AVscript: Accessible Video Editing with Audio-Visual Scripts

Mina Huh*
The University of Texas at Austin
Austin, TX, USA
minahuh@cs.utexas.edu

Saelyne Yang
KAIST
Daejeon, Republic of Korea
saelyne@kaist.ac.kr

Yi-Hao Peng
Carnegie Mellon University
Pittsburgh, PA, USA
yihaop@cs.cmu.edu

Xiang 'Anthony' Chen
UCLA
Los Angeles, CA, USA
xac@ucla.edu

Young-Ho Kim
NAVER AI Lab
Republic of Korea
yghokim@younghokim.net

Amy Pavel
The University of Texas at Austin
Austin, TX, USA
apavel@cs.utexas.edu

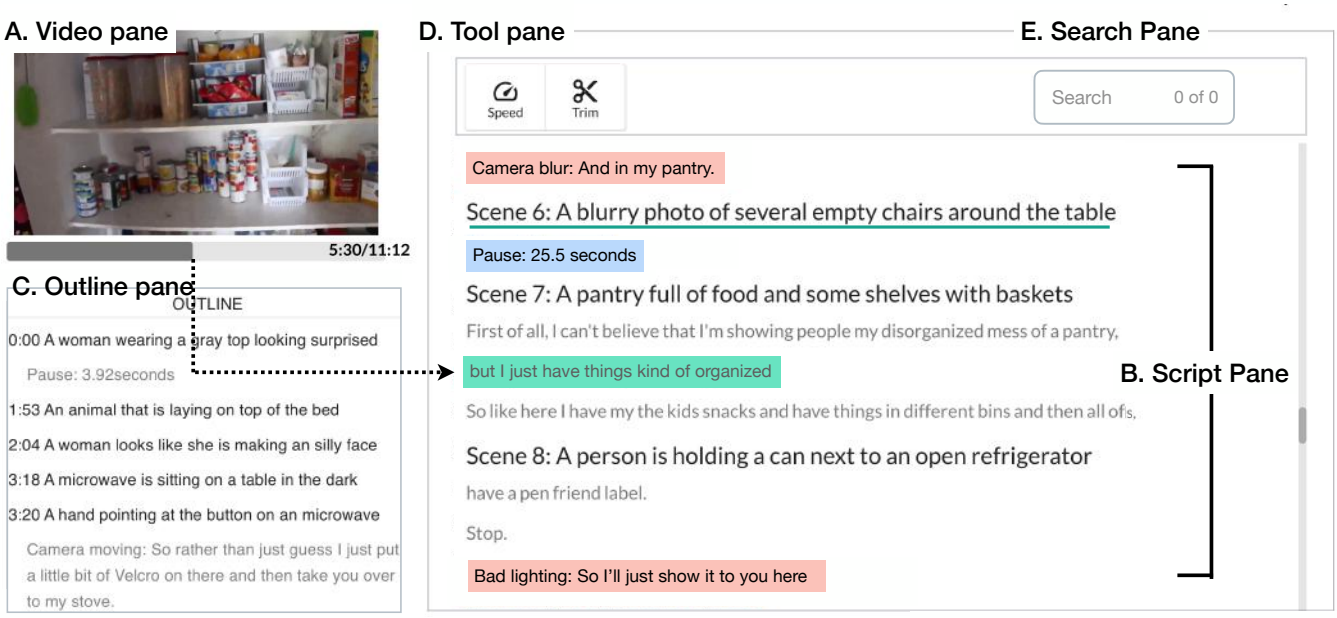

Figure 1: AVscript is an accessible text-based video editing tool that enables blind and low vision (BLV) creators to edit videos efficiently using a screen reader. The *video pane* (A) provides notifications for visual errors and supports inspection of visual objects. The *script pane* (B) provides a narration transcript segmented by scenes, scene descriptions, and highlighted visual errors (*e.g.,* blur, bad lighting). The *outline pane* (C) a navigable summary of the video scenes and errors. The *tool pane* (D) allows users to change the playback speed or trim of a selected video clip. The *search pane* (E) supports both visual search and narration search of the video.

ABSTRACT

Sighted and blind and low vision (BLV) creators alike use videos to communicate with broad audiences. Yet, video editing remains inaccessible to BLV creators. Our formative study revealed that current video editing tools make it difficult to access the visual content, assess the visual quality, and efficiently navigate the timeline. We present AVscript, an accessible text-based video editor. AVscript enables BLV creators to edit their video using a script that embeds the video's visual content, visual errors (*e.g.*, dark or blurred footage), and speech. BLV creators can use AVscript to efficiently navigate between scenes and visual errors or to locate objects in the frame or spoken words of interest. A comparison study (N=12) showed that AVscript significantly lowered BLV creators' mental demand while increasing confidence and independence in video editing. We further demonstrate the potential of AVscript through an exploratory study (N=3) where BLV creators edited their own footage.

*Mina Huh conducted part of this work as a research intern at NAVER AI Lab.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3581494

CCS CONCEPTS

• **Human-centered computing**; • **Accessibility systems and tools**;

KEYWORDS

video, authoring tools, accessibility

ACM Reference Format:
Mina Huh, Saelyne Yang, Yi-Hao Peng, Xiang 'Anthony' Chen, Young-Ho Kim, and Amy Pavel. 2023. AVscript: Accessible Video Editing with Audio-Visual Scripts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3581494

1 INTRODUCTION

People create videos to share their experiences and expertise with a broad audience. To create a compelling video, creators first capture raw video footage then edit the video to remove irrelevant or low-quality footage and add effects. For instance, creators speed up repetitive actions (*e.g.*, walking or cleaning), remove long pauses in their speech, and cut blurry footage due to camera shakes. With current video editing tools (*e.g.*, Premiere [13], Final Cut Pro [15], Descript [25]), creators first need to *visually inspect* the video footage and the corresponding video timeline [13, 15] or transcript [25], to identify edit points. Similar to accessibility barriers encountered by BLV creators authoring static visual media (*e.g.*, photos [17], presentations [62], documents [24]), requiring visual inspection makes video editing tools inaccessible to a growing number of blind and low vision (BLV) video creators [70] who author and share general-purpose and accessibility-focused videos including vlogs, reviews, and tutorials. While prior work explored how to make videos accessible to BLV audience members [49, 50, 58], and how to make video editing more efficient for sighted creators [18, 23, 34], existing work has not yet explored how to make video editing accessible to BLV creators.

To better understand BLV creators' current video production strategies and challenges, we interviewed 8 BLV video creators and analyzed 24 videos about screen reader-based video editing. While BLV creators devised creative techniques to film and edit their videos such as describing the visual content during video capture, and editing the video using audio editing tools, the creators reported that video editing remained challenging due to four core accessibility barriers of videos and video editing tools: lack of access to the visual content of the video (*e.g.*, settings, objects), lack of access to the visual quality of the video (*e.g.*, lighting, blurriness), lack of efficient ways to navigate to different parts of the video, and limited screen reader support (*e.g.*, deeply nested menus or icons listed as "button"). As a result, BLV creators reported that they either recruited sighted collaborators to review and edit their video, or uploaded their original video recording without editing the footage to their preferred level of polish.

To address accessibility barriers of current video editing tools, we present AVscript, a system that supports accessible video editing via *audio-visual scripts* (Figure 1). AVscript's audio-visual script (Figure 1B) features a transcript of the narration in the video (*e.g.*, "First of all,...") overlaid with information about the visual content in the video (*e.g.*, "Scene 7: A pantry full of food...") and visual errors in the video (*e.g.*, "Camera blur"). We align the audio-visual script to the video such that BLV creators can directly review, navigate, and edit the video via text. As creators play the video, AVscript surfaces visual information by alerting creators to scene changes and visual errors using audio notifications. AVscript also allows creators to inspect objects in the current frame using the "Inspect" feature. To facilitate efficient navigation, AVscript features an outline (Figure 1C) and search feature (Figure 1E). AVscript's outline (Figure 1C) of key scenes and errors lets creators skim to gain a high-level overview of the video or click to navigate to the corresponding point in the script and video. AVscript's search (Figure 1E) lets creators navigate the video by searching for visual objects (*e.g.*, "microwave") or transcript words of interest.

To assess AVscript, we conducted a within-subjects study with 12 BLV editors comparing AVscript to the their existing workflows and invited 3 BLV creators to edit their own footage using AVscript. In the within-subjects study with 12 BVI editors, creators reported lower mental demand and greater confidence in their final video when editing videos with AVscript compared to using their own video editing tools (*e.g.*, Reaper, a timeline-based editor, and FFmpeg, a command-line tool). All creators expressed that they wanted to use the tool in the future as it helped them efficiently review their video footage and identify visual errors. BLV creators editing their own footage with AVscript used AVscript's visual descriptions to efficiently recall what they filmed, and AVscript's error detection to remove visual errors. After using AVscript to edit their own footage, creators reported that AVscript would enable them to edit more videos without assistance, thus decreasing the time required to produce videos and empowering them to create new types of videos with more diverse content and styles.

Our work contributes:

- A formative study revealing current practices and unique challenges of video editing by BLV creators.
- Design and development of AVscript, a novel system that uses *audio-visual script* to improve accessibility in reviewing, navigating, and editing videos.
- User studies demonstrating how BLV creators leverage AVscript to edit given videos and their own videos.

2 RELATED WORK

Our work builds upon previous work in accessible authoring tools (Section 2.1), video accessibility (Section 2.2), text-based editing tools for audio and visual media (Section 2.3), and interaction techniques for video navigation (Section 2.4).

2.1 Accessible Authoring Tools

Prior research has explored how creators with visual impairments currently author photos [11, 19], drawings [41], documents [79], websites [46], presentations [62, 69], audio [66], and videos [70]. Such work identified that authoring tools for visual content remain inaccessible because they do not convey visual information about the content the creator is authoring (*e.g.*, framing of a photo [11], layout of a website [46]). Thus, prior work created tactile displays to make authoring websites [46], documents [16], and maps [71] accessible. While tactile displays allow creators with visual impairments to access visual content, creators must have access to an embosser or

laser cutter to print tactile sheets for the initial and revised visual designs. To provide access to visual designs and revisions without tactile displays, Peng et al. generated visual descriptions that let presenters with visual impairments obtain information about the content, layout, and style of their slides [62]. Prior work also explores how to support collaboration between creators who are sighted and creators with visual impairments (*i.e.* mixed-ability collaboration) by providing descriptions of revisions [62], and access to awareness information that describes who was authoring what during collaborative editing [24, 43]. While these tools make authoring static visuals (*e.g.,* text documents, visual designs) accessible for creators with visual impairments, we explore how to make video authoring accessible to creators with visual impairments by representing the dynamic visual and audio content of a video as text.

2.2 Video Accessibility

Creating an accessible authoring tool for videos is challenging partially due to the inaccessibility of videos themselves. Videos are inaccessible to BLV audiences when the visual content in the video is not described by the audio (*e.g.,* travel videos with scenic shots set to music) [49, 50, 61]. To improve video accessibility, video creators [58], volunteers [36], or professional audio describers [10] can add audio descriptions to describe important visual content that is not understandable from the audio alone. As audio description is a challenging and time consuming task, prior work developed tools that provide feedback on audio descriptions [51, 68], host audio description authoring tools [10, 36], and help authors recognize mismatches between the audio and visual content so that they can add descriptions as they capture [61] or edit [50, 58] their video. Beyond helping authors create accessible videos, recent work also makes inaccessible videos accessible on demand by generating automatic [81] or interactive [60] visual descriptions. While these approaches provide BLV audience members access to visual content in videos, existing approaches have primarily been designed to make videos accessible for consumption rather than editing, such that they lack important information for authoring purposes (*e.g.,* lighting, camera stability). We investigate how to improve the accessibility of video editing by providing text descriptions of the visual content and quality of the video to help BLV creators efficiently understand video content and decide where and how to edit.

2.3 Text-based Audio and Video Editing

Audio and video editing is time-consuming as it requires multiple iterations of finding edit points and applying edits [23]. To improve the efficiency of reviewing, navigating, and editing audio or video footage, researchers and practitioners introduced text-based editing, which allows users to edit audio or video as they would a text document by time-aligning the words in the speech transcript with words in the audio [18, 25, 27, 34, 35, 42, 64, 65, 72, 77]. Researchers augmented text-based editing tools to further improve the efficiency of editing by: highlighting pauses or repeated words in the transcript [65], suggesting opportunities for B-roll [34], or matching voice-over recordings with their relevant video segments in narrated videos [77]. In addition, prior research used text-based editing to improve the quality of the video output by creating seamless transitions when cuts or dialogue changes occur in talking head videos [27], dialogue-driven scenes [42], and interview videos [18]. However, existing text-based editing tools were designed for sighted video editors who can visually inspect the video footage to identify editing opportunities, and visually skim the text transcript to navigate efficiently. We explore how to make text-based video editing accessible by integrating visual content and quality into the script, improving non-visual skimming via an outline, and making the editing tool screen reader accessible.

2.4 Video Navigation Interaction Techniques

Traditional video players such as Premiere [13] and Final Cut Pro [15] and editors let users navigate the video using a timeline. However, timeline-based navigation is challenging as users need to scrub back and forth to find the content of interest. To help users skim and navigate to content of interest, prior work has introduced approaches to navigate videos based on transcripts [38, 55, 56], high-level chapters or scenes [22, 26, 39, 55, 57, 78, 82], or key objects and concepts [21, 48, 60]. While transcripts help users efficiently search for words used in the video [38, 55, 56], they can be difficult to skim as they are often long, unstructured, and contain disfluencies present in speech [57]. To help users skim and navigate videos more efficiently, prior work segmented videos into high-level segments (i.e. scenes or chapters) and let users browse these segments based on representative thumbnails or text descriptions [22, 26, 39, 55, 57, 78, 82]. Prior work segmented videos into chapters or scenes by using the transcript to automatically segment the video based on transcript topics [26, 57, 78, 82], crowdsourcing to annotate segmentation points [39], or metadata such as command logs to segment based on interactions [22, 26, 56]. As it is particularly challenging for screen-reader users to skim text [14], we similarly segment the video to create an outline of important moments (*e.g.,* scene descriptions, visual errors) such that readers can quickly navigate our the audio-visual script using the outline. We also build on prior work that uses low-level features in the video (*e.g.,* keywords [21], presentation slide elements [60]) to facilitate search, as we similarly enable search via speech and visual objects (*e.g.* "socks") to help BLV creators locate footage to edit.

3 FORMATIVE STUDY

Prior work explores practices of content creators with visual impairments creating media such as photos [11, 17], drawings [41], documents [79], and audio [66], and explores community aspects of YouTube content creation such as high-level motivations for content creation and engagement with viewers [70], but existing work has not yet explored BLV creators' unique practices and challenges in video editing. To understand how BLV video creators edit videos, we analyzed 24 YouTube videos[1] (V1-V24) by 20 BLV creators about their video editing processes and interviewed an additional 8 experienced BLV video creators (P1-P8, Table 2) about their video editing motivations, current practices, and accessibility barriers[2]. Participants were recruited using mailing lists and compensated $

[1]See Appendix C for details on our video collection approach
[2]See Supplemental Material for the full list of questions

20 USD for the 1-hour semi-structured interview. We transcribed[3] the YouTube videos and interview recordings, then two researchers first independently coded all videos using open-coding [30], then met to resolve conflicts. The list of codes was consolidated after reaching a consensus and updating the codes. Then, the same two researchers used affinity diagramming [31] to group the codes into themes: goals for editing videos, strategies for editing videos, and challenges in editing videos.

3.1 Findings: Goals for video editing

Interview participants reported that their motivation for editing was to make videos more engaging (6 participants), or tailor videos to a specific audience (2 participants). As P2 summarized: *"I only keep the highlights [...] because short and snappy videos are more popular"* (P2). When editing, 5 participants mentioned that they polished their videos by editing out visual or audio mistakes, and 4 participants highlighted that they make videos concise by removing unimportant footage. For example, participants mentioned they removed audio mistakes like 'um's and 'ah's in the video (P2, P3, P5), pauses in the speech (P5, P7), or answering a question incorrectly in class (P3). While participants currently edited primarily via the video's audio track, they were often creating for a broad audience: *"My video is not just for people with visual impairments. For sighted viewers, I want to make sure nothing is visually awkward"* (P7). P2 added that editing the visuals is particularly crucial for BLV video creators as they often make mistakes while capturing video (*e.g.*, filming with lights turned off), and re-filming can be a burden. Finally, in addition to cutting out unimportant footage and mistakes, participants wanted to capture viewer attention by adding music and intros to their videos.

3.2 Findings: Strategies for video editing

Describing visual content and mistakes. BLV creators are unable to recognize the visual content in a video (*e.g.*, objects, actions, background setting) unless the visual content is understandable from the audio alone (*e.g.*, described by a narrator, or accompanied by sound) [49]. Thus, most participants mentioned that they verbally described visual content (*e.g.*, where they are and what they are doing) as they filmed their video. Participants reported a dual benefit to visual descriptions: identifying visual content while editing, and making the final video more accessible to BLV audience members. In addition to describing visual content, P3, P4, and P8 added verbal cues for editing when they made mistakes during filming. For example, P8 explained *"When I drop something, I'd say out loud 'Don't use the earlier part' so that I will easily remove it later"*. P7 dealt with a lack of information about the visual content and quality by focusing on the audio: *"Because I cannot check the visual quality of the footage, I am very picky about the audio. If there is some traffic noise, I don't use that part."* P8 renamed his video files with visual descriptions or editing cues so that he could locate relevant clips without playing the video.

However, creators' reported that their descriptions were inaccurate or incomplete when they were not aware of all relevant mistakes (*e.g.*, bad lighting, blur, poor framing or composition) or visual content. P2 recalled that once *"When I was pointing at an object describing it, it wasn't there!"*. Creators also shared that constantly describing visual content during filming took attention away from being creative (V19) and forced them to replace their audio track with music when they did not want the narration to be included in the final video (P7).

Identifying accessible video editing tools. To edit videos, seven participants used timeline-based editing tools (*e.g.*, Final Cut Pro) and one participant used FFmpeg, a command-line tool (P4). Participants noted that video editing tools were largely inaccessible: *"Finding an accessible editing tool in the first place is difficult. Very few tools are accessible themselves and also have accessible documents or tutorials."* (P4). Participants identified accessible video editing tools via other BLV creators and then learned how to use these tools with a screen reader via text tutorials, videos aimed at BLV editors, and official documentation. Even with the most accessible timeline-based tools, participants reported that the menus, buttons, and sliders were often unclickable with a screen reader or not properly labeled (*e.g.,* only reading 'button' instead of describing the function). In addition, such tools have complex menus that are difficult to navigate with a screen reader: *"Having no access to GUI, I have to continuously tab to locate the button of my interest. This becomes tedious because video editing is a complex task"* (P1).

Navigating videos linearly. While most BLV creators edited their videos with timeline-based tools, the visual aids that these tools provide for sighted creators to browse, skim and select video footage (*e.g.*, visual thumbnails to preview video content by hovering, audio waveforms to preview start and end of speech) are not accessible to BLV creators. As a result, all participants reported that they review and edit videos by first watching the entire video all the way through, and either editing as they go (6 participants) or noting timestamps to edit later (P1, P4). P7 noted that he usually watched the entire clip several times because he cannot jump from place to place in the video. To avoid re-watching long videos in order to find content of interest, participants commonly filmed multiple short clips: *"Because navigating within a single clip is so tedious, I never create a long clip."* (P8).

Recruiting sighted collaborators. BLV creators commonly sought out assistance from sighted people for filming, editing, and reviewing the final video (V1, V3, V6, V19, P2, P3, P7, P8). For example, the creator of V3 has her sighted husband set up a camera for filming, make video intro templates, and apply color correction. The creator of V1 and V6 recently hired an editor for even basic editing tasks as editing takes too long causing back pain. Before publishing their video, 4 participants (P2, P3, P6, and P8) wanted a sighted person with or without video editing experience to view the video and provide a sense of how an "average viewer" would see it. For example, P3 often uses Be My Eyes[4] or Aira[5] to ask a volunteer to provide feedback on visual quality (*e.g.,* whether she is centered in the frame and well-lit). All interview participants mentioned that they wished to edit videos independently, as sighted assistance is not always available or affordable, and they wanted to gain control over the process as creators.

[3] https://clovanote.naver.com

[4] https://www.bemyeyes.com/

[5] https://aira.io/

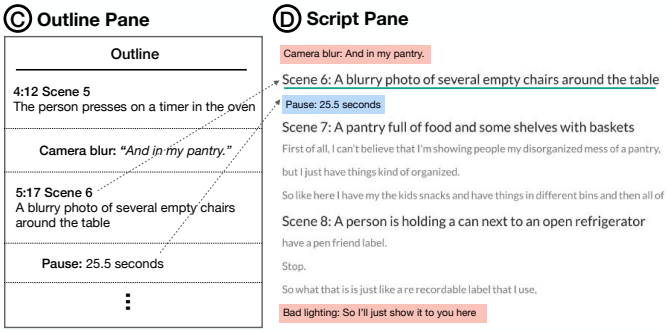

Figure 2: AVscript's outline pane displays a navigable summary of the audio-visual script including the high-level scenes and potential edit points (pauses and visual errors).

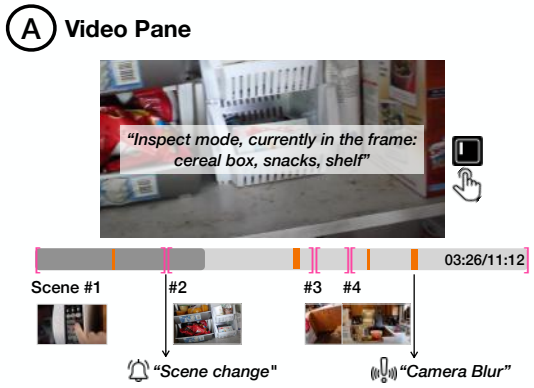

Figure 3: AVscript's video pane provides two types of audio notifications: 1) scene change notification (page-flipping sound) and 2) visual error notification (warning sound). By pressing an 'i' key, users can activate the inspect mode to access the objects in the frame.

3.3 Reflection: Opportunities for BLV creator support

While BLV creators resourcefully crafted strategies to work around inaccessible video editing tools, creators' remaining challenges (C1-C5) point to opportunities for technical or social support:

- **C1.** Recognizing visual content in a video (*e.g.*, setting, actions)
- **C2.** Assessing the visual quality of a video (*e.g.*, lighting)
- **C3.** Accessing editing tool menus with a screen reader
- **C4.** Non-linear browsing and skimming of videos
- **C5.** Performing visual edits (*e.g.*, color correction)

Our formative study indicates that BLV creators currently extend time, effort, creative agency, and social resources to overcome these challenges. For example, by narrating the visual content and noting mistakes as they film (**C1, C2**), losing and regaining editing task focus to navigate menus (**C3**), spending time watching a video linearly rather than jumping to the point of interest (**C4**) and recruiting sighted collaborators for inaccessible or overly tedious tasks (**C1-5**). Our work explores how to make video editing more accessible by providing creators' access to video visuals (**C1, C2**) and more efficient by improving the ability of creators to skim and browse for content of interest (**C4**), while the remaining challenges (**C3, C5**) indicate rich opportunities for future research and commercial accessibility improvements.

4 AVSCRIPT

AVscript (Figure 1) makes video editing accessible and efficient for BLV creators with *audio-visual scripts* that let creators navigate and edit based on a text script of visual content, visual quality, and speech. We first illustrate how BLV creators can use AVscript to edit videos through an example use scenario. Then we describe AVscript's interface and the computational pipeline that powers it.

4.1 Editing a video with AVscript

Anna, a YouTube content creator with a visual impairment, filmed a new cooking tutorial video to upload to her channel. Anna wants to improve the conciseness and quality of her tutorial to make it engaging to viewers, so she imports the tutorial video into AVscript to edit it. Using her screen reader, Anna first skims the outline to review her footage (Figure 2). Because the outline summarizes the video with the description of each scene, she can easily recall the content and plan what to edit from the footage. Reading through the outline, Anna notices that the second scene, where she shows her pantry, is over ten minutes long. To shorten the scene, she clicks the item of the outline and jumps to the pantry scene. As she plays the video from the start of the pantry scene, Anna hears a notification indicating that there is a visual error (Figure 3). To check the error, she pauses the video. As the position of her cursor in the audio-visual script updates alongside the video progress, she can easily read the audio-visual script which indicates there was a camera blur. To learn more about the visuals at that point, she presses an 'i' key to inspect the frame. As AVscript reads out the objects detected in the frame, Anna notices 'door' and 'hand' and realizes that the camera was shaking as she tried to open the pantry door. To remove the blurry footage, Anna selects the line that contains 'camera blur' and deletes the text. Anna also remembers that she spent a long time silently waiting for the microwave to finish while filming. She searches for 'microwave' (Figure 4) to find where the microwave appeared in the video and clicks on the relevant result. She shortens the pause by using the 'speed change' feature to make the clip two times faster.

4.2 Interface

The AVscript interface consists of: a *video pane* (Figure 3A), a *script pane* (Figure 2D), an *outline pane* (Figure 2C), a *tool pane* (Figure 4D), and a *search pane* (Figure 4E).

4.2.1 Video Pane. The *video pane* displays the video and the timeline (Figure 3). As the user listens to the video, the system provides sound notifications for the key visual events (e.g., "Scene Change", "Camera Blur"). Users can access visual information in the current frame by pausing the video and pressing the 'I' key to activate *inspect mode*, which reads out a list of detected objects in the frame.

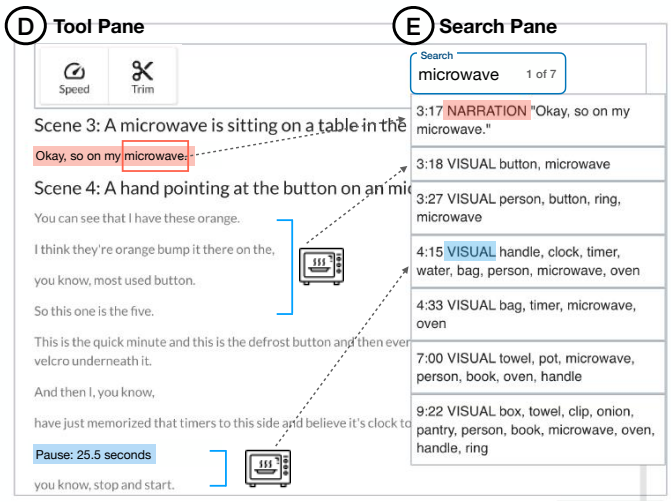

Figure 4: AVscript supports search over the transcribed speech and visual objects in the video. Creators can skim the results and click on a search result to jump to the corresponding point in the video.

4.2.2 Script Pane & Outline Pane. The *script pane*'s *audio-visual script* (Figure 2D) displays the narration and pauses in the video speech along with high-level visual scenes and visual errors. The audio-visual script is aligned to the video, so navigating within the script will navigate within the video (and vice versa), and edits to the script (*e.g.*, selecting and deleting a sentence) are reflected in the video. The script pane first includes lines that represent each sentence and comma-separated phrases greater than three words in the transcript (*e.g.*, "First of all..."). To inform users about the scene changes in the video, AVscript provides high-level scene headings in the script that summarize the key visual content in the scene (*e.g.*, "A person is holding a can next to an empty refrigerator"). In addition to the scene information, AVscript also provides recommendations for potential edit points which are highlighted alongside the text that occurs at that time. For visual errors (highlighted in orange), the system describes the type of error (e.g., "Bad lighting"), and for long silences (highlighted in blue), the system provides the duration of silence (*e.g.* "25.5 seconds"). AVscript's audio-visual script is designed to enable screen reader users to easily navigate the video at different levels of granularity (high-level visual scenes, narration or pause lines, and words) using key commands (*e.g.*, ctrl/cmd + →/← to jump forward or backward by a line).

In the *outline pane* (Figure 2C), the scene headings and recommendations for edit points are listed and sorted in timeline order to provide an overview of the major visual events. By clicking an item in the outline, creators can directly jump to the corresponding part of the video, with an updated cursor position in the script.

4.2.3 Tool Pane & Search Pane. To edit the video with AVscript, the creator selects a segment in the script and either presses delete (i.e. backspace) to remove that part of the video, shortens the segment by adjusting the start and end time with the "Trim" tool, or changes the playback speed of the segment by using the "Speed" tool. When creators have a specific editing target in mind (*e.g.*, a microwave), they can use *search pane* (Figure 4) to query a speech word, a visual object, or a visual error (*e.g.*, "microwave", "Camera Moving", or "Pause"). Then, creators can review and select a search result to jump to the start time of the result in the video and script.

4.2.4 Implementation. We implemented AVscript using React.js, HTML and CSS for the front-end web interface and Firebase for the back-end interface. For embedding a video player, we used Remotion [6] for efficient server-side rendering and parametrization. For *audio-visual scripts*, we used Draft.js [2], a text editor framework for React. We followed the guidelines of W3C [80] and tested the compatibility of the AVscript with all three major screen readers: *NVDA*, *JAWS*, and *VoiceOver*.

4.3 Computational Pipeline

AVscript's computational pipeline (Figure 5) transcribes and aligns video speech (Section 4.3.1), detects objects and segments scenes (Section 4.3.2), and detects visual errors (Section 4.3.3).

4.3.1 Transcribing and Aligning Speech. To enable word-level editing, AVscript transcribes the video speech using Otter.ai [6], then uses P2FA to align each word in the transcript to the corresponding word in the speech. Following Rubin *et al.* [65], we use CMU Sphinx Knowledge Base Tool [1] to obtain word phonemes of out-of-vocabulary words (*e.g.*, the coffee machine name "Keurig"). To enable phrase-level navigation and editing, AVscript then splits the transcript into sentences and comma-separated phrases that are three words or longer. AVscript also creates pause segments for any pause longer than three seconds. As widely used screen readers (*e.g.*, VoiceOver, NVDA, JAWS) read the text in HTML 'input' elements line-by-line, we place each phrase and pause on a different line for ease of screen reader navigation.

4.3.2 Segmenting and Labeling Scenes. Using OpenCV we extract frames from the video at a rate of one frame per second. For each frame, we detect objects in the frame using Detic [84] to retrieve visual information of the content for frame inspection, visual search, and detection of major visual changes for scene segmentation. In our pilot experiments, using all objects detected in the frame resulted in too much irrelevant information passed to the pipeline or presented to the creator (*e.g.*, listing all the objects in the background such as a coffee mug, spoons, forks). To limit our object detection to objects that are likely to be important, we only detect the objects referred to in the narration. To create a custom vocabulary set, we use Spacy's part-of-speech tagger [33] to extract all noun phrases in the transcript ('NN': noun, singular or mass, 'NNP': noun, proper singular, 'NNPS': noun, proper plural, 'NNS': noun, plural). Then we pass the custom vocabulary to Detic [84] to detect all instances of each noun in each frame. Detic provides the bounding box for each noun in each frame and a confidence value. We include all objects with a confidence value greater than 0.3. In the inspect and search mode, AVscript reads objects in order of the size of their bounding box (largest first). We segment videos into higher-level scenes by using a sliding window of width 4 to compare % of similar objects in the 2 frames before and 2 frames after a potential

[6] https://otter.ai/home

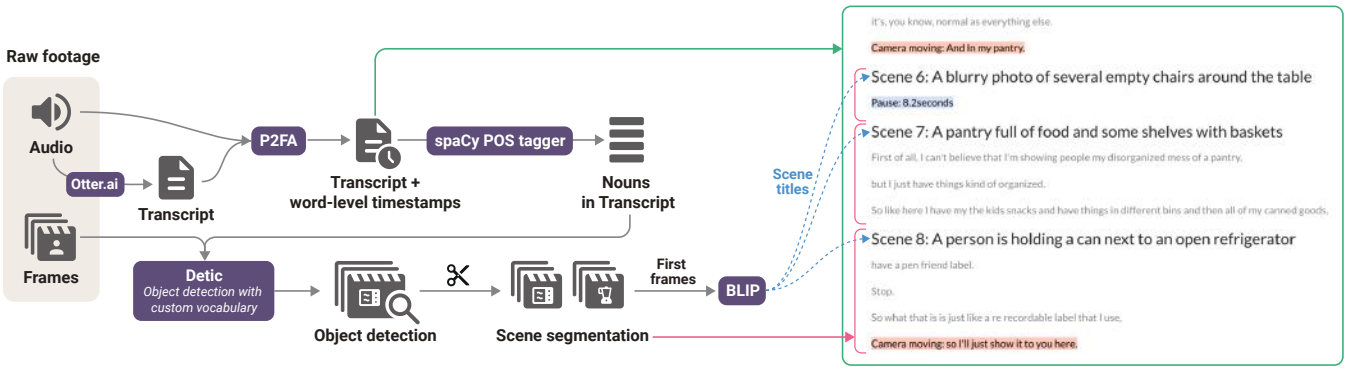

Figure 5: Computational Pipeline of creating an AVscript from raw footage. It takes two inputs: audio and frames. To generate an aligned transcript, we obtain the transcript from audio using Otter.ai and align using P2FA. To segment the footage into multiple scenes, we first detect objects in the frame with Detic using the noun extracted from the transcript as custom vocabulary. Then, we segment the footage when there is a salient change in the objects detected in the frame. For each scene, we generate the caption of the first frame using BLIP and use the caption as the scene title in the transcript.

boundary (similar to Haq et al. [29]). If a scene boundary occurs in the middle of a phrase boundary, we adjust the scene boundary to match the phrase boundary. We cut short scenes that did not encompass any entire phrase. Then, to obtain a description for each scene, we generate the caption of the first non-blurry frame of each scene using a BLIP [47]'s pre-trained model (CapFilt-L) with nucleus sampling. While BLIP produces state-of-the-art image captioning performance, BLIP occasionally misidentified objects, misgendered people, and cited incorrect emotions in pilot experiments.

We evaluated our scene segmentation on two videos (V1 and V2 in Section 5.1.2) by comparing our predicted scene boundaries to scene boundaries independently labeled by two researchers (Coders A and B who are authors of this paper). We measured percent similarity (i.e. Jaccard Index) between each set of scene boundary labels by dividing the number of matching labels by the total number of labels, considering any labels less than 3 seconds apart as the same. For V1, coders A and B shared 34% matching boundaries with each other, while our segmentation algorithm shared 37% and 38% matching boundaries with coders A and B respectively. For V2, coders A and B shared 57% matching boundaries with each other, while our segmentation algorithm shared 64% and 48% matching boundaries with coders A and B respectively. Overall, our segmentation algorithm achieved similar agreement with human coders as they did with each other. When disagreements occurred they typically represented high-quality segmentation boundaries provided at different levels of granularity (*e.g.*, a single segment for adding ingredients vs. three segments for adding flour, water and salt).

4.3.3 Detecting Visual Errors. Researchers have identified that the common components of photo quality that BLV people find difficult to achieve are blur, lighting, framing, and composition [11, 19]. Among these, AVscript supports identifying blur and bad lighting, and also considers camera motion blur to support video rather than photo content. To detect dark lighting, for each frame in the video we reduced the size of the frame to 100x100 to reduce the computation, then classify the frame as "dark" if the mean pixel luminescence value falls below an empirically determined threshold of 0.25. To detect blurry frames, we use the modified Laplacian method [59]. For each frame, we first convert the image to grayscale using OpenCV and then compute the variance of Laplacian to calculate the focus score. Then, we classify the frame as "blurry" if the focus score falls below an empirically determined threshold of 5. Using the detection results of each frame, we mark a segment as 'dark' or 'blurry' when more than three consecutive frames are identified as such. Finally, to avoid naively identifying all the camera moving parts (*e.g.*, facing the camera to a different object) as 'blurry', we also used the object detection results to detect 'camera moving' between scenes (frequent change in the object set over time). For segments that were classified as both "blurry" and "camera moving", we label them as "camera moving" to indicate that the motion blur may make objects in the frame difficult to see.

We evaluated our error detection pipeline on two videos (V1 and V2 in Section 5.1.2) by first creating a set of ground truth labels of visual errors based on existing video editing guidelines [3–5, 7–9]. Two researchers first met to group established guidelines into common themes (See Supplemental Material for aggregated guidelines), then each researcher annotated edit points for a single video (V1) and met to resolve conflicts and revise the guidelines. After reaching a consensus, one of the researchers annotated the other video (V2) following the revised guidelines. In total, the ground truth labels included 18 errors for V1 and 15 errors for V2. When compared with the ground truth edit points, AVscript's pipeline achieved high precision and low recall for visual error detection (precision=38.89%, recall=100% for V1, precision=46.67%, recall=87.50% for V2). The most common error type not detected by our pipeline was a partial blur due to the main object being out of focus because our pipeline only calculates the focus score of the entire frame. One of the reasons for the high precision and low recall is that we empirically set the threshold of AVscript's pipeline low to avoid false notification of errors, or presenting users with too many error suggestions.

5 COMPARISON STUDY

We conducted a within-subjects study to examine how AVscript impacts experienced BLV creators' video editing practice compared to their personal video editing tools.

5.1 Method

5.1.1 Participants. We recruited 12 participants who all had a visual impairment, used a screen reader to access their device, and had prior experience editing videos (8 males and 4 females, aged 28–58) using mailing lists and social media (Table 2). Three of the participants also participated in the formative study (P1, P4, P8). Among the participants, ten participants had a YouTube channel where they posted videos to the public, while two participants shared their videos privately (*e.g.*, to the company they work for or family and friends). All 12 participants mentioned that they create videos for both sighted and BLV audience members. Participants authored a variety of videos including vlogs, tutorials, product reviews, presentations, and more (see Table 2 for a complete list). To edit videos, 11 participants used timeline-based editing tools (Reaper, Windows Movie Maker, Microsoft Photos, Final Cut Pro, and VideoReDo), and 1 participant used scripting tools (FFmpeg, Python). Reaper is primarily an audio production tool, but it also supports videos. All participants used one or more of the three popular screen readers (NVDA, JAWS, VoiceOver) which are all compatible with AVscript.

5.1.2 Materials. We selected three videos from YouTube authored by BLV creators that contained (Table 3): primarily raw video footage with few edits, real-world camera footage rather than screen recordings, and narration in English by the video author. We selected a short video (V0) for the tutorial. Videos used in the main sessions (V1-V2) were created by the same YouTube creator[7] and were selected to be similar in terms of length, amount of narration, and shot changes. For both videos, we only used around the first 11 minutes of the video such that participants could edit the video within the study time. For each video, we did not manually correct algorithmic results except for replacing the incorrect gender identification of the speaker.

5.1.3 Procedure. We conducted a 120-minute remote study on Zoom[8] where all participants had a 1:1 session with one of the researchers. We first asked participants demographic and background questions about their prior video editing experience. We then gave a 20-minute tutorial on the AVscript interface in which participants edited V0 to learn system features. Participants then edited one video (V1 or V2) with AVscript and the other video (V1 or V2) using their existing editing tools (within subjects). The order of system type (their own editing tools vs. AVscript) and video clips (V1 or V2), was counterbalanced and randomly assigned to participants. During the task, we answered participant questions about AVscript's screen reader controls and the amount of task time remaining but did not provide any help with understanding or editing the video. We encouraged participants to take a short break between two sessions. For each interface, we conducted a post-stimulus survey that included three types of questions: NASA-TLX ratings, ratings about the final video output, and ratings about the perceived helpfulness of system operations. As we did not provide assistance with video understanding or editing during the study, ratings related to *assistance* (Figure 6) intend to capture participants' perceptions of their ability to use each tool independently. All ratings were on a 7-point Likert scale. After the session using AVscript, the edited video was downloaded to our server. We also asked the participants to share the output video edited using their personal video editing tools. At the end of the study, we conducted a semi-structured interview to understand participants' strategies using AVscript and the pros and cons of both AVscript and their own tools. We compensated participants with a 40 USD Amazon Gift Card. This study was approved by our institution's Institutional Review Board (IRB).

5.1.4 Analysis. We collected the video recordings, the interaction logs, the output videos, and the survey responses to perform both quantitative and qualitative analyses. AVscript's interaction logs were collected using Google Firebase[9]. We reviewed both the session recording and interaction logs to extract the *operations* participants performed using their baseline video editing systems and AVscript. We triangulated the logs with the output videos to validate the extraction (*e.g.,* comparing the edit points in the video to the edit operations). We did not count the number of primitive actions such as moving cursors without playing a video or subsequent playing/pausing of the same video segment. We transcribed the exit interviews and participants' spontaneous comments during the tasks and grouped the transcript according to (1) strategies of using AVscript and (2) perceived benefits and limitations of our system.

5.1.5 Study Limitations. In this study, participants used AVscript for the first time and compared it to their own editing tools, which they are already familiar with. Thus, this study neither reveals how long-term use might impact the editing experience of users, nor how participants who have never edited videos before might use the system. We selected the video length (around 11 min) and the editing time provided (around 30 min) to balance providing a realistic use scenario while keeping the study time short, especially as editing is cognitively demanding. As a result, not all participants were able to complete editing within the time provided.

5.2 Results

Overall, participants rated using AVscript to edit videos as requiring significantly less effort (μ=4.58, σ=1.51 vs. μ=2.17, σ=1.11; Z=2.96, p<0.01), frustration (μ=3.58, σ=2.11 vs. μ=2.08, σ=1.31; Z=2.39, p<0.05), mental demand (μ=3.17, σ=1.80 vs. μ=2.00, σ=0.95; Z=2.03, p<0.05), and temporal demand (μ=4.5, σ=1.93 vs. μ=2.58, σ=1.16; Z=2.54, p<0.05) compared to their own editing tools that they were experienced with (Figure 6). Perceived performance and physical demand were not significantly decreased for AVscript, and all significance testing was performed with the Wilcoxon Signed Rank test. All participants stated they would like to use AVscript in the future for reviewing and editing their videos.

We report the statistics of the videos edited by participants in Table 1. While 30 minutes were given for each editing session, six

[7] https://www.youtube.com/c/BlindMovingOn
[8] https://zoom.us/
[9] https://firebase.google.com/

participants using AVscript finished the task early. Due to the limited time, ten participants using their own editing tools did not edit the later part of the footage (P4, P8, P9, P11-17). The Video Timeline column in Table 1 shows the edited time segments over the timeline of the videos. As participants using their own tools often did not reach the second half of the video, the output videos in the baseline condition included notable errors in the latter half of the video such as leaving in dark scenes (V1), long pauses (V2), and repetitive actions (V2). However, across both conditions, short edits to the video timeline often introduced jump cuts [5, 7] in the final output video.

Figure 7 summarizes how creators used AVscript by visualizing operation sequences relevant to navigation and editing. Overall, participants frequently jumped between different parts of the video using the headings, transcript lines, and words in the *audio-visual script* (Figure 7, light blue "Text Jump" cells). Participants did not actively jump using the search; four participants (P10, P14, P16, P17) used the search feature once (Figure 7, blue "Search Jump" cells). Participants frequently deleted speech, pauses, and visual errors in the video (Figure 7, yellow, orange and red "deletion" cells). Because AVscript's *audio-visual script* is aligned with the video timeline and contains descriptions of pauses and errors (*e.g.*, duration, error type), five participants (P4, P8, P9, P16, P17) often subsequently deleted problematic segments only using text descriptions without actually playing the video. In addition to deleting clips, participants tried to recover from pauses and visual errors by trimming or changing the speed; five participants trimmed the pause segments (P8, P10, P15, P16, P17) and one participant changed the playback speed (P1).

5.2.1 Reviewing Videos and Identifying Errors to Edit. Participants rated AVscript as significantly more helpful for reviewing their video footage to identify errors compared to their existing editing tools (μ=4.25,σ=2.22 vs. μ=2.00,σ=1.04; Z=2.17, p<0.01). When reflecting on their final video, participants expressed they were more confident with their final result (μ=4.67,σ=1.37 vs. μ=3.00,σ=1.41; Z=2.34, p<0.01), and needed less assistance reviewing it (μ=5,σ=1.54 vs. μ=2.75,σ=1.66; Z=2.31, p<0.01) compared to their typical editing tools.

Text-based vs. Timeline-based Video Review. Using AVscript, participants primarily reviewed the video by reading the text of the audio-visual script and outline, while with their own video editing tools participants primarily reviewed the video by playing the video. For example, of 7 participants who reviewed the entire video before editing it with AVscript, five participants read the entire audio-visual script using a screen reader or braille display without playing the video (P4, P9, P10, P16, P17), and three read the outline to gain an overview of the video (P10, P11, P13). P10 did both. On the other hand, when using their baseline tools, all participants played the video from the beginning to identify points to edit. Participants expressed that reading the text of the audio-visual script or outline allowed them to skim the footage faster than the video alone. P16 who reviewed the 11-minute video with AVscript in 3 minutes remarked, *"I've been using NVDA [screen reader] for so long that I can understand a very fast TTS [Text-To-Speech]. Because I read 1,075 words-per-minute reading the script instead of playing video saves so much time for me."*

Gaining an Overview of Visual Content and Errors. In addition to providing options for faster review, participants reported

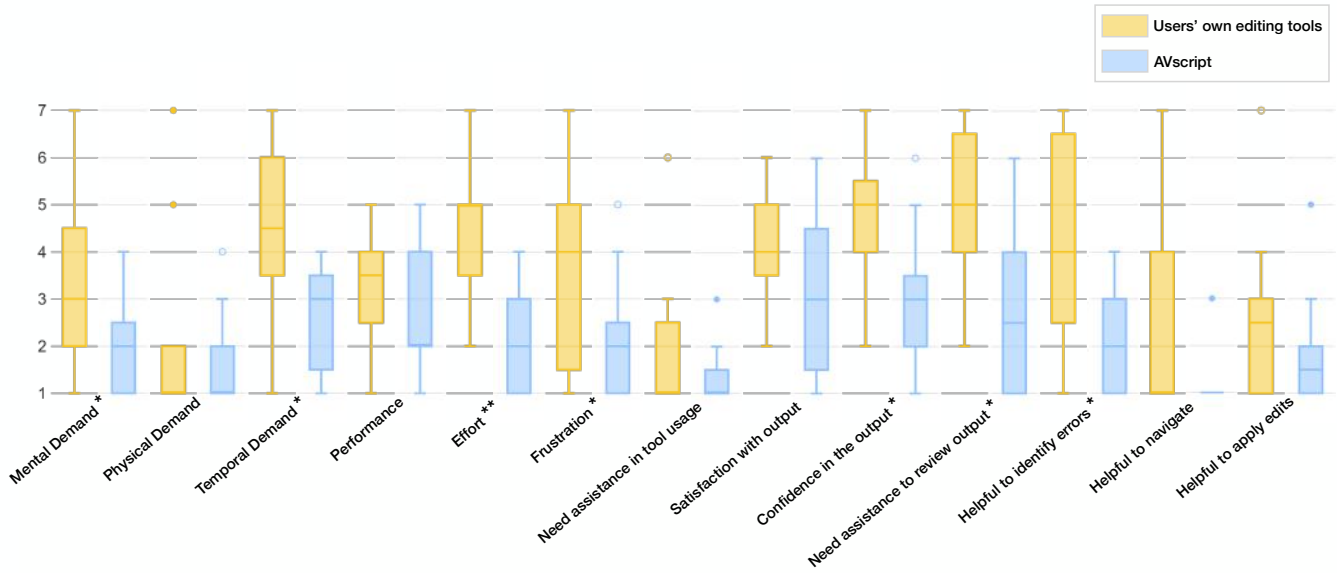

Figure 6: Boxplot of mean scores for the participants' personal editing tools and AVscript (1 = low, 7 = high). Note that a lower value indicates positive feedback and vice versa. The asterisks indicate the statistical significance as a result of Wilcoxon text (p < 0.05 is marked with * and p < 0.01 is marked with **). AVscript significantly outperformed users' own tools in mental demand, Temporal demand, effort, frustration, confidence in the output, independence in reviewing output, and helpfulness in identifying errors.

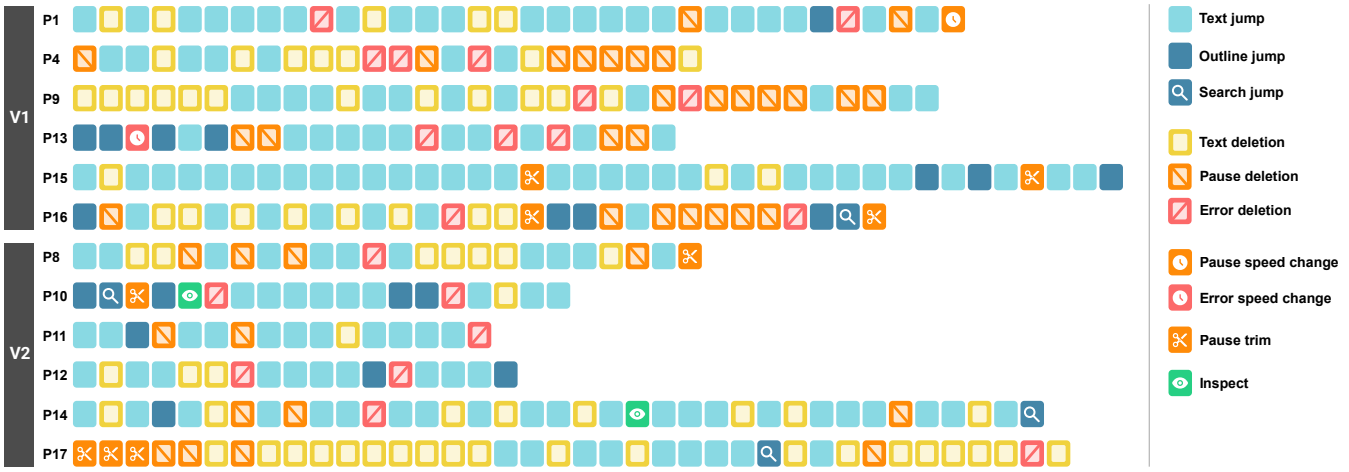

Figure 7: Sequences of operations that are relevant to AVscript's navigation and editing features by participant (comparison study). Participants' data is grouped by video ID and then sorted by participant ID. Note that trivial cursor movements in the transcript without triggering the video control were not included for brevity.

Table 1: Summary of the number of edits made by participants using their personal video editing tools (Baseline) and AVscript. Edit time refers to the time spent on editing each video, and length refers to the duration of the output video. We also report the total number of edits (d: deletion, s: speed change, t: trim, a: audio effect, i: insertion). Note that 'trim' has the same effect as deletion but is counted as separate in AVscript session to distinguish text-based deletion and timestamp-based trim. The video timeline visualizes the position of edits in the video.

PID	Edit Time		Length (mm:ss)		Total # of edits		Video Timeline	
	Baseline	*AVscript*	*Baseline*	*AVscript*	*Baseline*	*AVscript*	*Baseline*	*AVscript*
1	30m	30m	8m 2s	9m 35s	1 (1d)	10 (9d, 1s)	V2	V1
4	30m	30m	6m 39s	5m 57s	2 (2d)	12 (12d)		
9	30m	25m	6m 58s	8m 16s	5 (5d)	22 (22d)		
13	30m	20m	9m 10s	9m 9s	4 (3d, 1a)	8 (7d, 1s)		
15	30m	30m	10m 47s	9m 37s	6 (6d)	5 (5d)		
16	30m	30m	7m 30s	8m 39s	5 (5d)	19 (17d, 2t)		
8	30m	28m	10m 47s	7m 2s	5 (5d)	13(13d, 1t)	V1	V2
10	30m	30m	10m 51s	7m 54s	4 (4d)	4 (3d, 1t)		
11	30m	22m	8m 37s	10m 45s	21 (21d)	4 (4d)		
12	30m	18m	11m 4s	7m 51s	15 (15d)	5 (5d)		
14	30m	30m	10m 44s	7m 30s	8 (8d)	12 (12d)		
17	30m	25m	11m 24s	9m 26s	9 (4d, 4i, 1a)	27 (24d, 3t)		

that they used AVscript's high-level description of visual scenes and errors to (1) form a mental picture of the visual content (*e.g.*, connecting background sounds with the scene descriptions, or imagining what the scene contained), (2) plan what edits they would like to make later (*e.g.*, get an overview of the parts of the video that they needed to edit), and (3) mentally bookmark their progress as they edited (*e.g.*, using a scene title to remember they had left off editing). P16 remarked that the descriptions were particularly helpful for silences: *"Even in silence, I know what is going on in this video! Reading these scene labels, I can construct mental imagery of what the footage looks like."* P10 and P14 also used the *inspect* feature in concert with the high-level descriptions of visual content and errors (*e.g.*, to understand the content of a pause, and to access objects at the beginning of a scene).

Identifying Opportunities to Edit Video Footage. Participants considered AVscript's visual errors in making decisions for visual editing, while they edited audio errors (*e.g.*, pauses, and repeated words) with both systems. Using AVscript, all participants reviewed the visual errors in the video, and 11 of the 12 participants AVscript edited a visual error (*e.g.*, by deleting, speeding up, or trimming the

error). When evaluating visual errors, participants read the error along with the speech associated with the error to decide whether to delete it or not. For example, when assessing a visual error that overlapped with an important sentence in the speech that would harm the meaning of the speech if deleted, participants left the footage intact. On the other hand, if participants could make a natural edit (*e.g.*, cutting out an unnecessary sentence, or trimming the length of the error) they would cut it out. To edit the errors detected by AVscript, 11 participants deleted the entire segment of the error, whereas one participant changed the playback of the error segment leaving some part of it. P13 stated, *"If I just get rid of the error, it might result in a jumpcut or leave a too small gap between the sentences which is unnatural."* While participants expressed that getting informed of the visual errors made them more confident in their edits, but P4, P9, P11, and P12 noted they would like severity information about the error to inform the error vs. content trade-offs. P12 noted *"It says bad lighting, but what I want to know is how bad so that I can make a decision whether to keep it, fix it, or remove it."*

With both systems, participants edited out irrelevant footage and audio errors (*e.g*, pauses, repeated words). With AVscript, participants made edits at word level or line level (a sentence, a long phrase, or a pause) and sometimes removed multiple lines at once when they decided not to keep a big chunk of the scene that they did not find interesting. Using their own editing tools, all participants made edits to remove filler words or pauses between speeches, and some participants similarly deleted uninteresting content.

5.2.2 Navigating and Applying Edits. While participants received AVscript to be beneficial for high-level navigation and editing operations (*e.g.*, by scenes, lines, words, long pauses) and non-linear navigation, the current version lacked the fine-grained navigation and editing provided by their typical video editors that enables participants to edit fine-grained audio (*e.g.*, short pauses). As participants found AVscript to be helpful for some navigation tasks more than others, participants did not rate AVscript to be significantly more helpful for their existing tools for navigation (μ=2.5,σ=2.11 vs. μ=1.3,σ=0.78; Z=1.63, p>0.05) or applying edits (μ=2.58,σ=1.73 vs. μ=1.83,σ=1.19; Z=0.99, p>0.05).

Coarse-Grained Navigation. Using AVscript's audio visual script, all participants efficiently navigated the video content by moving the cursor in the transcript both line-by-line (up/down arrow keys) and word-by-word (alt/option + right/left arrow keys). P12 and P16 also jumped to the next scene in the audio-visual script by pressing the 'H' key in the screen reader's browser mode (used to navigate to the next heading element). As participants edited the video, 7 participants also used the outline pane to quickly navigate to a scene or an error suggestion. In contrast, using their typical video editors' timelines all participants navigated by skipping ahead in a fixed time or frame interval (*e.g.*, skip ahead 5 seconds) rather than by content (*e.g.*, sentence, word, pause, error or scene). Participants then needed to iterate multiple times to find the relevant cut point, as described by P11: *"To delete one word, I have to navigate so many times to precisely set the start and end of what I want to cut out. So I often create a small loop around the target just for editing."* Four participants also scrubbed backward or forwards to navigate to a near word or pause target (P8, P10, P11, P14) despite its disadvantages: *'The scrubbing audio makes no sense to me, but it can still be used to detect pauses"* (P11). P10 and P11 also used the tab key in Reaper to jump to the next audio peak to locate the end of long silences.

Fine-Grained Navigation. While AVscript was convenient to edit words or pauses, participants asked that in the future that AVscript also include frame- and interval-level navigation to facilitate fine-grained adjustments to the cursor placement, especially when speech is not present. In addition, as the system limited the pauses displayed to screen reader users to 3s long to optimize skimming the audio-visual script, participants expressed that they wanted a mode for fine-grained edits that would display small pauses.

Non-linear navigation. Participants also used AVscript to efficiently navigate the video non-linearly, using the outline to navigate to an error they would like to edit, then moving their cursors back to play from a few lines prior to figuring out where to make the edit by considering the audio content and the visual error together (P4, P9). Four participants used the search pane to find and skip to a specific part in the script (P10, P14, P16, P17). P10 exclaimed: *"This search feature is revolutionary! I can search not just for text, but an object or even pauses so easily."* Yet, participants who never used the search feature to navigate the video speculated that searching for visual content would be more useful for their own videos. P9 noted *"I didn't know what to search for as I didn't film this video. If I use it (AVscript) for my own video, I will definitely find it useful."*

Applying edits. The ability to apply edits with AVscript was limited to high-level edits of the video footage itself. With their own editing tools, participants additionally applied effects to improve the audio or visual quality of the footage, including: applying a high pass filter to remove background noise and heavy breaths (P13), inserting music and adjusted its volume so that the original audio is louder than the music (P17), adding an intro and credit to the footage by inserting a black image with white text (P17). After making a cut in the video, P15 and P17 used a transition effect to avoid the abrupt jump in the audio or visual. With When making edits, 2 participants often referred to a help menu, or a self-created list of hotkeys and commands to remember the keys they should use (P4, P8). Participants who didn't use the built-in video player of the editing tool read the timestamps from the player and then passed them into the command line (P4 using FFmpeg), or to the input field of the tool (P16 using VideoReDo). Both P4 and P16 noted the inconvenience of switching between two different interfaces. P16 said "*Because the video player and VideoReDo use different time formats, I cannot directly copy and paste. When I manually read and type them, I sometimes make typos and this makes very confusing results.*" P4 also mentioned "*While the script-based editing is very accessible, I have to run the command after each edit to check the results. If it's a long video, I have to wait for a long time for the video to be processed.*"

6 EXPLORATORY CASE STUDIES

Through the comparison study, we could learn that BLV creators can understand the concept of AVscript and actively use the main features to optimize given videos. As the next step, we conducted an exploratory study with 3 BLV creators (P14, P18, P19 in Table 2)

where the creators edited their own footage using AVscript to examine: How would BLV creators use AVscript to edit their own videos?

6.1 Method

We recruited 3 video creators with visual impairments who used screen readers to access their devices using mailing lists and social media (P14, P18, P19). P14 also participated in the comparison study. All three participants created and uploaded videos to their YouTube channels on a regular basis, and two of the three participants had not edited videos before. Before the study, we collected footage from each participant that they had filmed themselves (Figure 8)[10]. During a 120-minute remote study session, we asked participants background questions, provided a tutorial of AVscript, invited participants to edit their own footage with AVscript, and asked participants semi-structured interview questions about their experience (see Supplemental Material). We compensated participants $80 via Amazon Gift Card for filming their footage and participating in the study.

6.2 Three Vignettes: How BLV Creators Use AVscript in Context

6.2.1 V3: Growing with Bryan. Bryan (P18) regularly posts videos to demonstrate how nature is accessible on his YouTube channel. To film his planting demonstration video (V3), he strapped a camera to his chest or forehead to use both hands freely and filmed four clips over four different days. Because the first two clips were filmed more than a month ago, Bryan quickly reviewed the footage by skimming through AVscript's script. He mentioned *"I usually make videos comparing how a plant changed after several months. Using [AVscript], I don't have to watch all the clips again. I can save so much time reviewing and remembering what I filmed!"* With AVscript, Bryan first used the *outline* to jump to the start of each scene (clip), then deleted the first few lines of the script where he mentioned the date it was filmed. When he noticed that he pointed at a plant to describe it in the video, he used the *inspect* feature to make sure the plant was in the frame. Bryan described that with AVscript that he can be more independent in making videos, which will enable him to create videos more quickly. He explained that he typically required sighted reviewers: *"Because it is so difficult to make sure everything I mention is in the picture, I usually film the same content several takes, and ask sighted friends to ask which one is the most appropriate."*

6.2.2 V4: An Adventure to Dinner. Rachel (P19) is a content creator who makes a wide range of different media: podcasts, interviews, live streams, Vlogs, and tech demos. While she is an experienced audio editor, she has never tried editing a video due to a steep learning curve. For the study, she filmed a Vlog on her way to dinner (V4). Rachel mentioned that AVscript is easy to learn and use with a screen reader: *"Absolutely fantastic, I have never been able to edit videos before, but after 15 minutes of learning how to use this, I can edit my video. It's a giant leap forward."* While editing, Rachel found AVscript's search feature useful because she still remembered most content of the footage that she filmed two days ago: *"When I was walking on the street, I met a family and chatted with them for a while. To jump and edit that part, I tried searching for 'boy' or 'person'."* She enjoyed having the option to search for the visual content of the footage, as she might forget the exact word she said, but still remember what was visible in the frame. Rachel also noticed that AVscript had errors in the speech-aligned transcript and scene description. When she read one of the scene labels, she said *"Oh it says I'm holding a purple leash! That is my purple cane. I guess this is created by AI?"*

Figure 8: Three videos filmed by BLV creators for exploratory case studies. (Section 6).

Overall Rachel mentioned that she feels more confident showing her video to more people after fixing the visual errors detected by AVscript. As an individual without light perception, Rachel has often experienced filming videos with bad lighting (*e.g.,* turning the light off, or facing back to the sunlight). She noted *"[AVscript] is also guiding me on how to film with fewer visual errors."*

6.2.3 V5: Blind Construction Tools. Lewis creates workout videos, product reviews, and tips for people with visual impairments. He often shares the video on social media or participates in workout video contests. To film a video on construction tools (V5), P14 set up a camera with a tripod and used TalkBack to guide him with the filming position (e.g., TalkBack giving directions "Face detected - upper right"). To quickly skim through his footage, Lewis pressed and held the down arrow key to mimic the scrubbing feature of Reaper (his typical video editor). He described that the lines helped him navigate efficiently: *"I don't have to read the entire line to check where I am (the cursor is) in the video. Just listening to the first word or first syllable is enough."* When Lewis reached a part of the video that AVscript detected as *blurry* he mentioned: *"Oh this is not a bad thing here, I had to walk quickly, and it's probably because of that."* Lewis also used the *inspect* feature to choose an editing point. To find the first few seconds where he started the recording and was not on the frame, he continuously clicked *inspect* to find the exact timestamp where he appeared, then trimmed the video up to that point. He noted: *"The script does tell when the word begins and ends, but it doesn't tell when an action begins and ends."* Lewis also reported speech recognition errors: *"I mumbled something here, but it wasn't caught in the transcript. Maybe because of the radio music. It is difficult to edit that part out when I don't see it on the transcript."*

Using AVscript, Lewis anticipated that collaboration with sighted reviewers will be easier because he can show them only the errors detected by the system instead of asking them to review the entire footage. He also noted that he wanted to create different content and styles of videos with the help of AVscript: *"In the past, I always used a tripod to avoid camera shakes. Now that I can check whether my footage is shaky, I want to try carrying around my camera."*

[10]If participants provided multiple clips we concatenated them in order of time filmed

6.2.4 Reflection on Three Vignettes. All three BLV creators used AVscript to speed up certain stages in video editing (*e.g.,* Rachel browsing the video for a specific scene), or locate visual errors or actions (*e.g.,* Lewis noticing a camera blur) which wasn't possible prior to use AVscript. They also reported the limitations of AVscript: (1) errors in the speech-aligned transcript and scene description, (2) lack of detailed information on visual content such as motion details or object colors, especially for parts without much narration. Overall, all three creators wanted to use AVscript in the future to be more creative with the content and styles of videos.

7 DISCUSSION AND FUTURE WORK

In this section, we reflect on our findings from the design, development, and evaluation of AVscript. We also present future directions for research exploring accessible authoring tools.

Navigating Videos based on Visual Content. Our formative study revealed that BLV video creators' current tools did not enable access to visual content in their video footage (**C1.** Recognizing visual content in a video). To address this challenge, AVscript provides access to visual content including: a summary of key visual moments via *scene descriptions*, a list of low-level objects via *inspect mode*, and on-demand access to visuals of interest via *search*. While creators using AVscript occasionally listened to the video and scene descriptions linearly (similar to how BLV audiences currently listen to audio descriptions that describe visual content in a video alongside the video narration [44, 58]), creators also used scene descriptions for new use cases including skimming the outline of scene descriptions to gain an overview of visual content, and clicking on scene descriptions to navigate to video scenes (similar to how sighted video creators use video keyframes to navigate with timeline-based video editing tools [13, 15]). Scene description-based navigation helped address an existing challenge for video editors (**C4**. Nonlinear browsing and skimming of videos), and future work may explore extending this navigation approach to video consumers. However, state-of-the-art scene descriptions still include errors. In our studies, BLV creators editing their own footage were able to recognize and recover from errors that mismatched their expectations (*e.g.*, "leash" vs. "cane" in a walking video), but creators editing unfamiliar footage missed notable errors (a pile of laundry described as "animal on bed"). Future improvements to scene description accuracy could help AVscript better support BLV creators using unfamiliar footage (*e.g.*, when adding stock video b-roll). While creators used AVscript's low-level inspect mode less often than the high-level scene descriptions, one creator used inspect mode to achieve fine-grained navigation to a visual scene boundary, similar to the fine-grained navigation to audio pause boundaries that BLV creators currently perform via audio. Future work may explore how navigation practices change with long-term use of AVscript for video editing, and how to further facilitate fine-grained visual navigation.

Editing based on Visual Error Suggestions. To address the challenge of assessing the visual quality of a video (**C2**), AVscript informs users of potential edit points (blur, bad lighting, camera motion, and audio pause). Participants frequently used the visual errors provided by AVscript to remove distracting and low-quality visuals from the video (*e.g.*, camera shakes, dark lighting), and participants reported that edit point suggestions improved their confidence in their final video. While participants occasionally noticed errors in visual content descriptions, none of the participants expressed skepticism toward visual quality predictions. However, participants asked for information about the severity of the predicted visual errors to help them weigh the content and quality of a clip before removing it. In the future, AVscript will provide confidence scores and severity levels for predicted visual errors to better support BLV creators in making editing decisions. In addition, describing errors in more detail and explaining potential causes (*e.g.,*"*Blur – out of focus, possibly due to the camera being too close to the target object*") could help novice video editors understand errors and film better footage. Finally, AVscript could recognize other types of visual errors in the future, such as *composition errors* [12] and *jump cuts* [7] which we observed in the videos edited by BLV creators.

Text-based Video Editing for BLV Creators. Prior research on text-based editing primarily focused on sighted video authors [18, 34, 77]. We explored using text-based editing to help BLV creators navigate videos efficiently (**C3**). Our studies revealed benefits of using text-based editing that echo findings in prior work (*e.g.*, lower mental load than timeline-based interface [34, 77]), and demonstrated unique benefits for BLV creators (*e.g.*, better screen reader compatibility than timeline-based interfaces, and access to rapid screen reader text-to-speech for quick skimming and editing). Still, creators mentioned that timeline-based video editing interfaces enable granular access to the audio track without word-level constraints (*e.g.,* editing out background noise which is not captured in the transcript). In the future, we plan to integrate timeline-based editing into AVscript to enable creators to use the *audio-visual script* for coarse navigation and the timeline for fine-grained navigation.

Supporting New Editing Tasks. While our system explored deleting or speeding up video segments — core tasks in video production — future can explore how to support BLV creators in making visual edits such as composing title slides or adding visual effects. For example, an editing system could describe the impact of an applied effect on the visual content in the video (*e.g.*, "the vignette effect now covers the hands") using techniques from prior work in BLV visual design authoring [62] and computer vision approaches for captioning differences between pairs of similar images [37]. Recent strides in prompt-driven text generation [20], image generation [63, 67], and image editing [54] suggest that prompt-driven video editing (*e.g.*, make this clip moody) may be possible in the future [32]. Future research is needed to help BLV creators evaluate their results with such tools. In addition, AVscript considers single-track videos as the format common in BLV creators' videos today. However, in the future, we could explore approaches to help creators enhance their videos with b-roll (*e.g.*, by helping creators find and insert their own footage using text [77] or suggesting opportunities for adding online b-roll [34]).

Supporting New Stages of Video Production. Our formative study suggested that BLV creators currently use creative but effort-intensive filming strategies (*e.g.*, describing visual content) and editing strategies (*e.g.*, navigating video footage only linearly) to produce and share their videos to broad audiences. As AVscript enabled BLV creators to edit videos more efficiently, with less mental demand, and more confidence in their end result, BLV creators

mentioned it would change their filming practice by capturing additional desired footage. Future work may explore how improved access to video editing will impact filming practices, and further improve the filming process by providing additional information about the visual content and errors, as provided in our system, at capture time. Similar to prior work in supporting BLV photographers, future systems could information about framing the shot [11] paired with the presence and severity of potential visual errors. When BLV creators move as they film videos like Vlogs and tutorials, approaches to alert creators of potential errors may distract them (similar to the demand of describing visual content). Thus, future work could explore enabling BLV creators to capture with a wide field of view at capture time (*e.g.*, 360 or 180-degree video) and edit the video to produce a smooth normal field of view footage that captures the content of interest [76]. Finally, our work points to solutions in the video publishing process including improving the acceptance of sighted audiences to visual errors, and platform-supported funding for BLV creators seeking to hire sighted reviewers.

Beyond Manual Text-based Editing. We designed AVscript to use text as BLV video creators who we interviewed were highly proficient at using screen readers, and text enabled creators to use their screen reader experience to review and navigate video at high speeds. We are currently exploring multimodal approaches for editing videos by combining a screen reader and voice input together to facilitate fast and low-burden navigation and editing (*e.g.*, "*jump to 5 minutes*", "*delete this scene*"). The visualization community has demonstrated ample applications that support multimodal data exploration with touch and speech (*e.g.,* [40, 73–75]). In similar vein, we plan to build on work in tactile displays [83] to surface the visual content in the video. While consuming video with tactile displays may be challenging, editing video may benefit from providing creators access to slow frame-by-frame content (*e.g.*, to assess when a person moves out of the frame) and waveform visualizations.

8 CONCLUSION

In this work, we designed and developed AVscript, an accessible text-based video editing tool that enables BLV creators to edit videos using a text script that describes the visual content and visual errors in the footage. The design was informed by a formative study consisting of YouTube video analysis and interviews with BLV creators. The comparison study (N=12) showed that AVscript significantly reduces the mental demand of BLV creators when compared to their own video editing tools. In the exploratory case study (N=3) we also explored how BLV creators edit their own videos using AVscript. We hope our research catalyzes future work on improving accessibility in media authoring process.

ACKNOWLEDGMENTS

We want to thank Naver AI Lab for supporting the work.

REFERENCES

[1] 2021. cmu sphinx knowledge base tool version 3. http://www.speech.cs.cmu.edu/tools/lmtool-new.html
[2] 2022 (accessed Dec 12, 2022). https://draftjs.org/
[3] 2022 (accessed Dec 12, 2022). Bright trip - How To Vlog. https://assets.ctfassets.net/bhsr3r63z25m/4hf7ysCBudTkv1N5SQCzgY/0b3546e2fdd77a6e72149a2eb26eb337/HOW-TO-VLOG-PDF.pdf
[4] 2022 (accessed Dec 12, 2022). A friendly guide to video production 2020. https://extension.colostate.edu/docs/comm/video-handbook2.pdf
[5] 2022 (accessed Dec 12, 2022). NCSL Video editing guideline.
[6] 2022 (accessed Dec 12, 2022). Remotion: Make videos programmatically in react. https://www.remotion.dev/
[7] 2022 (accessed Dec 12, 2022). Video editing and screencast guide best practices. https://www.depts.ttu.edu/english/student_involvement/labs_studios/media_lab/events/workshop_pdfs/vide-editing-guide.pdf
[8] 2022 (accessed Dec 12, 2022). Video production 101 - delivering the message. https://ptgmedia.pearsoncmg.com/images/9780321990198/samplepages/9780321990198.pdf
[9] 2022 (accessed Dec 12, 2022). Video production handbook fourth edition. https://digitalmindsphotography.files.wordpress.com/2020/05/video-production-handbook-fourth-edition.pdf
[10] 3PlayMedia. 2021. 3PlayMedia. https://www.3playmedia.com/
[11] Dustin Adams, Sri Kurniawan, Cynthia Herrera, Veronica Kang, and Natalie Friedman. 2016. Blind photographers and VizSnap: A long-term study. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 201–208.
[12] Dustin Adams, Lourdes Morales, and Sri Kurniawan. 2013. A qualitative study to support a blind photography mobile application. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. 1–8.
[13] Adobe. 2022 (accessed Dec 13, 2022). Premiere Pro. https://www.adobe.com/products/premiere.html
[14] Faisal Ahmed, Yevgen Borodin, Yury Puzis, and IV Ramakrishnan. 2012. Why read if you can skim: towards enabling faster screen reading. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*. 1–10.
[15] Apple. 2022 (accessed Dec 13, 2022). Final Cut Pro. https://www.apple.com/final-cut-pro/
[16] Mauro Avila, Francisco Kiss, Ismael Rodriguez, Albrecht Schmidt, and Tonja Machulla. 2018. Tactile sheets: using engraved paper overlays to facilitate access to a digital document's layout and logical structure. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. 165–169.
[17] Cynthia L Bennett, Jane E, Martez E Mott, Edward Cutrell, and Meredith Ringel Morris. 2018. How teens with visual impairments take, edit, and share photos on social media. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
[18] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for Placing Cuts and Transitions in Interview Video. *ACM Transactions on Graphics* 31, 4, Article 67 (jul 2012), 8 pages. https://doi.org/10.1145/2185520.2185563
[19] Erin L Brady, Yu Zhong, Meredith Ringel Morris, and Jeffrey P Bigham. 2013. Investigating the appropriateness of social network question asking as a resource for blind users. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1225–1236.
[20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[21] Minsuk Chang, Mina Huh, and Juho Kim. 2021. RubySlippers: Supporting Content-Based Voice Navigation for How-to Videos. 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 97, 14 pages. https://doi.org/10.1145/3411764.3445131
[22] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: Automatic Generation of Step-by-Step Mixed Media Tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/2380116.2380130
[23] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. 2013. DemoCut: Generating Concise Instructional Videos for Physical Demonstrations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 141–150. https://doi.org/10.1145/2501988.2502052
[24] Maitraye Das, Thomas Barlow McHugh, Anne Marie Piper, and Darren Gergle. 2022. Co11ab: Augmenting Accessibility in Synchronous Collaborative Writing for People with Vision Impairments. In *CHI Conference on Human Factors in Computing Systems*. 1–18.
[25] Descript. 2022 (accessed Sep 6, 2022). Descript. https://www.descript.com/
[26] C. Ailie Fraser, Joy O. Kim, Hijung Valentina Shin, Joel Brandt, and Mira Dontcheva. 2020. Temporal Segmentation of Creative Live Streams. 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, 1–12. https://doi.org/10.1145/3313831.3376437
[27] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh

Agrawala. 2019. Text-Based Editing of Talking-Head Video. *ACM Transactions on Graphics* 38, 4, Article 68 (jul 2019), 14 pages. https://doi.org/10.1145/3306346.3323028

[28] Rae Green. 2020. College Life...As A Blind Girl! https://www.youtube.com/watch?v=EFbU7g7dbp0

[29] Ijaz Ul Haq, Khan Muhammad, Tanveer Hussain, Soonil Kwon, Maleerat Sodanil, Sung Wook Baik, and Mi Young Lee. 2019. Movie scene segmentation using object detection and set theory. *International Journal of Distributed Sensor Networks* 15, 6 (2019), 1550147719845277.

[30] Rex Hartson and Pardha S Pyla. 2012. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.

[31] Karen Holtzblatt and Hugh Beyer. 1997. *Contextual design: defining customer-centered systems*. Elsevier.

[32] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2022. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. *arXiv preprint arXiv:2205.15868* (2022).

[33] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[34] Bernd Huber, Hijung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J. Mysore. 2019. B-Script: Transcript-Based B-Roll Video Editing with Recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300311

[35] Imvidu. 2022 (accessed Sep 6, 2022). Imvidu. https://imvidu.com/

[36] The Smith-Kettlewell Eye Research Institute. 2019. YouDescribe.com. https://youdescribe.org/.

[37] Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. *arXiv preprint arXiv:1808.10584* (2018).

[38] Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, and Robert C. Miller. 2014. Data-Driven Interaction Techniques for Improving Navigation of Educational Videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 563–572. https://doi.org/10.1145/2642918.2647389

[39] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 4017–4026. https://doi.org/10.1145/2556288.2556986

[40] Young-Ho Kim, Bongshin Lee, Arjun Srinivasan, and Eun Kyoung Choe. 2021. Data@ hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[41] Martin Kurze. 1996. TDraw: a computer-based tactile drawing tool for blind people. In *Proceedings of the second Annual ACM Conference on Assistive technologies*. 131–138.

[42] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-Driven Scenes. *ACM Transactions on Graphics* 36, 4, Article 130 (jul 2017), 14 pages. https://doi.org/10.1145/3072959.3073653

[43] Cheuk Yin Phipson Lee, Zhuohao Zhang, Jaylin Herskovitz, JooYoung Seo, and Anhong Guo. CHI 2022. CollabAlly: Accessible Collaboration Awareness in Document Editing. (CHI 2022).

[44] Franklin Mingzhe Li, Jamie Dorst, Peter Cederberg, and Patrick Carrington. 2021. Non-Visual Cooking: Exploring Practices and Challenges of Meal Preparation by People with Visual Impairments. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–11.

[45] Franklin Mingzhe Li, Franchesca Spektor, Meng Xia, Mina Huh, Peter Cederberg, Yuqi Gong, Kristen Shinohara, and Patrick Carrington. 2022. "It Feels Like Taking a Gamble": Exploring Perceptions, Practices, and Challenges of Using Makeup and Cosmetics for People with Visual Impairments. In *CHI Conference on Human Factors in Computing Systems*. 1–15.

[46] Jingyi Li, Son Kim, Joshua A Miele, Maneesh Agrawala, and Sean Follmer. 2019. Editing spatial layouts through tactile templates for people with visual impairments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[47] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086* (2022).

[48] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScape: Collaborative Concept Mapping for Video Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173961

[49] Xingyu Liu, Patrick Carrington, Xiang 'Anthony' Chen, and Amy Pavel. 2021. What Makes Videos Accessible to Blind and Visually Impaired People?. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–4.

[50] Xingyu Liu, Ruolin Wang, Dingzeyu Li, Xiang'Anthony' Chen, and Amy Pavel. UIST 2022. CrossA11y: Identifying Video Accessibility Issues via Cross-modal Grounding.

[51] Rosiana Natalie, Ebrima Jarjue, Hernisa Kacorri, and Kotaro Hara. 2020. ViScene: A Collaborative Authoring Tool for Scene Descriptions in Videos. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–4.

[52] Ashley Nemeth. 2016. How Blind Mom Cooks. https://www.youtube.com/watch?v=ZaQtx54N6iU

[53] Ashley Nemeth. 2020. Day In The Life Blind Mom. https://www.youtube.com/watch?v=YCF5LVGfGJE

[54] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[55] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2807442.2807502

[56] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: Video-Based Asynchronous Video Review. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 517–528. https://doi.org/10.1145/2984511.2984552

[57] 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

[58] Amy Pavel, Gabriel Reyes, and Jeffrey P. Bigham. 2020. Rescribe: Authoring and Automatically Editing Audio Descriptions. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 747–759. https://doi.org/10.1145/3379337.3415864

[59] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martinez, and Joaquín Fernández-Valdivia. 2000. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 3. IEEE, 314–317.

[60] Yi-Hao Peng, Jeffrey P Bigham, and Amy Pavel. 2021. Slidecho: Flexible Non-Visual Exploration of Presentation Videos. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–12.

[61] 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*. 1–12.

[62] Yi-Hao Peng, Jason Wu, Jeffrey P. Bigham, and Amy Pavel. 2022. Diffscriber: Describing Visual Design Changes to Support Mixed-ability Collaborative Presentation Authoring. In *Proceedings of the 35rd Annual ACM Symposium on User Interface Software and Technology* (Bend, Oregon, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA.

[63] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

[64] Reduct.Video. 2022 (accessed Sep 6, 2022). Reduct.Video. https://reduct.video/

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

[66] Abir Saha and Anne Marie Piper. 2020. Understanding audio production practices of people with vision impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–13.

[67] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

[68] José Francisco Saray Villamizar, Benoît Encelle, Yannick Prié, and Pierre-Antoine Champin. 2011. An adaptive videos enrichment system based on decision trees for people with sensory disabilities. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*. 1–4.

[69] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O Wobbrock. 2021. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–19.

[70] Woosuk Seo and Hyunggu Jung. 2021. Understanding the community of blind or visually impaired vloggers on YouTube. *Universal Access in the Information Society* 20, 1 (2021), 31–44.

[71] Lei Shi, Yuhang Zhao, Ricardo Gonzalez Penuela, Elizabeth Kupferstein, and Shiri Azenkot. 2020. Molder: an accessible design tool for tactile maps. In *Proceedings*

of the 2020 CHI Conference on Human Factors in Computing Systems. 1–14.

[72] Hijung Valentina Shin, Wilmot Li, and Frédo Durand. 2016. Dynamic Authoring of Audio with Linked Scripts. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 509–516. https://doi.org/10.1145/2984511.2984561

[73] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M. Drucker, and Ken Hinckley. 2020. InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376782

[74] Arjun Srinivasan, Bongshin Lee, and John T. Stasko. 2020. Interweaving Multimodal Interaction with Flexible Unit Visualizations for Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 0, 0 (2020), 15 pages pages. https://doi.org/10.1109/TVCG.2020.2978050 (Early access).

[75] Arjun Srinivasan and John Stasko. 2018. Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 511–521. https://doi.org/10.1109/TVCG.2017.2745219

[76] Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. 2016. Pano2Vid: Automatic Cinematography for Watching 360 Videos. In *Asian Conference on Computer Vision*. Springer, 154–171.

[77] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. QuickCut: An Interactive Tool for Editing Narrated Video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 497–507. https://doi.org/10.1145/2984511.2984569

[78] Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. 2021. Automatic Generation of Two-Level Hierarchical Tutorials from Instructional Makeup Videos. 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 108, 16 pages. https://doi.org/10.1145/3411764.3445721

[79] Lourdes M Morales Villaverde. 2014. Facilitating blind people to independently format their documents. *ACM SIGACCESS Accessibility and Computing* 108 (2014), 38–41.

[80] W3C Web Accessibility Initiative (WAI). 2022 (accessed Dec 12, 2022). Introduction to web accessibility. https://www.w3.org/WAI/fundamentals/accessibility-intro/

[81] Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu. CHI 2021. Toward Automatic Audio Description Generation for Accessible Videos.

[82] Saelyne Yang, Jisu Yim, Juho Kim, and Hijung Valentina Shin. 2022. CatchLive: Real-Time Summarization of Live Streams with Stream Content and Interaction Data. 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 500, 20 pages. https://doi.org/10.1145/3491102.3517461

[83] Kai Zhang, Lawrence H Kim, Yipeng Guo, and Sean Follmer. 2020. Automatic Generation of Spatial Tactile Effects by Analyzing Cross-modality Features of a Video. In *Symposium on Spatial User Interaction*. 1–10.

[84] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. 2022. Detecting Twenty-thousand Classes using Image-level Supervision. In *ECCV*.

A STUDY PARTICIPANTS DEMOGRAPHICS

Table 2: Study Participants (P1-P8: Formative study participants, P1, P4, P8-P17: Controlled study participants, P14, P18, P19: Exploratory study participants. Three participants marked with * participated in both formative study and controlled study (P1, P4, P8), and one participant with † participated in both controlled study and exploratory study (P14). All participants are screen reader users.)

PID	Age	Gender	Visual impairment	Onset	Video editing tool	Content type	Experience (yr.)
P1*	27	M	Legally blind	Acquired	Microsoft Photos	User interviews	4
P2	23	M	Totally blind	Acquired	VirutalDub 2	Sports videos, Product reviews	5
P3	22	F	Legally blind	Congenital	Final Cut Pro	Live streams, Presentations	1
P4*	35	M	Low vision	Acquired	Python & FFmpeg	Art demonstrations, Tutorials	7
P5	28	F	Legally blind	Congenital	iMovie	Video podcasts	11
P6	24	M	Low vision	Acquired	Final Cut Pro	Short-form videos	4
P7	41	M	Legally blind	Congenital	iMovie (mobile)	Vlogs	2
P8*	41	M	Legally blind	Acquired	Final Cut Pro	Short film	20
P9	40	F	Totally blind	Congenital	Microsoft Photos	Accessibility videos	1
P10	54	M	Totally blind	Congenital	Reaper	Podcasts, Music videos	1
P11	31	F	Legally blind	Acquired	Reaper	Fashion videos, Accessibility videos	8
P12	30	M	Totally blind	Congenital	Reaper	Twitch streams, Short-form videos	3
P13†	58	M	Legally blind	Acquired	Reaper	Workout videos, product reviews, Accessibility videos	2
P14	40	M	Legally blind	Congenital	Reaper	Tech demonstrations	1
P15	29	F	Totally blind	Acquired	Windows Movie Maker	Accessibility videos, Video editing tutorials	5
P16	31	M	Totally blind	Congenital	VideoReDo	Conference talks	6
P17	30	F	Totally blind	Acquired	Windows Movie Maker	Video podcasts, Short-form video	3
P18	63	M	Totally blind	Congenital	None	Planting tutorials	5
P19	29	F	Totally blind	Acquired	None	Vlogs, Video podcasts, Live streams	0

B EVALUATION STUDY VIDEO DATASET

Table 3: Videos used in the evaluation study.

Video ID	Title	Duration (Original)	Creator	URL
V0	College Life...As A Blind Girl!	3m 12s (9m 10s)	Rae Green	[28]
V1	How Blind Mom Cooks	11m 12s (20m 38s)	Ashley Nemeth	[52]
V2	Day In The Life Blind Mom	11m 5s (20m 16s)	Ashley Nemeth	[53]
V3	Growing With Blind Bryan: New border, Rampant Runner and a Juicy Peach Tree	10m 2s	P14	None
V4	An adventure to dinner: Demonstrating O&M techniques	12m 57s	P18	None
V5	Blind construction tools	9m 37s	P19	None

C FORMATIVE STUDY VIDEO DATASET

To collect videos demonstrating how BLV creators edit videos, we first searched YouTube for all combinations of a set of vision-related keywords (blind, low vision, visual impairment, screen reader), and a set of video editing keywords (editing videos, making videos, creating videos), following prior work [44, 45]. For each search phrase, we included all unique videos that had a title related to vision and video editing and stopped the search when the results of an entire search page were irrelevant. We then filtered out videos that did not cover video editing (1 filtered) or had poor audio or video quality (3 filtered). Our final dataset contained 24 videos (V1-V24) uploaded before October 12, 2021. The videos contained overviews of the video production process (2 videos) and tutorials of video editing software (22 videos). For the full list of videos, see Supplemental Material.