos that cover a wide range of challenging in-lecture instructor activities commonly seen in blackboard-style videos [21], including scrolling, partial/complete erasures, zoom-in explanations, and fast-forwarded dynamic drawing.

6.2 Video Labeling

We recruited a graduate volunteer from the local university who self-identified as a regular e-learner, having used online videos for their learning and revision for 6-8 years. The labeling task involved grouping video content into meaningful elements (by viewing it) based on their judgment and noting the time at which each element begins and ends.

6.3 Method

Following previous work [12, 36], we measure the accuracy of our segmentation by using boundary similarity [11]. Boundary similarity describes the similarity of two segmentations using a scoring that measures how far apart predicted segments are from their ground truth. It ranges from 0 (segmentation is different) to 1 (segmentation is identical). We discretise the segmentation in 1s intervals and use a boundary edit distance of 8 seconds (the average length of segments) similar to prior work. We compare our pipeline's performance with a baseline segmentation approach that uniformly segments a video based on the average segment length labeled by humans. The boundary similarity scores are computed using the segeval python package [11].

6.4 Results

As shown in Table 2, our content extraction algorithm has a better boundary similarity score (0.6 ± 0.1) than the baseline approach (0.46 ± 0.03), indicating that the pipeline's extraction is reasonably aligned with users expectations. We recognise that our algorithm's performance is not yet perfect, but this was expected given that

Table 2. The results of technical evaluation reported for blackboard lecture videos of varying subject domains and duration.

Video (subject domain)	Duration (mm:ss)	Boundary similarity	
		Ours	Uniform
Cell structure (biology)	03:57	0.61	0.52
Blackbody radiation problem solving (physics and maths)	08:36	0.48	0.42
Alkene naming (chemistry)	08:26	0.76	0.47
Functions of money (economics)	06:09	0.55	0.46
Gravitational force (physics)	10:00	0.61	0.45

the idea of grouping content into meaningful units is a highly subjective one and there is often more than one "right" way of segmenting the video [12]. A deeper look at the segmentations created by our extraction algorithm revealed that the reduction in boundary similarity scores was largely due to the end time points of an element being overestimated by the algorithm. As a result, the extracted content was still meaningful, being either in line with the user's expectation or grouping more information as an element. In the following section, we will explore how the end-user's experience is impacted given the current performance of our pipeline.

7 USER EVALUATION

We conducted a user evaluation with experienced video editing users to gain insights about VidAdapter and understand how it compares with traditional methods for the task of adapting blackboard lecture videos for ubiquitous viewing. The task was similar to the formative study, with users having to adapt the lecture video into an OHMD-friendly format. We also observed how VidAdapter was used by users with little to no experience in video editing and collected feedback on its design aspects.

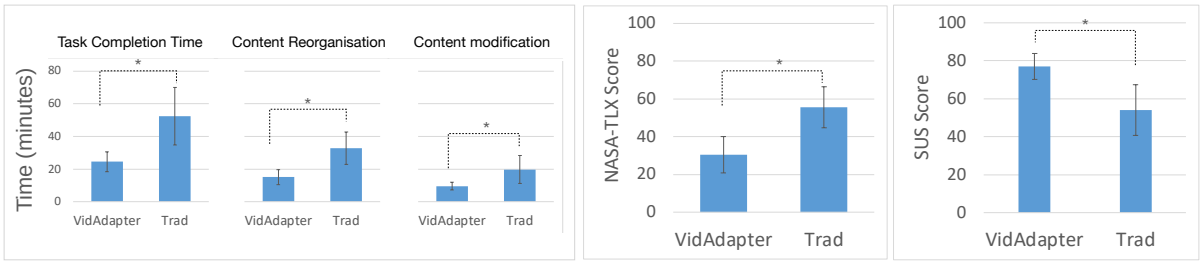
7.1 Experiment Design

We conducted a within-subject study comparing the two adaptation approaches (VidAdapter, Traditional). The order of adaptation approach was counterbalanced to account for order effects.

Stimuli. For the adaptation task we consider three blackboard-style Khan Academy videos having an average duration of (3.15 ± 0.08) minutes. Participants were assigned to one of the three videos in counterbalanced order. Each video had to be adapted into three shorter clips of ($M = 1.04,\ SD = 0.28$) minute duration and similar number of elements as in the formative study. Once again, sample versions of the adapted clips were provided as references.

7.2 Dependent Measures

7.2.1 Adaptation Efficiency. We measure the task completion time T_{tot} taken to adapt the original video into its respective clips. Based on the formative study, we further analyse T_{tot} in terms of the time needed for content modification T_{mod} and content reorganisation T_{reorg} operations to delineate the source of the efficiency. We measure T_{mod} as the time taken to edit or decompose any automatically extracted elements (if needed) before modifying their visual appearance. T_{reorg} includes the time needed to rearrange the spatial layout of elements and adjust their temporal position/duration in the clips.



(a) Adaptation Efficiency (b) Cognitive Load (c) User Experience

Fig. 5. Comparison of VidAdapter with traditional adaptation approaches in terms of (a) the adaptation efficiency (b) Cognitive load, and (c) Users' subjective experience

7.2.2 Cognitive Load. We use an unweighted NASA-TLX form to record users' subjective task load for each adaptation approach.

7.2.3 Subjective Preference. We used the System Usability Scale (SUS) [6] to record users' impressions of the usability of each approach. We compute the overall SUS score ranging from 0-100, where higher SUS scores indicate improved usability. In addition, we also asked users to rate the efficiency and ease of use of either approach using a 5-point Likert scale.

7.3 Participants & Apparatus

We recruited 12 users (6 female; 5 male; 1 prefer not to say; 23.75 ± 3.25 years) experienced with professional video editors (7 in Adobe Premiere Pro, 5 in Final Cut Pro). Ten of the 12 users self-identified as advanced or expert users with 3.5 ± 1.57 years of experience, while the remaining users had intermediate proficiency with 2-3 months of experience. We remunerated them at the standard rate of 7.4$ per hour.

7.4 Procedure

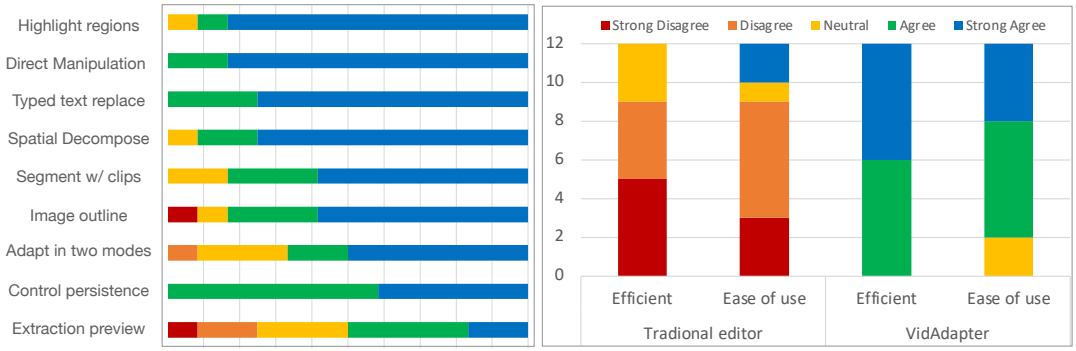
The study began by collecting participant's informed consent followed by a description of the task. In each approach, participants first practiced the adaptation task on a training video for 30 minutes. They were then asked to create the adapted clips for their assigned video and we measured the task completion time. Users were reminded of features to use where appropriate during the study.

After each adaptation approach, participants completed a NASA-TLX survey and rated its "efficiency" and "ease of use" using a 5-point Likert scale. For the VidAdapter approach, participants also indicated the usefulness of the various features in the tool. We followed this with a semi-structured interview to gain more feedback about the tool's usability.

7.5 Results

All measures (except T_{mod}) were normally distributed as indicated by a Shapiro-Wilk test (all $p > 0.1$). Therefore, a paired t-test was used for statistical analysis. T_{mod} showed a significant deviation from normality ($p = 0.003$) and hence was analysed with a Wilcoxon signed-rank test.

7.5.1 Adaptation Efficiency. The task completion time T_{Tot} was significantly lower using VidAdapter ($M = 24.58$, $SD = 6.08$ minutes) as compared to the traditional approach ($M = 52.32$, $SD = 17.66$ minutes) ($t[11] = 6.32$, $p < .001$, $d = 1.82$). A deeper analysis revealed that the lower T_{Tot} was largely due to improved efficiency



(a) Usefulness (b) Efficiency and Ease of Use

Fig. 6. (a) Usefulness of each feature of VidAdapter. (b) Comparison of VidAdapter and traditional video editors by experienced users in terms of their efficiency and ease of use for adapting a blackboard video.

of the content reorganisation operations. This is indicated by the significantly lower T_{reorg} for the VidAdapter ($M = 15.06,\ SD = 4.67$ minutes) as opposed to that for traditional approach ($M = 32.73,\ SD = 9.93$ minutes) ($t[11] = 4.19,\ p = 0.002,\ d = 1.21$). T_{mod} was also lower using VidAdapter ($M = 19.73,\ SD = 8.50$ minutes) than the traditional approach ($M = 24.58,\ SD = 6.07$ minutes) ($Z = 3.06,\ p < .01,\ r = 1.0$).

7.5.2 Cognitive Load. Users' cognitive load was significantly lower when adapting content with VidAdapter ($M = 30.36,\ SD = 9.54$) than their traditional approach ($M = 55.5,\ SD = 10.73$) ($t[11] = 11.02,\ p < .001,\ d = 3.18$) in terms of the unweighted NASA-TLX scores.

7.5.3 User Experience and Subjective Preference. Experienced users unanimously preferred VidAdpater over their traditional approach for the task of adapting videos, with VidAdapter ($M = 76.88,\ SD = 6.92$) being scored significantly higher than traditional approach ($M = 53.96,\ SD = 13.29$) in terms of overall SUS score ($t[11] = 4.93,\ p < .001,\ d = 1.42$) . They also rated VidAdapter to be more efficient ($M = 4.5,\ SD = 0.52$) and easy to use ($M = 4.16,\ SD = 0.71$) than their traditional approach, which had respective scores of ($M = 1.83,\ SD = 0.83$) and ($M = 2.33,\ SD = 1.37$).

7.6 Discussion

Overall, VidAdapter allowed experienced users to complete the adaptation process more efficiently than possible with their traditional approach, yielding a 53% reduction in task completion time. We now describe more about users' perception of VidAdapter by grouping them into three main themes.

7.6.1 Direct Element-Level Manipulations are Better Than Frame/Pixel-Level Manipulations. Adapting videos using traditional editors often involves searching through the video frames and modifying them at a pixel level using a host of image-editing tools. VidAdapter side steps these frame/pixel level manipulations by automatically extracting video content as semantic elements that are directly manipulable, thereby adopting an element-level manipulation approach.

All participants found the ability to directly manipulate elements on the video canvas to be useful for content reorganisation, with average ratings of ($M = 4.83,\ SD = 0.40$) for usefulness as seen in Figure 6a. In particular, direct manipulation allowed users to quickly reorganise the spatial layout of the extracted content while simultaneously viewing the adapted video: "It feels natural to directly edit the content in the output [adapted]

video into the form I want (P3)". Direct manipulation also helped improve the intuitiveness and efficiency of content modification operations in VidAdapter, especially those that can benefit from having the video elements as context during the modification process. For instance, P2 highlighted their preference for the text modification operation in VidAdapter " I can refer to the content [element] on the display as I modify it".

Similarly, by using highlighted regions to represent the extracted elements temporally and automatically aligning them with the audio, VidAdapter minimised the need for temporal edits and enabled faster temporal reorganisation. Consequently, users rated this feature to be useful ($M = 4.75,\ SD = 0.62$). For instance, P5 contrasted their smooth experience using VidAdapter "It's much easier to navigate and identify where changes should be made with the highlighted regions" with the repetitive "click and locate" browsing in traditional video editors, a feeling also shared by other experienced users.

7.6.2 Support Frequent Multi-step Adaptations with Simplified Operations. Participants appreciated the ease with which VidAdapter facilitated frequently performed multi-step content modification operations. In particular, the automatic image outline feature was found to be advantageous with ratings of ($M = 4.25,\ SD = 1.21$) for "usefulness", replacing the otherwise tedious process with a single click. While a few experienced users (2) mentioned that the quality of automatic outlining was "not like what I draw", the remaining users (10) found it to be sufficiently similar for the task.

Experienced users also positively rated the segment-based clipping ($M = 4.41,\ SD = 0.79$) and persistence control ($M = 4.41,\ SD = 0.51$) approach enabled by VidAdapter to control information flow. In traditional video editors, content (represented as clips in a track) is set to persist by occupying the required duration in its track. Persisting multiple contents thus requires multiple tracks, leading to a vertically stacked clip representation that can quickly become "overwhelming to view" (P9) and "confusing to manage" (P4). Instead, VidAdapter enabled a more succinct horizontal representation by controlling content persistence using the segment heads as "gates" that block the content within that segment and the persistence controller that passes content to selected segments, unblocking these gates.

7.6.3 Hierarchical Division of Adaptation Functions. VidAdapter was designed as a hierarchical two-stage interface, providing relevant content reorganisation and modification functions in the create stage and allowing fine-grained editing of the extracted content in the editing stage. We expected that this division of functions would make the adaptation process more efficient and intuitive for users. While our study indicates improved efficiency, users reported mixed feelings about the intuitiveness the two-stage design, indicating future possibilities for improving this aspect of the interface.

8 OVERALL DISCUSSION

Adapting lecture video styles for ubiquitous learning can be a challenging and time-consuming task, even for existing video editing users as shown in our formative studies. Our evaluation suggests that VidAdapter was largely successful in circumventing these problems through its many design components. Chief among them is the benefits offered by direct element-level manipulation for the adaptation of the elements. In particular, direct manipulation was key to supporting operations that enabled the decomposition or recomposition of elements along spatial and temporal dimensions. Furthermore, it facilitated efficient modification of an element's visual appearance through idiomatic operations for images and by providing contextual information useful for converting text elements into typeface form. While VidAdapter is primarily intended to support ubiquitous learning, we believe that it can also be useful for making existing videos more accessible, especially for students with learning disabilities. For instance, students with Dyslexia can find handwritten text in videos difficult and prefer videos to be in a certain format over others [19]. With VidAdapter, lecturers or content creators can easily make their videos accessible to these communities.

Another aspect of VidAdapter's design that warrants discussion is the tradeoff between the complexity of the video that needs to be created and the granularity of the manipulations offered by the creation tool. Professional video editors, in general, offer a range of intricate functions that allow for fine-grained editing and the creation of complex videos. However, in situations such as ubiquitous learning where simpler video designs suffice, evidence from our formative studies suggests that these fine-grained capabilities may not be as useful or efficient. Instead, the creation of simpler videos could be supported through more coarse-grained functions that achieve the same objective while being more intuitive and efficient. An example of such a function in VidAdapter is the way a content's presence in the video is defined at a segment level (rather than a frame level) using the segment heads and a persistence controller. In addition, realizing persistence in this form also led to a more manageable visualisation of the elements in the video bar. Thus, such coarse-grained functions could also simplify the design of other components in the editor, reducing the entry barrier for novice users.

Lastly, the hierarchical division of VidAdapter's functions into two modes provides insights for developing more efficient video creation workflows. While traditional editors allow for any edits or effects to be applied to video content throughout the creation process, as indicated by our formative studies, this flexibility may not be useful when predefined video designs need to be created. This is because when tasked with creating a new video style, users tend to alternate between different kinds of editing or modification operations, often ending up with a sub-optimal workflow. In such situations, users can be scaffolded towards a more efficient workflow by limiting their access to functions that are relevant to the current stage of video creation.

8.1 Limitations & Future Work

8.1.1 Capability to Adapt for Multiple Platforms and Lecture Video. Although users experienced VidAdapter's capabilities in terms of adapting a video for an OHMD, there was general agreement that such a tool could be useful for adapting videos to other platforms as well. For instance, the mobile-friendly adaptations proposed by Kim et al [16] are a subset of those required for on-the-go OHMD viewing and thus can be easily achieved using VidAdapter. Nevertheless, we recognise that we did not formally evaluate adaptation for phones or other platforms and we hope that future work can shed more light on this. On the other hand, VidAdapter is currently limited to the adaptation of blackboard lecture videos by the content extraction pipeline. Thus, future work can look into leveraging deep learning to extract elements more accurately across a broad range of lecture video styles.

8.1.2 Enhancing the Functionalities of VidAdapter. VidAdapter provided several functionalities that made the video adaption process more efficient for users. However, some aspects of the adaptation process such as the conversion of the handwritten text objects into typed text, which is currently performed manually by the user, can be further simplified. Future work can look into automatically detecting and recognizing the text in the videos using OCR models such as [18] to further reduce the burden on users.

8.1.3 Do We Need Yet Another Tool for Video Editing/Adaptation? Given the plethora of existing video editors available for use, a reasonable question that readers may have is whether we need yet another tool to edit or modify videos. However, the intention of this work is not to argue for a separate tool but rather the need for more efficient workflows for video adaptation and to identify potential features that can support it. While this suggests that VidAdapter could have been implemented as a plugin to an existing video editor, we believe that realizing these functionalities using a custom implementation has its unique merit, especially for research. In particular, building VidAdapter's functionalities as a plugin could have stunted innovation due to both the legacy bias and implementation constraints imposed by the existing video editor. Moving forward, however, the features and designs identified in this work could either inspire new video editing tools or be integrated into existing video editors to enable a more efficient video adaptation.

9 CONCLUSION

We presented VidAdapter, a novel tool that allows the adaptation of blackboard lecture videos into alternative video styles that can support ubiquitous learning. This tool was developed based on insights from a formative study that revealed the challenges that users with video editing experience faced while adapting a lecture video for on-the-go learning on OHMDs. To enable a more efficient adaptation workflow, VidAdapter enables direct manipulation of video content by extracting them as meaningful elements and further supports a wide range of spatial, temporal, and visual appearance modification operations on these elements. Together, these functionalities allow VidAdapter to support adaptation workflows for multiple viewing platforms and/or usage contexts in an efficient and streamlined manner. We further presented the results of an evaluation of VidAdapter with experienced video editing users, which demonstrated a strong preference for VidAdapter over traditional video editing tools. Our findings contribute to the literature that seeks to support ubiquitous learning, and we discuss how the design of our tool can inform the design of future video editing interfaces that can better support ubiquitous computing.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-016) and the programme DesCartes supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. It is also supported in part by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 programme (MOE-T2EP20221-0010), and by a research grant No: 22-5913-A0001 from the Ministry of Education of Singapore. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation or the Ministry of Education, Singapore. This research is also partially supported by the Research Grants Council of the Hong Kong Special administrative region, China (T45-401/22-N). We also thank reviewers for their invaluable time and feedback.

REFERENCES

[1] Khan Academy. 2014. *Cerebral cortex - Khan Academy.* Retrieved July 25, 2023 from https://www.youtube.com/watch?v=mGxomKWfJXs&t=356s&ab_channel=khanacademymedicine

[2] Esha Baidya and Sanjay Goel. 2014. LectureKhoj: Automatic tagging and semantic segmentation of online lecture videos. In *2014 Seventh International Conference on Contemporary Computing (IC3)*. 37–43. https://doi.org/10.1109/IC3.2014.6897144

[3] Meltem Huri Baturay and Murat Birtane. 2013. Responsive Web Design: A New Type of Design for Web-based Instructional Content. *Procedia - Social and Behavioral Sciences* 106 (2013), 2275–2279. https://doi.org/10.1016/j.sbspro.2013.12.259 4th International Conference on New Horizons in Education.

[4] Vivek Bhuttoo, Kamlesh Soman, and Roopesh Kevin Sungkur. 2017. Responsive design and content adaptation for e-learning on mobile devices. In *2017 1st International Conference on Next Generation Computing Applications (NextComp)*. 163–168. https://doi.org/10.1109/NEXTCOMP.2017.8016193

[5] Arijit Biswas, Ankit Gandhi, and Om Deshmukh. 2015. MMToC: A Multimodal Method for Table of Content Creation in Educational Videos *(MM '15)*. Association for Computing Machinery, New York, NY, USA, 621–630. https://doi.org/10.1145/2733373.2806253

[6] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).

[7] Xiaoyin Che, Haojin Yang, and Christoph Meinel. 2013. Lecture Video Segmentation by Automatically Analyzing the Synchronized Slides. In *Proceedings of the 21st ACM International Conference on Multimedia* (Barcelona, Spain) *(MM '13)*. Association for Computing Machinery, New York, NY, USA, 345–348. https://doi.org/10.1145/2502081.2508115

[8] Michael B. Dillencourt, Hanan Samet, and Markku Tamminen. 1992. A General Approach to Connected-Component Labeling for Arbitrary Image Representations. *J. ACM* 39, 2 (apr 1992), 253–280. https://doi.org/10.1145/128749.128750

[9] Peter E Doolittle and Gina J Mariano. 2008. Working memory capacity and mobile multimedia learning environments: Individual differences in learning while mobile. *Journal of Educational Multimedia and Hypermedia* 17, 4 (2008), 511–530. https://psycnet.apa.org/record/2008-16147-003

[10] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video Browsing by Direct Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/1357054.1357096

[11] Chris Fournier. 2013. Evaluating Text Segmentation using Boundary Edit Distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, 1702–1712. https://aclanthology.org/P13-1167

[12] C. Ailie Fraser, Joy O. Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. 2020. Temporal Segmentation of Creative Live Streams *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376437

[13] Jane Hoffswell, Wilmot Li, and Zhicheng Liu. 2020. Techniques for Flexible Responsive Visualization Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376777

[14] Hyeungshik Jung, Hijung Valentina Shin, and Juho Kim. 2018. DynamicSlide: Exploring the Design Space of Reference-Based Interaction Techniques for Slide-Based Lecture Videos. In *Proceedings of the 2018 Workshop on Multimedia for Accessible Human Computer Interface* (Seoul, Republic of Korea) *(MAHCI'18)*. Association for Computing Machinery, New York, NY, USA, 33–41. https://doi.org/10.1145/3264856.3264861

[15] Jeongyeon Kim, Yubin Choi, Minsuk Kahng, and Juho Kim. 2022. FitVid: Responsive and Flexible Video Content Adaptation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 501, 16 pages. https://doi.org/10.1145/3491102.3501948

[16] Jeongyeon Kim, Yubin Choi, Meng Xia, and Juho Kim. 2022. Mobile-Friendly Content Design for MOOCs: Challenges, Requirements, and Design Opportunities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 92, 16 pages. https://doi.org/10.1145/3491102.3502054

[17] Francis C. Li, Anoop Gupta, Elizabeth Sanocki, Li-wei He, and Yong Rui. 2000. Browsing Digital Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 169–176. https://doi.org/10.1145/332040.332425

[18] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2021. Trocr: Transformer-based optical character recognition with pre-trained models. (2021). https://doi.org/10.48550/arXiv.2109.10282

[19] Lois MacCullagh, Agnes Bosanquet, and Nicholas A. Badcock. 2017. University Students with Dyslexia: A Qualitative Exploratory Study of Learning Practices, Challenges and Strategies. *Dyslexia* 23, 1 (2017), 3–23. https://doi.org/10.1002/dys.1544 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/dys.1544

[20] Richard E. Mayer. 2002. Multimedia learning. Psychology of Learning and Motivation, Vol. 41. Academic Press, 85 – 139. https://doi.org/10.1016/S0079-7421(02)80005-6

[21] Toni-Jan Keith Palma Monserrat, Shengdong Zhao, Kevin McGee, and Anshul Vikram Pandey. 2013. NoteVideo: Facilitating Navigation of Blackboard-Style Lecture Videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1139–1148. https://doi.org/10.1145/2470654.2466147

[22] Roxana Moreno and Richard Mayer. 1999. Cognitive Principles of Multimedia Learning: The Role of Modality and Contiguity. *Journal of Educational Psychology* 91 (06 1999), 358–368. https://doi.org/10.1037/0022-0663.91.2.358

[23] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 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* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 573–582. https://doi.org/10.1145/2642918.2647400

[24] Wenhui Peng and Yaling Zhou. 2015. The Design and Research of Responsive Web Supporting Mobile Learning Devices. In *2015 International Symposium on Educational Technology (ISET)*. 163–167. https://doi.org/10.1109/ISET.2015.40

[25] Yi-Hao Peng, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. 2021. Say It All: Feedback for Improving Non-Visual Presentation Accessibility. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 276, 12 pages. https://doi.org/10.1145/3411764.3445572

[26] Ashwin Ram and Shengdong Zhao. 2021. LSVP: Towards Effective On-the-Go Video Learning Using Optical Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 1, Article 30 (March 2021), 27 pages. https://doi.org/10.1145/3448118

[27] Ashwin Ram and Shengdong Zhao. 2022. Does Dynamically Drawn Text Improve Learning? Investigating the Effect of Text Presentation Styles in Video Learning. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 89, 12 pages. https://doi.org/10.1145/3491102.3517499

[28] Luz Rello, Gaurang Kanvinde, and Ricardo Baeza-Yates. 2012. A Mobile Application for Displaying More Accessible eBooks for People with Dyslexia. *Procedia Computer Science* 14 (2012), 226–233. https://doi.org/10.1016/j.procs.2012.10.026 Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2012).

[29] Hijung Valentina Shin, Floraine Berthouzoz, Wilmot Li, and Frédo Durand. 2015. Visual Transcripts: Lecture Notes from Blackboard-Style Lecture Videos. *ACM Trans. Graph.* 34, 6, Article 240 (nov 2015), 10 pages. https://doi.org/10.1145/2816795.2818123

[30] Bernardo Tabuenca, Stefaan Ternier, and Marcus Specht. 2013. Supporting Lifelong Learners to Build Personal Learning Ecologies in Daily Physical Spaces. *Int. J. Mob. Learn. Organ.* 7, 3/4 (Oct. 2013), 177–196. https://doi.org/10.1504/IJMLO.2013.057160

[31] Clive Thompson. 2011. *How Khan Academy is changing the rules of education.* Retrieved November 14, 2022 from https://www.wired.com/2011/07/ff-khan/

[32] Miles Thorogood. 2016. SlideDeck.Js: A Platform for Generating Accessible and Interactive Web-Based Course Content. In *Proceedings of the 21st Western Canadian Conference on Computing Education* (Kamloops, BC, Canada) *(WCCCE '16)*. Association for Computing Machinery, New York, NY, USA, Article 13, 5 pages. https://doi.org/10.1145/2910925.2910941

[33] Shoko Tsujimura, Kazumasa Yamamoto, and Seiichi Nakagawa. 2017. Automatic Explanation Spot Estimation Method Targeted at Text and Figures in Lecture Slides.. In *INTERSPEECH*. 2764–2768.

[34] Nicholas Vanderschantz, Claire Timpany, and Annika Hinze. 2015. Design Exploration of EBook Interfaces for Personal Digital Libraries on Tablet Devices. In *Proceedings of the 15th New Zealand Conference on Human-Computer Interaction* (Hamilton, New Zealand) *(CHINZ 2015)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/2808047.2808054

[35] André Vandierendonck, Baptist Liefooghe, and Frederick Verbruggen. 2010. Task Switching: Interplay of Reconfiguration and Interference Control. *Psychological bulletin* 136 (07 2010), 601–26. https://doi.org/10.1037/a0019791

[36] Bryan Wang, Meng Yu Yang, and Tovi Grossman. 2021. Soloist: Generating Mixed-Initiative Tutorials from Existing Guitar Instructional Videos Through Audio Processing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 98, 14 pages. https://doi.org/10.1145/3411764.3445162

[37] Aoyu Wu, Wai Tong, Tim Dwyer, Bongshin Lee, Petra Isenberg, and Huamin Qu. 2021. MobileVisFixer: Tailoring Web Visualizations for Mobile Phones Leveraging an Explainable Reinforcement Learning Framework. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 464–474. https://doi.org/10.1109/TVCG.2020.3030423

[38] Xiang Xiao and Jingtao Wang. 2017. Undertanding and Detecting Divided Attention in Mobile MOOC Learning. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2411–2415. https://doi.org/10.1145/3025453.3025552

[39] Saining Xie. 2015. *Holistically-nested edge detection.* Retrieved March 19, 2023 from https://github.com/s9xie/hed

[40] Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[41] Chengpei Xu, Ruomei Wang, Shujin Lin, Xiaonan Luo, Baoquan Zhao, Lijie Shao, and Mengqiu Hu. 2019. Lecture2Note: Automatic Generation of Lecture Notes from Slide-Based Educational Videos. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. 898–903. https://doi.org/10.1109/ICME.2019.00159

[42] Kuldeep Yadav, Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Saurabh Srivastava, and Om Deshmukh. 2016. ViZig: Anchor Points Based Non-Linear Navigation and Summarization in Educational Videos. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. Association for Computing Machinery, New York, NY, USA, 407–418. https://doi.org/10.1145/2856767.2856788

[43] Haojin Yang and Christoph Meinel. 2014. Content Based Lecture Video Retrieval Using Speech and Video Text Information. *IEEE Transactions on Learning Technologies* 7, 2 (2014), 142–154. https://doi.org/10.1109/TLT.2014.2307305

[44] Haojin Yang, Maria Siebert, Patrick Luhne, Harald Sack, and Christoph Meinel. 2011. Lecture Video Indexing and Analysis Using Video OCR Technology. In *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*. 54–61. https://doi.org/10.1109/SITIS.2011.20

[45] Baoquan Zhao, Shujin Lin, Xiaonan Luo, Songhua Xu, and Ruomei Wang. 2017. A Novel System for Visual Navigation of Educational Videos Using Multimodal Cues. In *Proceedings of the 25th ACM International Conference on Multimedia* (Mountain View, California, USA) *(MM '17)*. Association for Computing Machinery, New York, NY, USA, 1680–1688. https://doi.org/10.1145/3123266.3123406

[46] Baoquan Zhao, Songhua Xu, Shujin Lin, Ruomei Wang, and Xiaonan Luo. 2019. A New Visual Interface for Searching and Navigating Slide-Based Lecture Videos. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. 928–933. https://doi.org/10.1109/ICME.2019.00164