

# Fast-forwarding, Rewinding, and Path Exploration in Interactive Branched Video Streaming

Albin Vogel  
Linköping University  
Sweden

Erik Kronberg  
Linköping University  
Sweden

Niklas Carlsson  
Linköping University  
Sweden

## ABSTRACT

With interactive branched video, the storyline is typically determined by branch choices made by the user during playback. Despite putting users in control of their viewing experiences, prior work has not considered how to best help users that may want to quickly navigate, explore, or skip parts of the branched video. Such functionalities are important for both impatient users and those rewatching the video. To address this void, we present the design, implementation and evaluation of interface solutions that help users effectively navigate the video, and to identify and explore previously unviewed storylines. Our solutions work with large, general video structures and allow users to effectively forward/rewind the branched structures. Our user study demonstrates the added value of our novel designs, presents promising tradeoffs, provides insights into the pros/cons of different design alternatives, and highlights the features that best address specific tasks and design aspects.

## CCS CONCEPTS

• **Information systems** → **Multimedia streaming**; • **Human-centered computing** → **Interactive systems and tools**.

## KEYWORDS

Branched video, Non-linear video, Interactive film, Progress bar, Fast-forwarding, Rewinding, Path exploration, User study

### ACM Reference Format:

Albin Vogel, Erik Kronberg, and Niklas Carlsson. 2021. Fast-forwarding, Rewinding, and Path Exploration in Interactive Branched Video Streaming. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21)*, October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3474085.3475308>

## 1 INTRODUCTION

*Branched videos streaming* [14, 17, 37] (sometimes also called *non-linear*, *hypervideo*, and *multi-path* video streaming [4, 13, 22–24, 27, 29, 38]) is a promising approach to attract the attention of future generations of movie lovers. With this type of video, the storyline is typically determined by the branch choices made by the user during playback. This provides a personalized, interactive experience not offered by traditional video streaming services and meets the end-users’ expectation of increasingly interactive service.

While interactive branched video is not a new idea, only recently have big production houses (e.g., BBC, Netflix) started to produce and stream such content. In 2019, Netflix won several awards and achieved great success with its Black Mirror episode “Bandersnatch” [37]. Since then, Netflix has actively produced and streamed more branched content. While this may be the first steps toward becoming mainstream, *many research problems remain open concerning how to best aid viewers of branched video*.

Clearly, branched video offers content creators and production companies a flexible tool for innovative content creation, and users can be provided a highly personalized, interactive service. However, perhaps equally importantly, branched content allows users to rewatch a video from different perspectives, with alternative plots, and provides content creators a way to give access to extended content. Despite these important use cases, thus far, *users rewatching branched content have mostly been forgotten in the interface design*.

While the media itself may help engage users, we note that *users by nature are impatient* and are unlikely to want to rewatch exactly the same branch segment of a video over and over again. It is therefore important to provide user-friendly tools that allow users to *effectively navigate and explore* the branched video content during both first-time viewing and subsequent viewings. For example, a first-time viewer may want to redo branch choices (e.g., if the outcome is unwanted) or fast-forward a movie when they do not have the time or patience to watch the movie in full. Alternately, a user rewatching a movie may want to find new plot sequences and explore different storylines and avoid rewatching already viewed branch segments. However, despite the playback bar’s central role in the interface design of linear video players, no prior work incorporates and evaluates playback bars and similar elements for users rewatching branched video.

In this paper, we present the design, implementation, and evaluation of interface designs and features that help the user to *effectively navigate* branched videos during first-time viewing and to *effectively identify and explore* storylines of the video that they have not yet viewed. The candidate designs of consideration are implemented as a set of new (optional) features within the branched video player framework originally developed by Lindskog et al. [17]. This allows us (and our user-study participants) to quickly test and switch between different optional designs and features. The extended player is open source and are made available with this publication [34]. Most importantly, the paper addresses two important aspects not addressed by prior work: (i) scenarios in which branched-video users rewatch or otherwise want to fast-forward/rewind through a video structure, and (ii) scalable playback-bar designs that provide a partial view of the branch structure around the current play point together with additional information about past/future branch choices. All implemented and tested features are new.

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MM '21, October 20–24, 2021, Virtual Event, China

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8651-7/21/10.

<https://doi.org/10.1145/3474085.3475308>

The set of designs and features implemented include novel playback-bar designs and fast-forward/rewind functions that, together with other implemented features, help the user keep track of aspects such as the current play point (e.g., depth of current branch, point on current branch, buffer conditions, upcoming branches), and the extent to which paths have been explored. In addition to implementing and evaluating new functionalities, compared to Lindskog et al. [17], our designs are also made scalable in that they provide the user with an easily understood, localized (partial) view of the branch structure around the branch being played. This ensures that our designs can be used on branched video structures of any size.

To evaluate the proposed designs and provide insights into the best design tradeoffs for the above context, we performed a three-part user study. First, participants were asked to complete three specific navigation/exploration tasks using three different example designs that were selected to capture key design tradeoffs such as how much of the branch structure to show and how to best obtain user input. During each experiment, we carefully measured and logged the user’s activity, actions, and other events. At the different stages of these tests, the participants answered ease-of-use questions (e.g., SEQ) for each task and cognitive load questions (e.g., NASA-TLX) for each design. They also provided opinion scores about different design aspects. Second, we performed head-to-head comparison of the designs and specific design elements. Here, we combined relative rankings (along several dimensions) with participants picking a “winner” when performing different tasks. Finally, in the third part, we studied the features that best address specific aspects. For each set of questions, we used a combination of feature-by-feature evaluation (e.g., Likert questions) and head-to-head comparisons (e.g., participants ranking features and/or picking a winner for different tasks + dimensions).

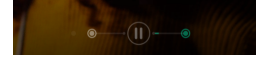
The user study highlights the added value of our novel designs, including two scalable playback-bar designs that provide a *partial* and *extended partial* view of the branch structure, respectively, that focus on the current portion of the tree and provide natural transitions from one branch to the next. The user tests and design comparisons show that our playback-bar solutions provide significant help, come at small additional cognitive load, and are well-liked. The playback-bar features are most liked when rewatching the video or when navigation/exploration tasks become more complex. In these cases, the participants preferred to be able to see more of the branch structure and take longer branch jumps. Similar observations are made for other evaluated features (e.g., forward/rewind functionalities, annotated branch buttons, and other features providing information about previously watched paths).

Throughout the paper we also provide design insights and share lessons learned (e.g., through preliminary think-aloud user tests), including how to best implement various design aspects (e.g., whether to pause playback or not when reaching a branch point) for users using forwarding/rewinding (e.g., when rewatching a video) compared to a first-time viewer (case studied by Lindskog et al [17]). Our findings highlight significant differences in how branched video viewing differs from regular videos viewing with respect to first-time viewing versus later viewings of a video.

**Outline:** Section 2 presents a brief background and related work. The subsequent two sections present our system design (Section 3) and user study (Section 4). Section 5 concludes the paper.



(a) Netflix’s *Puss in Boots*



(b) Eko’s video *Epic Night*

**Figure 1: Partial screenshots illustrating customized solutions used in two example videos by Netflix and Eko.**

## 2 BACKGROUND AND RELATED WORK

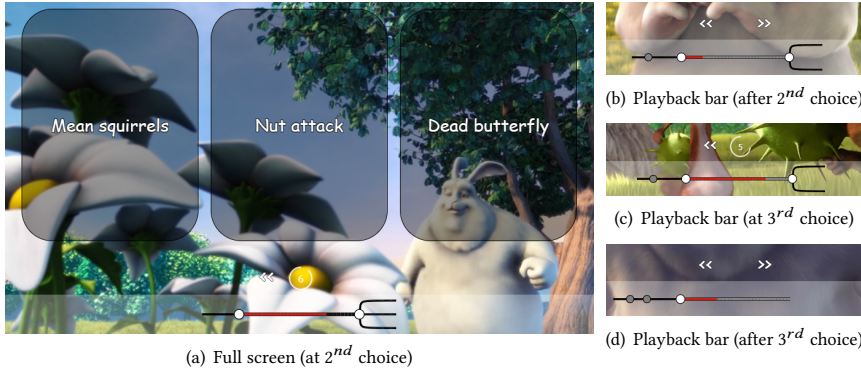
**Commercial examples:** Most interactive videos are either custom-made by big production houses (e.g., Netflix, BBC), custom-created using services such as Eko [7], or innovatively created using special features intended for other purposes (e.g., YouTube annotations). Common to most such videos is that the user periodically is asked to choose between alternate paths (typically expressed as multiple-choice questions). However, thus far, user interfaces are typically highly customized from case-to-case and there is a lack of generic tools to effectively navigate the video structure quickly (e.g., fast-forward or rewind) in ways analogous to what we are accustomed to in the traditional *linear* streaming context. We argue that generally applicable navigation/exploration functionalities that allow users to skip forward/backwards in the video are important for improving user engagement in the branched context as well.

While we provide the first generic playback-bar interface with such functionalities, we are not the first to allow users to jump to specific branch points. For example, in the Netflix video “Puss in Boots: Trapped in an Epic Tale” (Figure 1(a)), the user’s past choices are shown at the bottom of the screen (as pages in a book), providing an easy way to redo past choices. While intuitive, this interface is movie-specific and may not generalize well to other movies. Another example is Eko’s video “Epic Night” (Figure 1(b)), which allows users to fast-forward/rewind to the next/previous branch point by clicking on the start/end point of the current branch.

**Related work:** Various forms of interactive media allowing users to switch between videos have been around for 20+ years [29]. Since then, the multi-video concept has been generalized to 360° videos [27], different annotation and link types have been explored [22, 24], and videos have been integrated with game engines [35].

The closest is the work by Lindskog et al. [17], who present a generalized playback bar and the results from a user study. In this paper, we extend their player to allow users to rewind, fast forward, and in other ways more efficiently explore the video. These types of features were not considered by Lindskog et al. In contrast to their work, we focus on the context of users rewatching a video and provide playback-bar features that scale to large video structures.

Other related work includes implementations and testing of non-linear video player solutions for desktop and mobile units [22, 24]. These and other similar systems are often designed to illustrate some annotation and editing principles for linking videos; however, usability is seldom considered [19]. Additional orthogonal challenges include content creation [21, 23, 30], prefetching and cache management problems [18, 20], and optimization of the playback quality of clients using multicast [4, 38] or HTTP-based Adaptive Streaming [14]. Others have designed tag-based systems that automatically stitch together multiple videos into a single playback sequence [11] or prefetching frameworks for related contexts where users may quickly switch between videos [3, 15, 32, 36].



**Figure 2: Full screen view of example player, including partial playback bar, branch buttons, and fast-forward/rewind buttons.**

While there is a lack of work on fast-forward/rewind functionalities in the context of branched streaming, related works in the context of (traditional) linear video exist [6, 8, 31]. For example, Tan et al. [31] consider fast-forward and reverse playback using transcoding. It has also been shown that annotating the timeline of the playback bar (of linear players) with buffer state and bookmarks of prefetched segments can improve users’ playback experiences [2].

Using a playback bar to navigate a branch structure is also implicitly related to how best to fit large structures onto a small screen [10, 28] and topology-aware navigation [26]. Navigation among branch points can also be seen as a form of landmark navigation [25, 33]. Other inspirational, orthogonal works concerns direct manipulation [5, 12], mimic actions [16], and dynamic graph exploration [1]. While these works inspire, most inspiration came from the commercial examples above (e.g., keeping track of past branch points as a sequence of past choices and allowing forward-ing/rewarding to the next/past branch point using simple buttons).

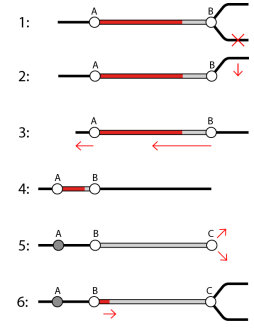
### 3 SYSTEM DESIGNS AND FEATURES

#### 3.1 High-level design

We implemented the candidate designs as a set of optional features within the framework by Lindsog et al. [17]. This allows us to easily switch between designs that we want to test or demonstrate. Our novel designs include branched playback-bar designs, annotated branch buttons, and other features that show the current play point, the previously explored path, and the extent that paths have been explored. We also extend the design to provide multiple means to fast-forward, rewind, and jump within the branch structure.

Figure 2(a) shows a basic example design (design B in Section 4.1). We use it to illustrate the three main components of our designs.

- **Playback bar with partial view:** In contrast to the branched playback bar implemented by Lindsog et al. [17], who in default mode shows the full tree structure, our default designs are made scalable in that our solutions provide a local (partial) view of the branch structure around the branch being played, ensuring that the designs can be used on branched video structures of any size. Our designs also include seamless branch transitions (e.g., Figure 3).
- **Branch-selection buttons with countdown counters:** In addition to the default buttons shown in Figure 2(a), we have



**Figure 3: Step-by-step breakdown of a branch transition.**

also implemented and evaluated versions that use annotations and colors to provide the user with different levels of information about what paths they have taken previously.

- **Fast-forward and rewind buttons:** To jump forward or backwards in the video, the user can either (i) use our fast-forward or rewind buttons, or (ii) click on branch points shown in the playback bar (optional feature). When branch buttons are shown (and the user is asked to make a choice), the forward option is not displayed (e.g., Figures 2(a), 2(c)), whereas during regular playback it is (e.g., Figures 2(b), 2(d)). Buttons are highlighted when the user hovers over them.

We have implemented these and several alternate designs and candidate features associate with each of these three main components.

**Scalability:** To ensure that they are scalable and applicable to any branch structure, our designs do not limit the number of branch points, edges, or relative play durations that can be used. To facilitate the use of large branch structures (not possible to visualize clearly in full on a playback bar), all presented and evaluated interface designs/features display only the branches one (or two) levels ahead and provide compact information about past and future choices. While our focus is presentation-oriented, the lookup of upcoming branches (constant-time lookups) and calculations of explored paths (state counters) are also highly scalable. We also allow re-use of play segments across branches. As long as the meta information fits in memory, arbitrarily sized graph structures can be used based on any underlying video(s). As of now, the metadata size is proportional to the number of branches, with the metadata of each branch typically below 100 bytes (text labels + branch IDs).

**Design challenges:** We next describe the main design challenges associated with each component. For each challenge, we describe key insights and/or the features that we implemented and tested to better understand the tradeoffs of that challenge.

#### 3.2 Branched playback bar

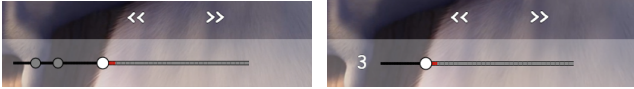
Several important questions must be addressed in the design.

**How to display the branch structure:** Clearly, it is not feasible to display the full branch structure when the structure is large (e.g., when the user is asked to make many path choices). We therefore created two novel playback bars that *provide a local (partial) view of the branches around the branch being played*. Figures 2(a) and 4 show example modes of these two playback bars. The first design only





**Figure 4: Playback bar with extended partial view and the multi-hop jumping feature (a) turned off and (b) turned on. Users can branch jump to any white (clickable) branch point.**



**Figure 5: Depth shown using (a) points and (b) numbers.**



**Figure 6: Playback bar showing full tree structure with clickable branch points (in white) and highlighting (in blue) when hovering over such branch point.**



**Figure 7: Playback bar showing the ratio  $n/N$  of the number of traversed branches ( $n$ ) and the total possible paths ( $N$ )**

shows the upcoming branch options, whereas the second design also shows one level beyond that. This ensures that the designs can be used on branched structures of any depth.

One challenge when designing a partial playback bar is *how to seamlessly transition from one branch to the next*. Figure 3 presents a step-by-step example of how we implement branch switching for the first of the two playback-bar designs. In the example, we transition from a two-path branch to another two-path branch. After the user makes the initial branch choice, we (1) remove the non-selected branch, (2) straighten out the upcoming branch, (3+4) re-scale the time axis, (5) add the old branch point to the sequence of prior visited branch points, transition to the next branch, and (6) display the next set of branch options at the end of this branch.

We have also found it valuable to *provide depth information* about the current play point. While we allow the depth (i.e., the number of branch points from the start of the video) to be illustrated using either circles (e.g., Figure 5(a)) or numbers (e.g., Figure 5(b)), we have found the first option (our default) desirable in most cases.

**How far to allow users to jump:** For each design, we have an optional feature that allows users to jump to any displayed branch point. Figures 4(a) and 4(b) provide examples of when this feature is turned off and on, respectively. Here, white branch points are clickable, are highlighted (in blue) when the mouse hovers over them, and the movie “jumps” to the selected branch point whenever the branch point is clicked. With the two partial playback bars (Figures 2 and 4), we limit forwarding to one or two steps, respectively, but allow the user to jump backward to any previously visited branch point shown. We also implemented these features for Lindskog et al.’s [17] design that shows the full tree (Figure 6).

### 3.3 Branch-selection buttons and pacing

Branch-selection buttons inform the user of the next branch options. Rather than trying to optimize the design of such buttons, we focus our discussion on three previously unanswered questions.

**How to best augment branch buttons to inform users about the degree that paths have been explored:** To help a user understand what paths they have explored thus far, and under which branch options there may still be unexplored paths, we maintain information about which end-to-end paths have been fully explored and share part of this information with the user. Here, we focus primarily on simple and relatively discrete design options:

- No information. This is the default case that we *always* use the first time the viewer watches the video (or branch). This ensures minimal information leakage about future choices.
- Colored buttons. The boundary around buttons is colored green (no paths explored), yellow (some paths explored), or red (all paths explored).
- Text-based. Buttons are annotated, stating “Fully explored” or “Partially explored”, when applicable.
- Underlining. Choices are underlined (fully explored), dotted (partially explored), or not underlined at all (none explored).
- Ratio. Buttons are annotated with the  $n/N$  ratio of the number of end-to-end paths explored thus far ( $n$ ) and the total possible number of such paths ( $N$ ). This option provides the most information about future branches.

While the first option (i.e., no information) has been found desirable for first-time viewing (context of Lindskog et al. [17]), the later options may see increasing value the more times a user rewatches a video. These features are evaluated and compared in Section 4.3.

**Is such information best communicated by annotating the branch buttons or the playback bar:** To answer this question, we also implemented and experimented with solutions that provided the  $n/N$  ratio information on the playback bar itself (Figure 7).

It is often good to hide the playback bar. In our experiments, we implement and use a feature that only shows the playback bar when the user moves the mouse (+ stays up  $X$  seconds after the user stops moving the mouse) or when a branch point is reached (at which time both the branch-selection buttons and playback bar are displayed). The advantage of this is that the information (e.g., above  $n/N$  ratio) is mostly hidden from non-active users, but can easily be identified by users actively using the playback bar.

**How to best help users make timely branch choices:** We have found that *countdown counters are highly appreciated*. These (seen in Figures 2(a) and 2(c)) inform users about the time (in seconds) within which they are expected to make a path selection. In default setting, we start the timer 10 seconds before the deadline. A harder and more subtle design choice is whether a default path should be selected on behalf of the user when no choice has been made when the counter reaches zero. For the context of Lindskog et al. [17], the answer to this question was clearly yes for most users. However, this is not the case for our context (first learned from think-aloud testing with four participants). While users still like this functionality when reaching the branch point in regular playing mode, they prefer playback to be paused when reaching the branch point using fast-forward or rewind functionalities. One reason for this difference is that using such functionalities indicates that the user is currently making active choices. For such user, an automated choice can be highly disruptive and cause frustration. We therefore implemented and used a *case-based pause decision* determined by how the user reaches a branch point.

## 4 USER STUDY

**Overview:** The user study consists of three main parts: users tests with three example designs (Section 4.1), design comparisons (Section 4.2), and feature-by-feature comparisons (Section 4.3).

**Participants:** Before starting the study, we informed the participants about the study, asked them to fill out a consent form, and collected background information. We recruited 24 participants (7 females, 17 males) from a local university using emails and in-person contacts; all gave written consent to participate in the study. The participants were all university students from a mix of majors, aged 21-31 ( $\mu=23.9$ ,  $\sigma=2.1$ ), and had varied but mostly limited prior exposure to branched video (4 had never watched branched video, 13 had watched 1-to-2 times, 5 had watched 2-5 times, and only 2 had watched >5 times). The study took approximately 45 minutes and participants received chocolate bars for their participation.

Our participant sample provides a relatively well-defined group (university students in the typical age-range of 20–30 years old) and the majority (12 of 18 who answered) listed Bandersnatch as the first branch video that they had watched. These two aspects may bias the results and influence the participants' expectations, but it also helps provide a relatively clear reference point (e.g., compared to a broader selection of participants).

**Setup:** Like Lindskog et al. [17], we hand-crafted a set of branched videos using *Big Buck Bunny* [9] as the underlying linear video. The storylines differ by the branch segments and branch points defined in their respective metafiles. In all experiments, we used H.264 (MPEG-4 AVC), 30 frames/second, and 1920×1080 resolution. The user study was performed on a laptop (HP Pavilion) with a 14-inch, non-touch, 1920×1080 screen, Intel Core processor (i5-7200U), running Windows 10. The same two test administrators administered all user tests. For each session, the player was setup to use the full screen, ensuring that all users received the same experience with each design. Questions were provided in writing, leading questions were avoided, and the administrators sought to clarify any (rare) questions without influencing the participants' answers.

**Familiarizing the user with the player and branched video:** Before testing our design functionalities, we first let the user watch a branched video with all the developed features turned off. Here, only the default branch buttons were used to navigate the video. To obtain a baseline reference point, we also asked the participants to fill out a NASA-TLX form at the end of this test.

### 4.1 Part 1: User tests with 3 example designs

We first performed user tests with three example designs, representing designs with vastly different navigation possibilities. Designs, tasks, and questions were selected to allow us to derive insights into key design tradeoffs when navigating or exploring a video.

**Designs and tasks:** The three considered designs were.

- **A. Only ff/rw buttons (no playback bar):** This is perhaps the simplest design to allow fast-forward (ff) and rewind (rw) functionality. For this design, we simply disabled and hid all playback-bar functionalities.
- **B. Playback bar with partial view + ff/rw buttons:** Here, we used the example design shown in Figure 2, but with branch jumping using the playback bar turned off. With this design, the user can visualize the tree structure using

the playback bar but must use the ff/rw buttons to jump forward/backward to the next/previous branch point.

- **C. Only playback bar (no ff/rw buttons) but with the extended partial view and multi-hop branch jumping:** Here, we disabled ff/rw buttons, but instead enhanced the playback bar to use the extended partial view (Figure 4(b)) and allow backward branch jumping to any previously visited branch point and forward branch jumping to any branch point up to two steps forward.

For each design, the participants were asked to solve three *navigation* and *exploration* tasks, each focusing on a different aspect.

- **Task 1 [First-time use]:** *How well were interfaces perceived by first-time users (e.g., with regard to ease of use, intuitiveness, and complexity)?* To capture these aspects, this task was made open-ended. Here, we asked the participants to explore the features of the design, while playing the video as they desired for one minute. Furthermore, each new design was introduced using minimal instructions and only explained between tasks 1 and 2. In general, the interfaces were found relatively easy to use. Task 1 therefore also worked as an implicit training phase (for which we collected user data).
- **Task 2 [Find specific path]:** *How well do designs help users navigate the branched video?* Here, the users were asked to use the design features to navigate to a named branch as quickly as possible in not more than two minutes.
- **Task 3 [Explore many/all paths]:** *How well do designs help users identify and explore storylines not yet viewed?* In this task, the users were asked to explore as many different endings as possible in two minutes.

The above tasks were designed to account for a limited time budget per participant and lack of multiple comparable production-quality movies (needed for fair comparison). For these reasons, we used tasks that engaged the user to actively solve a task using the interface. The use of high interactivity tasks also exposes the user to more branch decisions, helps the user see pros/cons of the designs more quickly, and hence better compare how well they like the designs. Of course, production movies may be longer, with more elaborate plots, and users may make fewer selections.

**Reducing order effects:** For fair comparison of the designs, we determined the order in which each participant test the three designs according to a balanced Latin square (6 possible combinations), rotated the underlying tree structure used in each experiment around selected branch points (chosen by us), and informed the participants that the video structure changed between each design. Rotating the video structures this way ensured that (i) the participants did not know where a certain path was, and (ii) the expected playback duration is the same for each experiment. In general, the average completion times of task 2 improved each round (78s, 70s, 59s), whereas they leveled off for task 3 (110s, 98s, 98s).

**Measurements, questions, and opinion score:** During each experiment, we carefully measured and logged the users' activity (e.g., mouse movement), actions (e.g., clicks, interactions with buttons, playback-bar features, branch selections), and other events (e.g., when the video was started and when branch points were reached). Finally, the participant answered ease-of-use questions (e.g., SEQ) for each task, cognitive load questions (e.g., NASA-TLX)

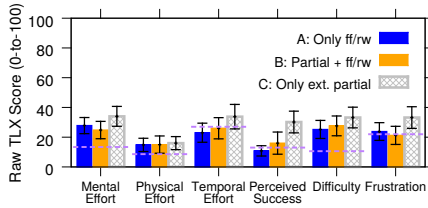


Figure 8: NASA-TLX effort scores.

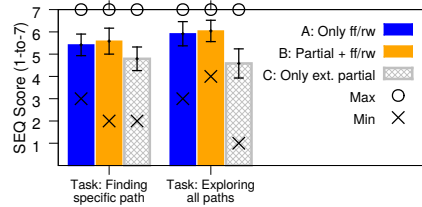


Figure 9: SEQ scores.

for each design, and provided opinion scores (on Likert scale) regarding different aspects of each design. We next present the results.

**Cognitive load and ease of use:** An ideal design should allow for fast task completions without adding any extra perceived effort.

Figure 8 summarizes the perceived workload scores (using NASA-TLX) of each design and Figure 9 summarizes the Single Ease Question (SEQ) results broken down per task and design. With NASA-TLX we use the raw scores (0-100 scale; low-to-high) and with SEQ we keep its default 1-to-7 scale (1 and 7 corresponding to “very difficult” and “very easy”, respectively). For both figures we include two-sided 95% confidence intervals for the average. For the SEQ we also include the minimum and maximum scores. To put the workload efforts to complete task 3 (“finding all paths”) into perspective, Figure 8 includes a (purple dotted) reference line for when watching the reference video at the start of the study. (Here, the task was to watch the video from start to end using only the basic branch selection buttons.) Comparing the winning design along each scale and this reference point, we only observed statistically significant workload increases for mental effort, physical effort, and difficulty.

In general, the perceived workloads are low across the six NASA-TLX scales, ease of use (SEQ) scores are high for all tasks, and the differences between the first two designs (A and B; both with ff/rw buttons) are non-significant (tested using t-test for paired samples with 95% confidence). The biggest differences are observed with design C (without ff/rw buttons). This design has noticeably higher workload NASA-TLX scores and lower SEQ values. Most noticeable are the worse perceived success (higher NASA-TLX values) and the SEQ of task 3 (i.e., “explore all paths”). Here, some participants did not feel as successful (or satisfied) with their own performance, even when the task was completed faster than with another design. These results demonstrate the value of allowing users to use ff/rw buttons, but also that the use of a partial playback bar comes at no additional cognitive workload or reduction in ease of use.

**Initial opinion scores:** To glean some initial understanding of (i) whether the users *liked* the different designs and (ii) whether they found that the designs *helped* solve the tasks, we posted Likert statements (i.e., “I like this design” and “The design helps solve the tasks”) and asked to what degree they agreed with the statements (1 and 5 corresponding to “strongly disagree” and “strongly agree”, respectively). Table 1 shows the average Likert scores for these two questions with 95% intervals. We note that design B (playback bar + ff/rw buttons) achieved significantly higher scores (and the smallest variations) for both questions. Using dependent t-test for paired samples we determined the *p*-values for these pairwise comparisons as 0.031 and 0.0059, respectively, when compared with design A (only ff/rw buttons) and 0.032 and 0.0022, respectively, when compared with design C (only ext. partial). All these pairwise

Table 1: How liked and useful (Likert scale 1-to-5) the designs were considered. Designs: A (only ff/rw), B (partial + ff/rw), C (only ext. partial).

	A	B	C
Like design	3.33 (.54)	3.83 (.38)	3.25 (.41)
Design helps	3.46 (.56)	4.38 (.32)	3.50 (.47)

differences are significant with 95% confidence. This shows that design B is both liked and helpful, suggesting that the playback bar and ff/rw buttons complement each other.

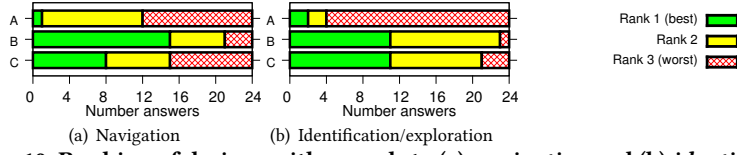
The lower liking of design C (without ff/rw buttons) is perhaps best understood by an extra Likert question asked only for that design: “I find it intuitive to use the branch points to jump”. While the majority agreed (eight 3s, eight 4s) or were neutral (three 5s), six disagreed (one 1 and five 2s) with this statement.

In summary, the above results are encouraging and show that the use of the playback bar (i) comes at low cognitive overhead, (ii) does not complicate the completion of the task, and (iii) design B (playback bar + ff/rw buttons) is both well liked and helpful.

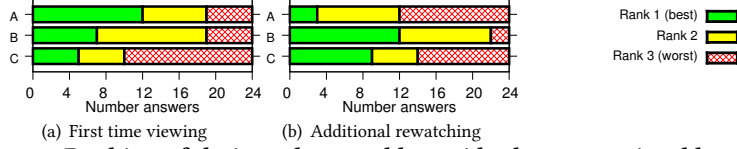
**Measured task performance:** We next looked closer at the participants’ *actual performance completing the tasks*. Consistent with their self-evaluation, we observed significant differences when comparing the designs with ff/rw buttons (i.e., designs A and B) and without (design C), as many users struggled to complete task 3 (“explore all paths”) using design C. While 20 out of 24 participants completed task 2 (“find specific path”) within two minutes using design C, only eight completed task 3 within two minutes. The corresponding numbers with design A were 22 and 19, and with design B, 20 and 18. As a reference point, it would take a user not allowed to ff/rw a minimum of 304 seconds to play all branches of our example structure once (assuming no stalls) and 448 seconds if playing all possible paths from start to finish. Compared to these baselines, all three designs provide significant improvements.

In general, we observed negligible differences between designs A and B. For task 2, participants completed the task 2.1% faster (62.1 vs. 63.4 seconds on average) and visited 5.4% less branches with design A than with design B. In contrast, for task 3, the participants were 7.6% faster (92.6 vs. 85.6 seconds) and visited 1.2% fewer branches with design B than with design A. In all four comparisons, the differences were non-significant at the 95% confidence level. (*p*-values using dependent t-test for paired samples: 0.45, 0.29, 0.15, 0.33.) These statistics and observations are consistent with the users’ self-evaluations (e.g., NASA-TLX analysis above).

The differences remained consistent also when considering the number of ff/rw actions used to complete the tasks. For example, to complete task 3, the participants forwarded 11.0, 11.8, and 5.8 times on average with the respective three designs A, B, and C. The number of forward jumps nicely match the total number of branches (11) in the full branch structure used here (as exemplified in Figure 6). The number of forward actions for task 2 where fewer by about half (6.6, 6.6, 3.5). The fewer ff/rw jumps when using design C help explain its lower task completion rate and demonstrate its longer learning curve. (A few participants completed task 3 quickly with design C, but for most task 3 took longer with design C.)



**Figure 10: Ranking of designs with regards to (a) navigation and (b) identification/exploration. Labels: A (only ff/rw), B (partial + ff/rw), C (only ext. partial).**



**Figure 11: Ranking of designs that would provide the most enjoyable video experience. Labels: A (only ff/rw), B (partial + ff/rw), and C (only ext. partial).**

**Table 2: Look + feel of branch switching design. (Likert: 1-5.)**

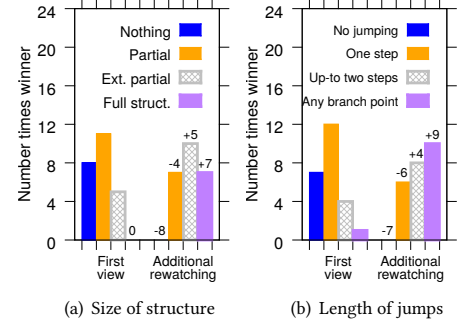
Likert statement	B: Partial + ff/bw	C: Only ext. part.
I like how the playback bar switches from one branch to another	3.92 (0.36)	3.38 (0.53)
Switching from one branch to another is too fast	1.96 (0.43)	1.88 (0.28)
Switching from one branch to another is too slow	2.46 (0.52)	2.67 (0.50)

**Branch switching implementation:** Table 2 presents the results from playback-bar specific *design questions* asked after using designs B and C. In general, the users liked the branch switching implementation (high agreement with first question) and found the branch switching timing adequate (low agreement with the last two questions). These findings are important since they show that we can successfully implement a playback bar that shows only a limited view of the full branch structure (which in practice can be very large) in an intuitive and easy-to-use manner. Finally, we note that the simpler design (B) again is more appreciated than the playback bar providing an extended partial view (design C).

## 4.2 Part 2: Head-to-head design comparisons

We next present results based on head-to-head comparisons that took place after all three designs had been considered individually. Before asking our comparison questions, we ensured that the users had understood the main difference between the interfaces evaluated in the previous part of the study.

**Rank-based analysis:** First, we asked the users to rank the three designs from best [1] to worst [3] with regards to how well different designs help the user (i) *navigate the branched video* and (ii) *identify and explore* storylines of the branched video that they have not already viewed. As shown in Figure 10, the designs with a playback bar (designs B and C) had the most top choices (rank 1) and the design without a playback bar (design A) only obtained 1/24 and 2/24 first place votes, respectively. This shows that the participants found value in the playback bar. For *navigation*, design B was the clear winner, whereas for *identifying and exploring* new paths there was a more even split between the two designs with a playback bar. For this type of task, users found value in the added information provided by design C, but the design still lost votes due to its lack of the well-liked ff/rw buttons.



**Figure 12: Winning designs for first-time viewing and additional rewatching. Shift in votes shown using annotation.**

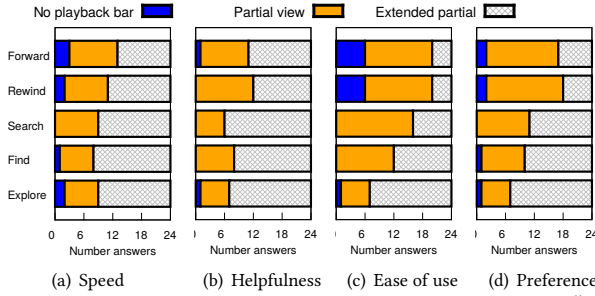
The above results show that more advanced playback-bar designs (which may help solve more complex tasks) may be preferred when rewatching a video, but not necessarily when watching a video for the first time. To validate this hypothesis, we next look at how the users ranked the designs based on which design they thought would give them the most enjoyable video experience (i) *when watching a branched video for the first time* and (ii) *when rewatching a branched video more times*. These results, shown in Figure 11, clearly validate our hypothesis. For first time use, half of the users (12/24) picked design A (without playback bar) whereas only 3/24 picked this design for rewatching the video. For this case, almost all users preferred to use a playback bar: 12 picked design B and 9 picked design C. Furthermore, half of the users picked design A as their least desirable design when rewatching the video (rank 3).

**How much of the branch structure should be shown:** To reduce biases introduced by combining features (e.g., lack of ff/rw buttons in design C), we next asked participants how much of the tree they would like displayed *regardless of other features being enabled/disabled* for the cases of (i) first-time viewing and (ii) video rewatching. Given this formulation, the desire to see (at least) part of the tree became even greater. Figure 12(a) shows these results. Here, the rewatching results (right) are annotated with the relative change (+/- notation) compared with first-time viewing (left). We again note a clear shift towards larger structures when rewatching the video, including some votes for the full structure (e.g., Figure 6).

**How to jump and how far to jump:** Also here, we observe significant differences between first-time viewing and rewatching. While 20 out of 24 preferred to use the ff/rw buttons over clicking on branch points displayed on the playback bar for first-time viewing, both options received the same number of votes when rewatching. Furthermore, as shown in Figure 12(b), users prefer much longer jumps when rewatching the video.

**Task-based winner selection:** The increased preference for using the extended view (if still allowed ff/rw buttons) became even clearer when users were asked which playback-bar design they would want when solving five different tasks: (i) “fast forwarding”, (ii) “rewinding”, (iii) “tree exploration and searching”, (iv) “finding new paths to explore”, and (v) “exploring all possible paths”. Figure 13 shows these results. For each of the tasks, we asked the participants to pick a “winner” from *no playback bar* (blue), *playback bar with partial view* (orange), and *playback bar with extended partial*





**Figure 13: Winning design for five different tasks (“fast forwarding”, “rewinding”, “tree exploration and searching”, “finding new paths to explore”, and “exploring all possible paths”) for four criteria.**

view (gray) with regard to four criteria: (a) “speed”, (b) “helpfulness”, (c) “ease of use”, and (d) “overall personal preference/choice”. As the tasks become more complex (top-to-bottom in the figure) we note a clear shift toward a desire to use the extended partial view (gray). We also note that very few participants picked “no playback bar” (blue) for these tasks. This again suggests that the participants gained an appreciation for the features through their very brief initial interaction with the designs.

### 4.3 Part 3: Feature-by-feature comparisons

Finally, we evaluated each feature individually (Likert) and asked participants to pick a “winner” for different tasks/criteria. Here, we focus on aspects not addressed earlier in the study.

**How to best show depth information:** We observed a statistically significant preference using points (e.g., Figure 5(a)) rather than with numbers (e.g., Figure 5(b)) to show depth information. For example, the average Likert scores of the two features were 3.79(0.52) and 2.42(0.50), respectively, 19 out of 24 users preferred the point-based feature when selecting a “winner” ( $p$ -value of 0.0033 using binomial hypothesis testing), and a dependent  $t$ -test for paired samples (on Likert scores) with  $p$ -value of 0.0007.

**How to best inform users about previously taken paths along each branch:** Referring to the branch annotation features outlined in Section 3.3, most users agreed that features showing the exact ratio  $n/N$  of paths explored thus far are good for their intended usage. For example, the two features showing the ratio on the buttons or playback bar, respectively, were the most frequent “winners” in 17 (11+6) and 22 (14+8) cases (out of 24) when picking the “winner” for helping the user to (i) “navigate the branched video” and (ii) “identify and explore storylines of the branched video that it has not already viewed”, respectively.

However, many users would still rather use the “color buttons”. For example, almost half of the users (11 out of 24) still preferred the “colored” buttons as “feature that they would prefer to use themselves”. (The two “ratio” features were picked in 7+3 of the cases.) These differences can also be observed when looking at the Likert scores of the features. Here, the “colored buttons” and “ratio buttons” scored 3.25 (0.63) and 3.63 (0.53), respectively. In contrast, the other button-based features (“text”, “underlining”) and the feature that shows the exact ratio on the playback bar obtained much lower average scores: 2.46 (0.54), 2.29 (0.48), and 2.96 (0.52).

In general, users had heterogeneous opinions about whether they wanted to be informed about the exact fraction of paths explored for each option (versus only whether the option is non-, partially, or fully explored). For example, when asked this as a Likert question, we observed high variance ( $2 \times 1s$ ,  $7 \times 2s$ ,  $5 \times 3s$ ,  $4 \times 4s$ ,  $5 \times 5s$ ).

**Participants like the feature that hides the playback bar when inactive:** When asked, the majority (13/24) “strongly agreed” (score=5) and only 3 (out of 24) gave a Likert score below 3. The high average Likert score 4.08 (0.53) confirms our intuition and matches what is done in commercial *linear* players such as YouTube.

## 5 DISCUSSION AND CONCLUSIONS

This paper considered the problem of how to best facilitate branched navigation for users rewatching a branched video and/or users wanting to identify and skip paths that they have, or have not, already explored. It is the first to present and evaluate playback-bar solutions for this problem.

First, we presented the design and implementation of interface designs and features that help users to effectively navigate and explore branched videos. Perhaps most promising are two playback-bar designs that we call *partial view* and *extended partial view*. In contrast to prior work [17], our branched playback-bar solutions use scalable designs that allow them to be used with general, large branch structures, and provide natural transitions from one branch to the next. Our user tests and design comparisons suggest that the playback-bar solutions provide significant help, add value, come at small additional cognitive load, and are well-liked by users.

The playback-bar features are most liked when rewatching the video or when the navigation/exploration tasks become more complex. In these cases, the users generally liked more of the branch structure to be made visible, longer branch jumps became popular, and a larger fraction of the participants started to prefer making branch jumps by clicking the playback bar. However, with many participants preferring simpler solutions (e.g., a design with only fast-forward and rewind buttons) when watching the video for the first time, interesting future work could include designs that adaptively enable more functionality for later viewings and/or when the viewer performs navigational or complex tasks. For example, the partial playback bar could be enabled when reaching the end of the video or when observing a sequence of rewind/forward jumps.

Other promising design features that were well-liked and found helpful were the use of progress circles to show past branch points (including the option to click on them to jump back to a prior branch point) and branch buttons with color coding or  $n/N$  ratio-notation to show the extent to which different paths options had already been explored. The novel designs and the insights provided by the user study deliver important building blocks that may help engage both impatient users who might otherwise quickly leave branch video services and content-hungry users wanting to explore the video multiple times, for example. Moreover, as shown here, the design features add value to the average user.

## ACKNOWLEDGMENTS

This work was funded in part by the Swedish Research Council (VR). The authors also thank Sebastian Flinck Lindström, Eric Lindskog, Madeleine Bäckström, Linn Hallonqvist, and Rami Latif.



## REFERENCES

- [1] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2013. Graphdiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics* (2013), 740–754.
- [2] Axel Carlier, Vincent Charvillat, and Wei Tsang Ooi. 2015. A Video Timeline with Bookmarks and Prefetch State for Faster Video Browsing. In *Proc. ACM Multimedia*. 967–970.
- [3] Niklas Carlsson, Derek Eager, Vengatanathan Krishnamoorthi, and Tatiana Polishchuk. 2017. Optimized Adaptive Streaming of Multi-video Stream Bundles. *IEEE Trans. on Multimedia* 19, 7 (7 2017), 1637–1653.
- [4] Niklas Carlsson, Anirban Mahanti, Zongpeng Li, and Derek L. Eager. 2008. Optimized Periodic Broadcast of Nonlinear Media. *IEEE Trans. on Multimedia* 10, 5 (2008).
- [5] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowicz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video Browsing by Direct Manipulation. In *Proc. ACM CHI*. 237–246.
- [6] Steven M Drucker, Asta Glatzer, Steven De Mar, and Curtis Wong. 2002. Smartskip: consumer level browsing and skipping of digital video content. In *Proc. SIGCHI conference on Human factors in computing systems (CHI)*.
- [7] Eko. [n.d.]. Eko (interactive video website). [helloeko.com](http://helloeko.com)
- [8] Marco Furrini. 2008. Fast play: A novel feature for digital consumer video devices. *IEEE Trans. on Consumer Electronics* 54, 2 (2008), 513–520.
- [9] Sacha Goedegebure, Andy Goralczyk, Enrico Valenza, Nathan Vegdahl, William Reynish, Brecht Van Lommel, Campbell Barton, Jan Morgenstern, and Ton Roosendaal. [n.d.]. Big Buck Bunny (video). <https://peach.blender.org/download/>
- [10] Carl Gutwin and Chris Fedak. 2004. Interacting with Big Interfaces on Small Screens: A Comparison of Fisheye, Zoom, and Panning Techniques. In *Proc. of GI*. 145–152.
- [11] Dag Johansen, Pål Halvorsen, Håvard Johansen, Håkon Riiser, Cathal Gurrin, Bjørn Olstad, Carsten Griwodz, Åge Kvalnes, Joseph Hurley, and Tomas Kupka. 2012. Search-based composition, streaming and playback of video archive content. *Multimedia Tools and Applications* 61 (2012), 419–445.
- [12] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. 2008. DRAGON: A Direct Manipulation Interface for Frame-Accurate in-Scene Video Navigation. In *Proc. ACM CHI*. 247–250.
- [13] Vengatanathan Krishnamoorthi, Patrik Bergström, Niklas Carlsson, Derek Eager, Anirban Mahanti, and Nahid Shahmehri. 2013. Empowering the Creative User: Personalized HTTP-based Adaptive Streaming of Multi-path Nonlinear Video. *ACM CCR* (2013), 53–58.
- [14] Vengatanathan Krishnamoorthi, Niklas Carlsson, Derek Eager, Anirban Mahanti, and Nahid Shahmehri. 2014. Quality-adaptive Prefetching for Interactive Branched Video Using HTTP-based Adaptive Streaming. In *Proc. ACM Multimedia*. 317–326.
- [15] Vengatanathan Krishnamoorthi, Niklas Carlsson, Derek Eager, Anirban Mahanti, and Nahid Shahmehri. 2015. Bandwidth-aware Prefetching for Proactive Multi-video Preloading and Improved HAS Performance. In *Proc. ACM Multimedia*. 551–560.
- [16] Myunghee Lee and Gerard J. Kim. 2010. Empathetic Video Experience through Timely Multimodal Interaction. In *Proc. ICMI-MLMI*.
- [17] Eric Lindskog, Jesper Wrang, Madeleine Backstrom, Linn Hallonqvist, and Niklas Carlsson. 2019. Generalized Playback Bar for Interactive Branched Video. In *Proc. ACM International Conference on Multimedia (ACM Multimedia)*.
- [18] Britta Meixner. 2016. A pattern-based evaluation of download and cache management algorithms for annotated interactive non-linear videos. *Multimedia Systems* (2016), 1–35.
- [19] Britta Meixner. 2017. Hypervideos and interactive multimedia presentations. *Comput. Surveys* 50 (2017).
- [20] Britta Meixner and Christoph Einsiedler. 2016. Download and Cache Management for HTML5 Hypervideo Players. In *Proc. ACM HT*. 125–136.
- [21] B. Meixner and H. Kosch. Sep. 2012. Interactive Non-linear Video: Definition and XML Structure. In *Proc. ACM DocEng*.
- [22] Britta Meixner, Johannes Köstler, and Harald Kosch. 2011. A Mobile Player for Interactive Non-linear Video. In *Proc. ACM Multimedia* (Scottsdale, Arizona, USA). ACM, 779–780.
- [23] Britta Meixner, Katarzyna Matusik, Christoph Grill, and Harald Kosch. 2014. Towards an easy to use authoring tool for interactive non-linear video. *Multimedia Tools and Applications* (2014), 1251–1276.
- [24] Britta Meixner, Beate Siegel, Peter Schultes, Franz Lehner, and Harald Kosch. 2013. An HTML5 Player for Interactive Non-linear Video with Time-based Collaborative Annotations. In *Proc. ACM International Conference on Advances in Mobile Computing & Multimedia*. 490–499.
- [25] Ehsan Sotoodeh Mollashahi, Md Sami Uddin, and Carl Gutwin. 2018. Improving revisitation in long documents with two-level artificial-landmark scrollbars. In *Proceedings International Conference on Advanced Visual Interfaces*.
- [26] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga, and Jean-Daniel Fekete. 2009. Topology-aware navigation in large networks. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*.
- [27] Luis A. R. Neng and Teresa Chambel. 2010. Get Around 360° Hypervideo. In *Proc. MindTrek*. 119–122.
- [28] Emmanuel Pietriga and Caroline Appert. 2008. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1343–1352.
- [29] Nitin Sawhney, David Balcom, and Ian Smith. 1997. Authoring and navigating video in space and time. *IEEE Multimedia* 4, 4 (1997), 30–39.
- [30] Ulrike Spierling, Sebastian A Weiß, and Wolfgang Müller. Dec. 2006. Towards Accessible Authoring Tools for Interactive Storytelling. In *Proc. ACM TIDSE*.
- [31] Yap-Peng Tan, YongQing Liang, and Jun Yu. 2002. Video transcoding for fast forward/reverse video playback. In *Proc. ICIP*.
- [32] Laura Toni and Pascal Frossard. 2017. Optimal Representations for Adaptive Streaming in Interactive Multiview Video Systems. *IEEE Trans. on Multimedia* 19, 12 (Dec. 2017).
- [33] Md. Sami Uddin, Carl Gutwin, and Alix Goguy. 2017. Using Artificial Landmarks to Improve Revisitation Performance and Spatial Learning in Linear Control Widgets. In *Proc. ACM Symposium on Spatial User Interaction (SUI)*. ACM.
- [34] Albin Vogel, Erik Kronberg, and Niklas Carlsson. 2021. Fast-forwarding, Rewinding, and Path Exploration in Interactive Branched Video Streaming (Project page + Player + Code). <https://www.ida.liu.se/~nikca89/papers/mm21.html>
- [35] Maarten Wijnants, Jeroen Leën, Peter Quax, and Wim Lamotte. 2014. Augmented video viewing: transforming video consumption into an active experience. In *Proc. ACM MMSys*. 164–167.
- [36] Maarten Wijnants, Peter Quax, Gustavo Rovelo Ruiz, Wim Lamotte, Johan Claes, and Jean-François Macq. 2015. An optimized adaptive streaming framework for interactive immersive video experiences. In *Proc. IEEE BMSB*. 1–6.
- [37] Wikipedia. 2019. [https://en.wikipedia.org/wiki/Black\\_Mirror:\\_Bandersnatch](https://en.wikipedia.org/wiki/Black_Mirror:_Bandersnatch)
- [38] Yanping Zhao, Derek L. Eager, and Mary K. Vernon. 2007. Scalable on-demand streaming of nonlinear media. *IEEE/ACM Trans. on Networking* 15, 5 (Oct 2007).