

Designing for Collaborative Video Editing

ANONYMOUS AUTHOR(S)*

This paper explores the design space of collaborative video editing through a series of design workshops with video editors. Collaborative video editing can be supported by adding awareness features or other well-known collaborative features found in existing software and introducing new features designed specifically for video editing software. The paper identifies different design concepts that illustrate how such collaborative features can be included in non-linear video editing software and discusses the challenges of introducing such features. Some design concepts are explicitly inspired by existing collaborative tools. However, we suggest that introducing such features might not be straightforward. In other cases, alternative abstract representations of time-based media might be necessary to support collaborative video editing.

CCS Concepts: • **Human-centered computing** → **Collaborative interaction**; **Collaborative content creation**; **Graphical user interfaces**.

Additional Key Words and Phrases: video editing, collaborative video work, video production, user interface design

ACM Reference Format:

Anonymous Author(s). 2022. Designing for Collaborative Video Editing. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 17 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Video editing is a complex activity embedded in a social setting, including collaborators with different roles and skillsets such as montage, colour correction, sound levelling and graphics production [Bartindale et al. 2012]. At the same time, the editing per se is most often done in solitude by a single person, and the supporting software is primarily designed for an individual user. Through a number of participatory design sessions, we investigate challenges and opportunities in enabling online collaboration in video editing.

Video editing software has a specific challenge that it tries to solve—how to represent the temporal dimension of media and allow its manipulation [Mackay and Pagani 1994; Santosa et al. 2013]. Existing video editing software, such as Adobe Premiere and Apple Final Cut, utilises abstract representations and metaphors such as a timeline and thumbnails [Goldman et al. 2006, 2008]. The timeline is commonly a horizontal representation of the temporal dimension, going from left to right. In the timeline, media elements, such as clips and images, are often represented as rectangles, with the length of a rectangle representing the duration of the media. The media elements are often supplemented by thumbnails demonstrating a frame from a clip [Goldman et al. 2006]. This abstract representation presents limited information about the underlying video and mediates the editing of vital aspects of the video such as narrative, movement and sound. Thus video editors are required to repeatedly recall and review the video during the editing process to see and evaluate the results of their editing [Goldman et al. 2006; Okopnyi et al. 2020; Pavel et al. 2016].

Given the emergence of new ways of working, such as remote work in TV and video production automation [Girgensohn et al. 2000; Kim et al. 2015; Okopnyi et al. 2021], as well as the availability of new technologies, there is

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

Manuscript submitted to ACM

53 a growing interest in the industry in the development of video workflows that would support online collaboration
54 and concurrent video editing. Video production used to be exclusive to large organisations due to its high cost. New
55 ways of producing video content have emerged as both high-speed communications networks and production tools
56 have become cheaper and more readily available for distribution. It has become more affordable to conduct editing
57 at multiple places due to the emergence of inexpensive digital video recorders and non-linear video editing software
58 [Juhlin et al. 2014]. Furthermore, the introduction of social distancing rules during the COVID-19 pandemic forced TV
59 and video professionals to adapt to the new ways of working and to tackle the idea of cloud-based video work [Okopnyi
60 et al. 2021]. In sum, there is a growing mismatch between the collaborative production setting and the individualised
61 functionality of the editing software [Pavel et al. 2016].
62

64 The topic of supporting collaborative video editing is further inspired by research on collaborative tools for other
65 types of media production. Examples include research to make writing [Baecker et al. 1995; Fish et al. 1988], music
66 creation [Biasutti 2015; Bligh et al. 2005] and drawing [Ignat and Norrie 2006; Tang and Minneman 1991] collaborative.
67 Further, research in the area of Computer-Supported Collaborative Work (CSCW) [Schmidt and Bannon 1992] has
68 shown the importance of “workspace awareness” [Dourish and Bellotti 1992; El Helou et al. 2008; Gutwin and Greenberg
69 2002] in real-time distributed collaborative work. Gutwin and Greenberg [2002] suggest that user interface design
70 should provide users with information about the state of the shared workspace by answering questions in several
71 categories, such as “Who is present?”, “What are they doing?” and “Where are they working?” [Gutwin and Greenberg
72 2002].
73

75 When collaborators are continuously cognisant of each other’s ongoing work, it helps them coordinate and adapt
76 to each other. The focus on concrete and ongoing practical details in work settings has led to the generation of new
77 applications and services [Karis et al. 2016; Okopnyi et al. 2021; Roseman and Greenberg 1996]. Additionally, CSCW
78 has over the years also identified a number of challenges in designing for collaboration [Cockburn and Jones 1995;
79 Grudin 1988, 1994; Larsen-Ledet and Korsgaard 2019].
80

82 The topic of collaborative video editing has already attracted interest in the area of human-computer interaction
83 (HCI), and then in the part of video editing that is currently most concretely about collaboration, that is, the reviewing
84 [Okopnyi et al. 2020; Pavel et al. 2016]. We argue for a potential to expand existing research on collaborative video
85 reviewing by providing a broader perspective on the possibilities for collaborative editing and especially on such
86 features that increase awareness in a video editing context. Specifically, we ask what the scope is for design research on
87 support for collaborative video editing.
88

89 To address the question, we carried out this study in two steps. First, we invited twenty persons to take part in five
90 design sessions to generate a set of design ideas on the topic of awareness support in video editing. The participants
91 ranged from media students to professional editors. The sessions resulted in design concepts such as “Time Slot
92 Separation” and “Scaffolding.” Second, we analysed the discussions around the concepts and identified high-level themes
93 such as “focus” and “resemblance” that put said concepts in the context of their potential uses.
94

96 In our analysis, we identify challenges of enabling collaboration in video editing, such as the need to account for
97 potential distractions introduced by new elements of the user interface. We conclude that the introduction of collaborative
98 elements might not be straightforward due to the fact that video editors work with abstract representations of media.
99

2 COLLABORATIVE VIDEO WORK: PREVIOUS RESEARCH AND EXISTING TOOLS

There is a strong interest in enabling collaborative video work in the industry illustrated by the emergence of multiple tools such as frame.io¹, ftrack² and ReviewStudio³. These services allow uploading or casting a video or other media to remote participants and gathering feedback through annotations and online video sessions. They also all aim to enable collaborative reviewing. In all, they are limited in that they do not support multiple video editors to collaboratively edit video in a shared workspace.

There are attempts to create a collaborative video editing software. Service motionbox.io⁴ provides a simplified web-based video editing tool that allows multiple participants to work on the same video project simultaneously. However, the service is still under development and does not provide any specific collaborative features that would help co-editors to communicate and coordinate their activities.

Some professional video editing suites, such as Avid Media Composer and DaVinci Resolve, allow multiple participants to work on the same project by locking certain parts of the project. This feature is known as “bin locking.” It implies that only one user can manipulate elements of video, that is, clips, that are inside the bin – a virtual folder, while other users can access the bin in a read-only mode. DaVinci Resolve also includes a built-in chat to enable communication between video production team members⁵.

In research, there have been multiple attempts to design, prototype and develop collaborative systems for video work when it comes to technological viability, means of interaction and reviewing practices.

As early as 2006, Fonseca and Carrapatoso [2006] provided a paper that argued for the potential of collaborative video editing given new Internet technologies. The topic was motivated by achievements in the research area of CSCW. They argued that collaborative activities, such as communication, sharing and joint visualisations, could be of benefit also in video editing. Their study showed that the idea was technologically viable, but at the same time stated that: “The implementation of a complete cooperative video editing tool is a quite complex task and would require the involvement of a large multidisciplinary team.”

The technical viability of collaborative video editing was further demonstrated by Klokrose et al. [2019], who presented a web-based toolkit “Videostrates.” The toolkit provided the basis for development of collaborative video tools with personalised interfaces that are tailored towards specific tasks, and emphasised the need for further investigation of real-time collaboration around video.

The possibility of digital video editing sparked further design research on the means of interaction with video. Although the media is visual and aural, it was argued that editing could be collaborative and allow mutual awareness if making cuts and realigning the time sequences became more tactile. Taylor et al. [2007] presented a tabletop editing system named VideoPlay, which presented video clips in the form of plastic tiles and allowed their combination into video sequences. The prototype aimed to make the video editing experience playful and more engaging and allowed several people to work on the video. Zigelbaum et al. [2007] presented a prototype that aimed to combine the benefits of traditional linear and more modern non-linear video editing. The prototype presented video clips as plastic tokens, allowing participants to rearrange them in order to create a coherent video. Terrenghi et al. [2008] designed an interactive tabletop application for collaborative video editing that aimed to enable video work as a spontaneous activity, allowing all interested parties to be involved in the editing process. Another tangible video editor prototype was presented by

¹<https://frame.io/>

²<https://www.ftrack.com/>

³<https://www.reviewstudio.com/>

⁴<https://motionbox.io>

⁵<https://www.blackmagicdesign.com/no/products/davinciresolve/collaboration>

157 Merz et al. [2018] who aimed to create a more engaging video editing experience for school children and promote
158 digital storytelling literacy. Similarly to previous works, the prototype required participants to be physically present in
159 the same space and was limited in its video editing functionality.

160 The call for research on collaborative video applications [Fonseca and Carrapatoso 2006; Novikov and Proskurnin
161 2003; Takhirov and Duchateau 2011] has generated a specific interest in reviewing practices. Pavel et al. [2016] developed
162 a video reviewing system that allowed reviewers to leave feedback for video editors. Their system allowed reviewers
163 to record video feedback that would include non-verbal cues to the editors. Okopnyi et al. [2020], in their study of
164 communication between video editors and other stakeholders, noted that video work is often open-ended and requires
165 video editors to devise various strategies to avoid or resolve conflicts and misunderstandings. They suggested three
166 design directions for video reviewing and editing software: scaffolding, iconic referencing and suggestive editing.

167 Bartindale et al. [2016, 2012] developed and evaluated experimental tools such as TryFilm [Bartindale et al. 2016] and
168 StoryCrate [Bartindale et al. 2012] to support collaboration during and immediately after the filming process. These
169 tools allowed filming crews to review the filmed footage, reflect upon its quality and perform basic video editing such
170 as cutting and rearranging clips. Similarly to other tangible and tabletop systems, these tools required the physical
171 presence of participants.

172 In sum, the technical opportunities in supporting collaborative video editing have been mostly topicalised in research
173 in two ways: by enabling playful, spontaneous and more engaging interaction in a co-present setting and by introducing
174 sharing mechanisms in the reviewing process. This research is interesting but still small in scope. The former type
175 provides new forms of co-located interaction in a studio given traditional linear editing principles where cuts are
176 made and sections are rearranged and glued back. The focus on reviewing is interesting as a starting point for remote
177 collaborative video editing since this practice is currently most salient when it comes to social interaction. Still, there
178 are many parts of the editing that are performed in a social setting, which is not accounted for in these applications.

185 3 METHODS

186 Our approach to expanding the design space for collaborative video editing includes a set of design workshops and
187 inductive analysis of the generated material, that is, the design concepts and the transcripts of participants' discussions.
188 We conducted two rounds of design workshops on the topic of supporting collaboration in a video production team.
189 The first round focused on the topic of workspace awareness [Gutwin and Greenberg 2002]. The second round had a
190 broader scope. In all workshops, the participants were instructed that the ideation should focus on remote video editing,
191 including both synchronous and asynchronous collaboration between video editors who use current non-linear editing
192 software tools in a professional capacity and work in production teams.

196 3.1 The design workshops (WS1-2)

197 The first round was dedicated to ideation in two separate workshops (WS1-2) with eight participants in total. The initial
198 workshop was conducted with three master's students who study media and human-computer interaction. With them,
199 we investigated the workshop protocol. The evaluation included topics such as whether the task descriptions and other
200 materials were clear enough, whether the time for completing the tasks was enough and whether the overall pacing of
201 the workshop was good.

202 In the second workshop, we recruited five people via social media. All participants had experience in video editing
203 ranging from participating in high school video projects and hobby projects to studying TV and video production at
204 university and doing professional video work.

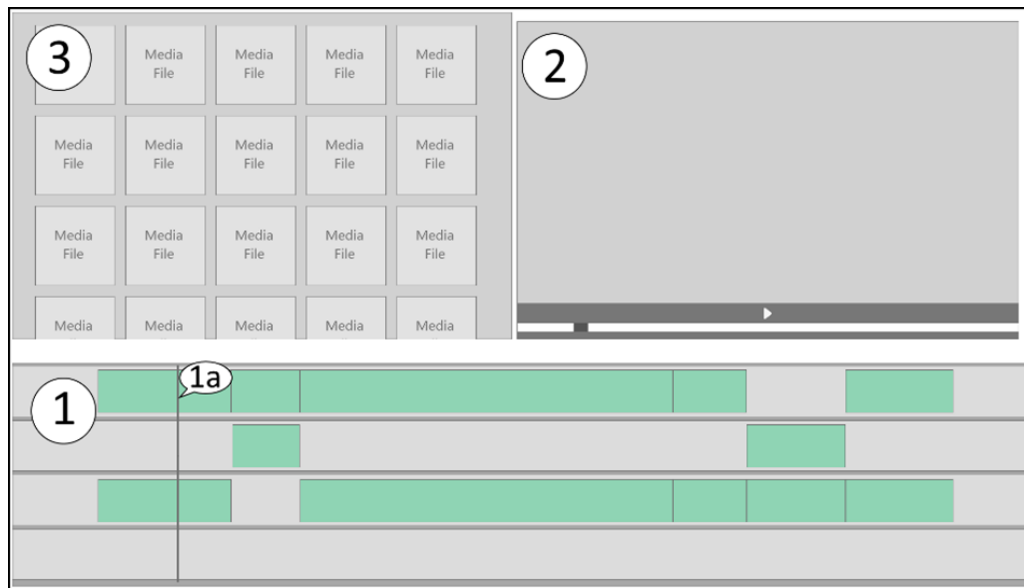


Fig. 1. Schematic Video Editor UI

Both workshops in the first round (WS1-2) followed the same protocol. All participants were introduced to the workshop's topic and provided with general CSCW theory as inspiration, that is, the definition of workspace awareness as presented in [Gutwin and Greenberg 2002], and examples of workspace awareness categories and features in existing online collaborative services, such as Google Docs, Overleaf and Balsamiq. We also presented a schematic illustration of the video editing software UI (Fig. 1). The schematic constitutes several elements that are commonly found in existing video editing software. First, there is a timeline (Fig. 1-1) that contains graphics, video and sound tracks. Inside tracks, there are clips that represent pieces of media. On top of the timeline, there is a playhead (Fig. 1-1a) that represents the current position of the user in the timeline. Second, there is a preview window (Fig. 1-2) that allows watching the current version of the video. Last, there is a media catalogue (Fig. 1-3), a set of media files that can be ingested into the timeline, such as raw footage files, pre-recorded audio files, and various graphical elements such as images.

Thus, we suggested that participants should think of ways to augment or change the existing software user interface along with new workflows that they would like to introduce. Therefore, we tried to ground design ideas into existing software and refrain from inventing completely new user interfaces that would be unfamiliar to video workers.

After the presentations, we had a brainstorming session in which participants worked individually to suggest as many design ideas as possible. The ideas were written down on sticky notes. They were then developed into prototyping sketches on paper. Then, each participant presented their ideas and discussed them. The most promising ideas were selected and re-developed by the participants. Finally, they discussed the relevance and feasibility of the produced ideas.

The second workshop in the first round was video-recorded and transcribed.

3.2 The second round of workshops (WS3-5)

In the second round (WS3-5), we used design ideas generated during the first round (WS1-2) as an inspirational starting point. Due to the COVID-19 pandemic and social distancing restrictions, we conducted three online workshops attended by four participants each. The participants included:

- three current TV editors, directors, and producers;
- one TV automation specialist;
- one multimedia production specialist with a background in motion design and TV show production;
- four current freelance video editors and filmmakers with experience ranging from YouTube video production and documentaries to professional TV work; and
- three UX designers with experience in design for professional visual media production tools and video work ranging from professional video editing to running a YouTube channel.

The participants have relevant work experience in various European countries, including Bulgaria, Denmark, Norway, Serbia, Ukraine and the United Kingdom.

Each workshop comprised three stages. First, participants were introduced to the workshop's topic – what features an online collaborative editing tool is required or expected to have. Participants also introduced themselves and described their video work experience. We demonstrated the design ideas developed in the first round of workshops (WS1-2) and discussed the relevance of presented ideas to the field and participants' work experience. Participants were encouraged to express their opinions and use these ideas to inspire the design. Second, we encouraged participants to develop the presented ideas further or propose new design concepts. We used the Balsamiq.cloud web application to work collaboratively on prototypes. We reflected on the ongoing online collaboration experience and discussed possible design interventions to enable concurrent work. Third, we concluded the workshop with a short discussion of the feasibility and relevance of the developed prototypes.

All workshops in the second round were video-recorded and transcribed.

3.3 Data Analysis

The workshops yielded two types of data that were further analysed: design concepts and the transcripts of participants' discussions. We searched the transcripts for the reasoning behind the design concepts. Specifically, for each design idea, we tried to answer three questions: How should it work? Why is it beneficial? What are the drawbacks or disadvantages? While we did not explicitly specify the setting of video production for the ideation process, the participants with their broad and varied experience from editing brought up many examples of relevant contexts and settings in their discussions.

We employed content analysis, a research technique that aims to make valid inferences from texts and other meaningful materials in the context of their use. With this approach, we were able to reduce collected data, that is, transcripts, by identifying core patterns and ideas [Patton 2014]. Then, from the volume at which the subject matter was discussed, we inferred the importance that workshop participants assigned to identified patterns and ideas [Krippendorff 2018].

The data were coded inductively in three iterations. We started with the most apparent aspect for each design idea as workshop participants voiced it. Then we looked for various motivations and possible challenges for design ideas. In the third iteration, we identified overarching themes and patterns that describe such motivations and challenges. The analytical work laid the foundation for the coding scheme, which was refined through revisiting the material between

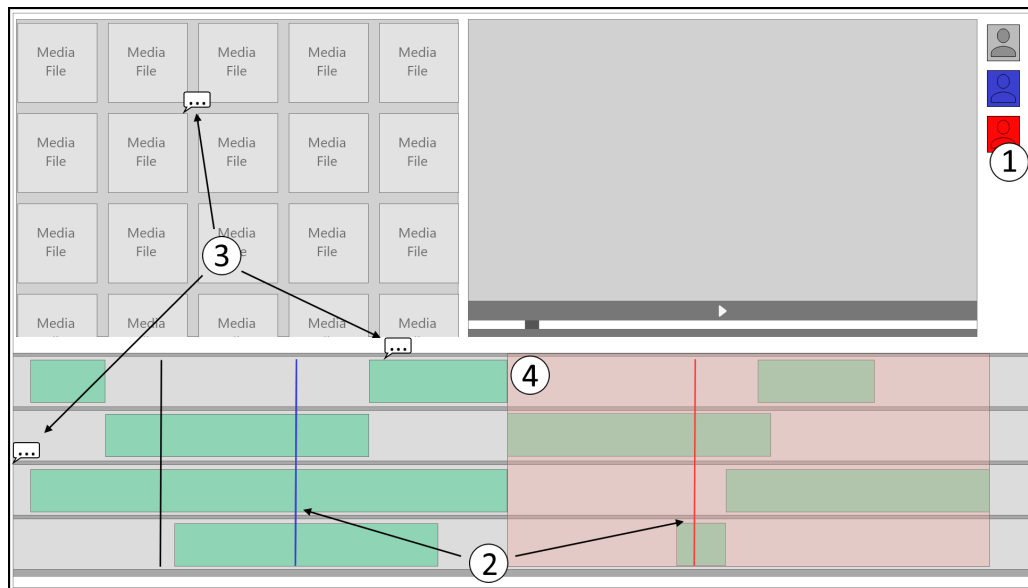


Fig. 2. Color-coded avatars and 2. playheads. 3. Annotations, and 4. Time Slot Separation

coding sessions. In the end, we arrived at high-level categories that address various aspects of collaborative video editing and put design concepts in the context of their potential use.

4 RESULTS AND ANALYSIS

During the workshops, the participants produced multiple design ideas. Here, we present design concepts, which were developed into either pen-and-paper or digital sketches and then discussed by the participants. In order to convey the meaning of the sketches better to non-participants, the authors of this paper have developed the sketches further. We also present our categories (in *italics*) that express participants' descriptions of the advantages and potential challenges of the concepts.

Here, we first mention several design concept that might be considered "generic" to collaborative tools in general. Then we describe in more detail design concepts which, in our opinion, require more attention from research and design communities.

4.1 Enabling Workspace Awareness

Some design concepts were to a large extent inspired by other collaborative tools, such as Google Docs, and the workspace awareness categories as described in [Gutwin and Greenberg 2002]. Those ideas include using colour-coding on avatars (Fig. 2-1) and other interface elements, such as playheads (Fig. 2-2), emphasising salient actions performed by other users with visual effects, integrating chat, and adding an editing log. The participants suggested that the implementation of such concepts "[would be] on par with most real-time collaborative systems, like Google Docs." Thus, due to the *resemblance* to existing tools, these features should be recognisable and not require additional effort from users to learn the functionality.

365 Even though these features exist in various collaborative software, the participants still expressed multiple concerns
366 regarding their presence in collaborative video editing software. Specifically, additional dynamic UI elements, such as
367 multiple coloured playheads (see Fig. 2-2) or new message notifications, might become a source of *distraction*:
368

369 “If I’m editing, and I’m doing my work, and I suddenly see a lot of highlighted [elements] around me,
370 it’s really distracting.”
371

372 At the same time, the participants recognised the potential of such additions to help work collaboratively by providing
373 awareness of “who” and “where” is present and “what” they are doing in the shared workspace: “you would have a
374 focus on what he’s doing, so you can follow what’s happening.” For example, an integrated chat could be augmented
375 with support for indexical referencing – timecodes entered in the chat should be transformed into references that take
376 users to the referred-to time frame when clicked on.
377

378 Previous research on reviewing [Okopnyi et al. 2020; Pavel et al. 2016] shows that indexical referencing is essential to
379 communication around video. In a co-located setting, pointing is one of the most common activities in which co-editors
380 engage [Terrenghi et al. 2008]. Thus, the ability to refer to certain points of video via timecodes and address certain
381 people in discussions is crucial for video workers and should be supported by online collaborative video editing software.
382
383

384 4.2 Annotations 385

386 The idea suggests adding textual notes to various elements of the video project, such as media clips, images, and different
387 elements of the timeline (Fig. 2-3). Notes can be addressed towards a person or a group of people and are intended to
388 draw collaborators’ attention to areas of demand for further work:
389

390 “I’m a video editor, and I’m making notes, for example, to a sound designer: Here, [the] noise is a little
391 too harsh, [can] you clean this a little bit?”
392

393 Annotations in time-based media have been topicalised in both academic research [Okopnyi et al. 2020; Pavel et al.
394 2016] and industrial applications such as frame.io. Our participants, who were inspired by tools such as Google Docs,
395 wanted to expand these ideas and employ annotations as a task-tracking tool, as well as means of communication
396 between collaborators. For example, in contrast to frame.io, our participants suggested that annotations should not be
397 attached to static points in the timeline (timecodes) but to clips and other elements that can be moved by editors.
398

399 There should also be an overview of all notes addressed to the editor in order to help track the progress and to
400 navigate problematic areas of the project, without the need to go through the timeline:
401

402 “You also need one place where all notes [are listed] so you don’t really miss on any of those. [For
403 example,] you give the notes [to] a sound designer [and] he can open all notes in one list, so he doesn’t
404 need to go through [the] timeline, so he’s sure he does not miss any of [the notes].”
405
406

407 Again, the participants argued that the idea was motivated by its *resemblance* to existing software, specifically the
408 commentary feature in Google Docs, which allows using notes to track progress, communicate vital information to
409 collaborators and assign tasks to each other:
410

411 “Like [in] Google Docs, [...] you make notes for someone, and when this problem is solved and [marked
412 as] solved, [the note] disappears.”
413

414 Further, it allows switching to a new *digital workflow* instead of “taking notes manually,” as a participant described
415 their collaboration:
416

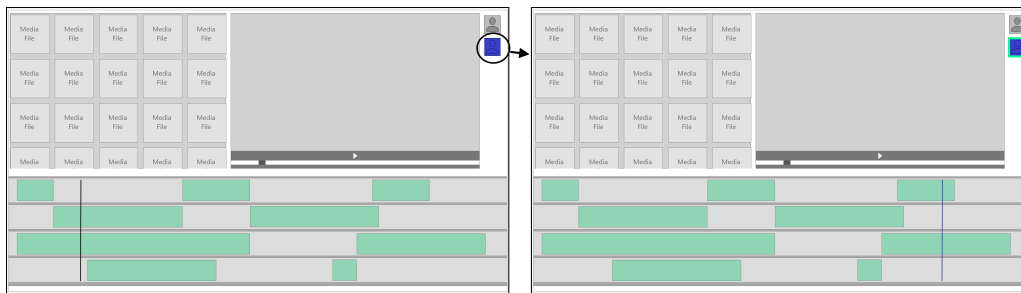


Fig. 3. Spectating mode: video editor UI (left) and spectating another editor (right)

“I used to take notes manually, so [the producer] would watch [the video], stop it, I will write timecode or sometimes mark it on the timeline, but still I will have to write the comment in my notebook.”

However, another participant expressed concern that the implementation of this design concept might be detrimental and obstructive to the editing, as it will take up *limited space of the user interface* and present too much information to the editor:

“[There is] a thing with taking up the space of the interface [...] you would always have to be aware of what’s the most important to look at.”

4.3 Spectating Mode

A spectating mode is a special mode added to the collaborative editing software that allows other editors to look at others’ work while working on other parts. Video work in a co-located setting has many advantages, as was demonstrated by previous research [Taylor et al. 2007; Terrenghi et al. 2008; Zigelbaum et al. 2007]. The participants drew on their experiences of working in a face-to-face setting and suggested this design concept that allows remote editors to look at what their collaborators are currently doing. The mode works as follows. The user clicks on the collaborator’s avatar in the UI (Fig. 3-left), and the software goes into spectating mode. It then presents the shared workspace from the point of view of the selected collaborator, allowing real-time observation but not interaction with it (Fig. 3-right).

As a participant suggested, this concept would allow a new *digital workflow* for participants who otherwise had to be physically near to the editor in order to collaborate:

“You can discuss what you’re doing. If the director or someone is not in the same place, instead of coming into the editing room, you can just do it like this.”

It would also allow *saving time* in situations when the editor needs to confirm the edits with other project members:

“I’ve lost [a lot of] time because when I had to confirm with my clients or my team project, I [had] to render [the video] and send [it to] them.”

However, one participant expressed concern regarding *privacy of work* in this kind of setup:

“What I would prefer [...] if I’m editing and someone wants to look at what I’m doing, they [would] have to be granted access to my editing for viewing, so they will not be able to monitor what I’m doing without me knowing.”

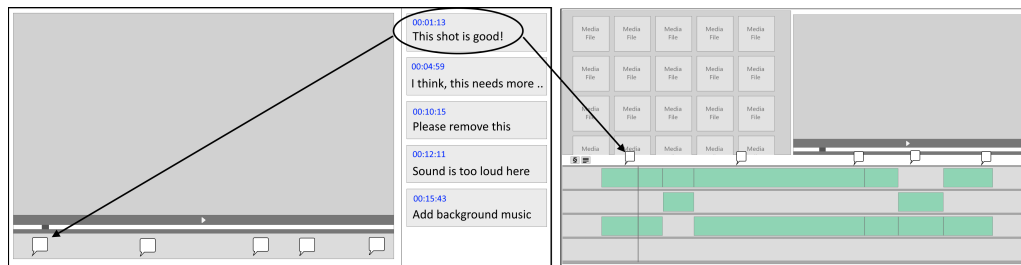


Fig. 4. Reviewing tool: reviewer UI (left) and video editor UI (right)

4.4 Integrated Reviewing Tool

This idea is inspired by existing software, such as frame.io, which allows third parties to leave comments on the video attached to certain timestamps and have a discussion around the video as a whole.

The integrated reviewing tool allows immediate communication between the editor and a collaborator who is not directly involved in the editing. For example, an editor can demonstrate specific fragments of the video to a producer without needing them to watch the whole video. The reviewing tool also allows reviewers to leave notes with their feedback attached to the timeline via a web-based interface. The notes are immediately available for the editor in the video editing software (Fig. 4).

Existing software, such as frame.io, lacks integration with editing software, and it requires video editors to perform *additional work* to get feedback:

“Frame.io is very convenient. I just would like to skip the step of rendering and sending [the video]. [It would be] a huge help [if] they could see the [video in] real time in good quality because either you send it to them in bad quality or they have to wait [for] a few hours.”

The concept was positively regarded by the participants. They suggested that having the ability to initialise a feedback session instantly, without the need to render the video file and upload it to a third-party service, is beneficial to the workflow.

Reviewing and quality of communication between parties are essential in commercial video production. However, in previous research [Pavel et al. 2016] and industrial applications such as frame.io, the reviewing process is treated as an asynchronous activity that happens in iterations. Our participants suggested that synchronous real-time collaboration between editors and reviewers would be a “huge help” to the video production process. Similar to the spectating mode (4.3), this design idea allows *saving time* by eliminating certain steps of the reviewing workflow:

“It would be really interesting [if] I can just call to my client [and say]: Hey, can you look at the 1 minute 32 seconds? Do you like this? what do you think about it?’ without rendering the video. This is probably the first problem which I have as an editor at the project.”

However, again one participant suggested that such a tool might be *distracting* to the editors if the feedback is not limited. They suggested that the editor should be in control of the situation: “[Clients] have to be let in only when they can be let in, when you’re ready.” Another participant mentioned that they would prefer not to *introduce new tools that other parties are not familiar with* into the workflows and communicate with other parties via more commonly used software: “If a client contacted me on Skype, we are going to have the call on Skype.”

4.5 Time Slot Separation

This design concept is based on the idea of dividing a project into several areas that can be distributed between co-editors to ensure concurrent work. An editor, who works on a segment of the video, defines a slot, that is, a section of the timeline, by selecting video clips in the timeline and marks this section as something they are working on. Other collaborators are not allowed to make alterations, but they can see what is being worked on by their colleagues (Fig.2–4). When the acting editor has completed the task, they mark the section as ready for the next step of post-production.

The concept suggests separating collaborators into various time segments of the shared workspace. Our participants discussed various ways of segmentation. They concluded that the separation should be based on media elements, that is, clips. However, this approach still needs to be evaluated and compared to other possible methods of separation, for example, based on concrete timecodes.

The participants argued that the benefits of such a concept were the way it allows editors to *focus* on their work and not worry about the integrity of their editing:

“[There is] a section that I want to focus on because I need to do some modifications within it, and I don’t want other people to do anything in this area right now.”

Further, the possibility to restrict access provides the editor’s *privacy of work*:

“Only I [should] see what I’m doing. [So] I could concentrate better. Because sometimes I try something [that] I don’t like.”

On the other hand, a participant suggested that this feature might be excessive and bring in more *complexity* to an already challenging process of video editing: “I think it adds a level of complexity [and video] editing is already complex.”

4.6 Master Styles and Functional Separation

Most current collaboration occurs between people with different roles, such as the interaction between an editor and the graphics producer. In order to improve on such work, the participants suggested a feature that allows decoupling of the content of graphics, such as text and style. The editor who works on montage and is responsible for the content of the video can insert a graphics placeholder and put the necessary text into a free-form input field while the graphics style is incomplete (Fig.5—left). When the graphics producer updates the style, all graphical elements in the project are automatically updated (Fig.5—right). The concept extends to organised, functional separation. It enables concurrent work in between multiple collaborators instead of the way such interaction is now sequentially pursued.

Again, the concept draws on its *resemblance with existing software*, specifically Adobe Photoshop and Adobe XD:

“The first [graphical element] is like a master component [in Adobe XD], and if you change the style of it all [graphical elements] are going to change as well.”

It is suggested to be beneficial by *saving time* when editing with long videos, such as feature films:

“When you have a project [that is] one hour long, and then you decide [that] you want to tweak up this template, then it will change like all, like 50 or 100 elements in the whole timeline. [This] could take you two hours to change by hand.”

573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624



Fig. 5. Master Styles: Editor enters text for graphics into a free form without template (left). Graphics template is finished and applied to graphical elements (right).

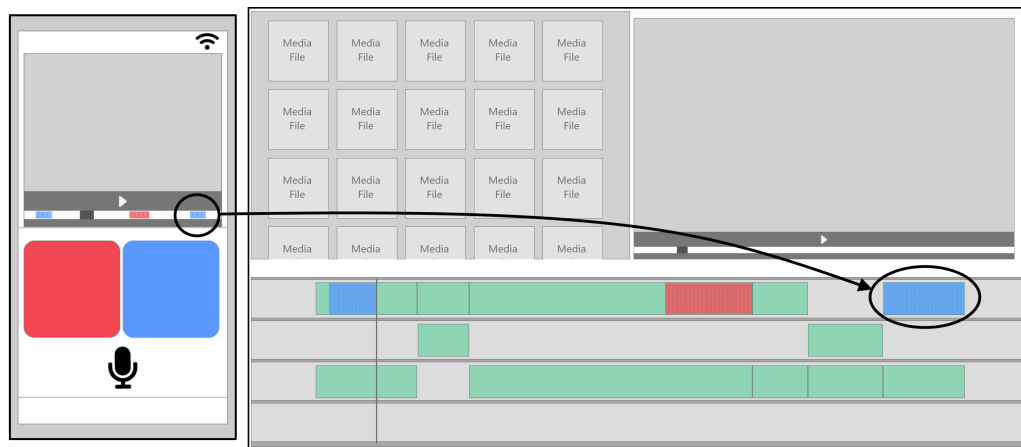


Fig. 6. Mobile application (left) and areas of the footage highlighted in the video editing software UI (right)

4.7 Scaffolding

Workshop participants emphasised that in collaborative work, there is a need to have a general organisational structure to the video. This is often provided by a director, a writer, or a project manager, who does not participate in the actual editing. Similar to the design ideas presented in [Bartindale et al. 2016, 2012; Okopnyi et al. 2020], they suggested attaching documentation describing such structures in the editing software. It would be represented as another layer of meta-information that is generated during the filming process:

“When [...] working with the director [...] there should be some kind of a file attached to [the project] where they write guidelines for general editing.”

In order to make such notes easier, the participants suggested a mobile app that would be synchronised with the editing software. Specifically, it would allow selection of a fragment of video and highlight it with colour and add a voice note during production and recording of videos (Fig.6—left). When the footage is ingested into the video editing application, the editor can identify where on the timeline there are additional comments (Fig.6—right).

The idea resembles the current usage of clapper boards in filmmaking [Dumont and Mérialdo 2008]: it is a device that helps designate and mark various video segments. It provides an opportunity to add metadata, such as scene title,

625 in the form of notes or plain text messages. The concept of scaffolding suggests enabling similar digital notes that
626 would be associated with certain timeframes and can be presented in the timeline. This enables a *digital workflow* that
627 otherwise unfolds with analogue artifacts, such as hand-written notes or via third-party software:
628

629 “Some clients have a certain idea of what they want to use and what not to use before the editing starts.
630 Instead of giving me just plain text messages, they could just access the software and do it in there.”
631

632 This also suggests a *holistic approach* to UI design, keeping all necessary tools in one software suite and preventing
633 users from having to “go between two screens all the time.”
634

635 At the same time, a participant suggested that this solution might not be welcomed by external people, such as
636 clients, as it suggests introducing *new tools that they are not familiar with*:
637

638 “Some clients might be a little bit apprehensive about going into a software and doing anything in there;
639 they would just prefer to [write a] message.”
640

641 5 DISCUSSION

642 Presented design ideas help illustrate and make concrete the scope of supporting collaborative video editing. As
643 mentioned in 4.1 and 4.2, some of the design concepts, such as colour-coded avatars and annotations are explicitly
644 inspired by existing collaborative tools (e.g., Google Docs and Frame.io) and the workspace awareness framework
645 [Gutwin and Greenberg 2002]. Still, these concepts have to be qualified for collaborative video editing software. As the
646 participants suggested, it might be challenging to successfully implement such concepts in a working environment.
647 Other concepts are not explicitly inspired by existing tools, though we can find similar design ideas in other software.
648 For example, Time Slot Separation (4.5) can be considered similar to file-based separation in text-editing and software
649 development tools [Ede and Lunsford 1990; Larsen-Ledet and Korsgaard 2019] as well as to bin-locking feature in
650 DaVinci Resolve and Avid Media Composer. However, the lack of mass adoption of such concepts in the video industry
651 suggests that they are still to be evaluated in a working environment. In our analysis, we identify three aspects that
652 present significant challenges to designers and developers of collaborative video editing tools: focus and distraction,
653 new workflows and tools, and work with temporal dimension.
654
655
656
657
658

659 5.1 Focus and Distraction

660 Previous works on collaborative video editing showed that it is possible to enable collaboration when the editing
661 happens in real time and in a face-to-face environment: the communication flows smoothly, and collaborators can
662 see what others are doing at any moment [Pavel et al. 2016]. In this study, the participants mention concerns about
663 suggested features as being distractive (4.1), complex (4.5) or articulating the need to be able to focus (see 4.1). Since
664 such topics are raised when discussing various applications, we suggest that it is a general and serious concern. Indeed,
665 distributed collaboration demands additional effort compared to a face-to-face environment: communication between
666 group members is limited, maintaining group focus is more difficult and the work itself can be confusing and chaotic
667 [Cockburn and Jones 1995; Ellis et al. 1991].
668
669
670

671 One of the presented design concepts (4.5: Time Slot Separation) suggests that the introduction of additional structural
672 mechanisms might be able to help deal with these issues. Specifically, this concept envisions a feature that prevents
673 access by other editors and, essentially, limits possible collaboration: other editors cannot edit the same part of the
674 video. Thus, the editor who is responsible for that part can work without distraction.
675
676

5.2 New Workflows and Tools

In a number of design concepts (see 4.2, 4.3 and 4.7), our participants suggested providing support for various activities that usually occur beyond the scope of video editing software. For example, the reviewing process typically involves rendering the video and sharing it with other parties via some file-sharing service. Previous studies suggest improving the reviewers' experience by providing them with additional instruments to support annotating and indexical referencing [Chua et al. 2017; Okopnyi et al. 2020; Pavel et al. 2016]. However, the introduction of these new instruments does not solve the underlying problem, that is, the time and effort required to share a video with another person.

One design concept (4.3: Spectating Mode) essentially proposes adding to video editing software functionality that is already implemented in remote desktop services and screen-sharing features of communications applications. Our participants suggested that a mere integration of such functionality would be beneficial for the overall video editing process. Specifically, video editors would be able to get feedback from their colleagues promptly and discuss the edits while in the process of editing. This change in the mode of reviewing from iterative and asynchronous to immediate and synchronous might have a significant impact on the whole editing process. However, it is hard to tell what this impact would be, as previous research has indicated that video editors sometimes limit communication with other parties to ensure the quality of the feedback [Okopnyi et al. 2020].

Besides video itself, video production often involves various media, such as textual notes and documentation. Two design concepts (4.2 and 4.7) specifically suggest integrating these types of media into video editing software to save time and reduce the effort that is spent on switching [Cockburn and Jones 1995] between different applications and contexts when editors work with documentation, messages, and notes. However, integrating new tools and inviting non-editors to participate in the editing process might not be welcome, as our participants suggested that the introduction of unfamiliar tools can confuse less experienced users. Thus, in line with [Klokmoose et al. 2019], we suggest that a more personalised approach to user interface design should be preferable.

5.3 Working with the Temporal Dimension

The interface of a video editing software application visually displays the time-based media in a time-independent form, that is, a line. In our design exercises, this time conversion is also a given when suggesting new collaborative features. All features suggested by the participants are ideas on increased collaboration in a setting where the time feature in a video is described as a line, that is, a so-called timeline [Goldman et al. 2006].

This preconception might have influenced the participants to draw inspiration from existing collaborative production tools of media, which are less time-based than video. Concrete examples refer to collaborative features in text writing. Such associations and claims on resemblance are mentioned in several of the suggested applications (see 4.1, 4.2 and 4.6).

However, it might be that the "resemblance" between line-based time representation in video software and text-based collaborative editors would not work as straightforwardly as first imagined. For example, indexical referencing (see 4.1 and 4.2) supports coordination and collaboration in a shared workspace [Dourish and Bellotti 1992; Gutwin and Greenberg 2002] in settings such as collaborative text writing [Larsen-Ledet and Korsgaard 2019]. However, we suggest that, in the case of video, there is a potential mismatch between a reference that exists in the user interface and a referred-to object that can exist in the medium itself. In terms of semiotic theory [Peirce and Buchler 1902], the sign and the object are on different levels of abstraction. This example highlights the mismatch between what is happening with the medium during rearrangements and what is visible in the interface. Pavel et al. [2016] address this mismatch in

729 their design, allowing users to video-record their feedback and gesture “over the source video with the mouse” during
730 the recording, essentially creating a new “feedback session” timeline.

731 In the editing process, the editors need to comprehend the connection between the abstract representation of video
732 and the video itself [Goldman et al. 2006, 2008]. We suggest that this need might restrict the introduction of new
733 collaborative elements similar to those that exist in other collaborative systems. If that is the case, the concerns raised
734 might be fundamental and challenging. It seems important to restate that video editing is organised and pursued mostly
735 as an individual activity for this reason.

736 Previous research demonstrated alternative approaches to present time-based media, such as blocks [Merz et al.
737 2018], tokens [Zigelbaum et al. 2007], and storyboards [Goldman et al. 2006]. These approaches include putting a
738 structure on top of continuous media, breaking it into pieces. Specifically, each block, token, or storyboard frame
739 represents a clip – a discrete segment of video; and editors rearrange these clips to create a new video. One design
740 concept (4.5: Time Slot Separation) suggests a similar approach – to impose a structure on top of the existing timeline
741 by splitting the video into a series of time slots. Thus, the whole video can be represented as a collection of time
742 slots, similar to a storyboard, which requires less effort to observe and comprehend [Goldman et al. 2006]. We suggest
743 that storyboard-based representation may be a viable alternative or a supplement to a timeline-based design which is
744 prevalent in current video editing software. The idea to use storyboards for video editing in itself is not particularly
745 new [Mackay and Pagani 1994], however, it is still to be explored in the context of online collaboration.

746 The fact that collaborative skills and experiences, along with new types of software, are introduced to edit other
747 types of media is inspiring but not decisive for the future of video editing. The intent of this study was to broaden the
748 design space of collaborative video editing. However, an important next step is to evaluate such concepts in practice
749 and possibly shrink the space.

750 6 CONCLUSION

751 In this paper, we have examined how we can support collaborative video editing. We have carried out a series of design
752 workshops and identified, illustrated and discussed design ideas to support collaboration. We have identified key topics
753 that put design concepts in the context of their potential use and discussed accompanying challenges.

754 In the workshops, we focused and limited the ideation process by introducing the topic of workspace awareness,
755 existing collaborative tools, and the timeline-based video editing software interface. Thus, design ideas are anchored
756 in existing video editing software, which is designed with a single user in mind. We suggest that further research on
757 collaborative video editing should investigate alternative ways of video representation in editing software and focus on
758 the evaluation of collaborative tools in a practical context.

759 REFERENCES

- 760 Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby. 1995. The user-centred iterative design of collaborative writing software. In
761 *Readings in Human-Computer Interaction*. Elsevier, 775–782.
- 762 Tom Bartindale, Guy Schofield, Clara Crivellaro, and Peter Wright. 2016. TryFilm: Situated Support for Interactive Media Productions. In *Proceedings*
763 *of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1412–1422.
764 <https://doi.org/10.1145/2818048.2819929> event-place: San Francisco, California, USA.
- 765 Tom Bartindale, Alia Sheikh, Nick Taylor, Peter Wright, and Patrick Olivier. 2012. StoryCrate: Tabletop Storyboarding for Live Film Production. In
766 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 169–178. <https://doi.org/10.1145/2207676.2207700> event-place: Austin, Texas, USA.
- 767 Michele Biasutti. 2015. Assessing a collaborative online environment for music composition. *Journal of Educational Technology & Society* 18, 3 (2015), 49.
- 768 J. Bligh, K. Jennings, and B. Tangney. 2005. *Designing interfaces for collaborative music composition*. Technical Report. Centre for.

- 781 Soon Hau Chua, Toni-Jan Keith Palma Monserrat, Dongwook Yoon, Juho Kim, and Shengdong Zhao. 2017. Korero: Facilitating Complex Referencing of
782 Visual Materials in Asynchronous Discussion Interface. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW (Dec. 2017), 34:1–34:19. <https://doi.org/10.1145/3134669>
783
- 784 Andy Cockburn and Steve Jones. 1995. Four principles of groupware design. *Interacting with Computers* 7, 2 (1995), 195–210.
- 785 Paul Dourish and Victoria Bellotti. 1992. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-*
786 *supported cooperative work*. ACM, 107–114.
- 787 Emilie Dumont and Bernard Mérialdo. 2008. Sequence alignment for redundancy removal in video rushes summarization. In *Proceedings of the 2nd ACM*
788 *TRECVideo Summarization Workshop (TVS '08)*. Association for Computing Machinery, New York, NY, USA, 55–59. <https://doi.org/10.1145/1463563.1463572>
- 789 Lisa S. Ede and Andrea A. Lunsford. 1990. *Singular texts/plural authors: Perspectives on collaborative writing*. SIU Press.
- 790 Sandy El Helou, Manolis Tzagarakis, Denis Gillet, Nikos Karacapilidis, and C. Yu Man. 2008. Participatory design for awareness features: Enhancing
791 interaction in communities of practice. In *Proceedings of the 6th International Conference on Networked Learning*. Issue: CONF.
- 792 Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. 1991. Groupware: some issues and experiences. *Commun. ACM* 34, 1 (Jan. 1991), 39–58. <https://doi.org/10.1145/99977.99987>
793
- 794 Robert S. Fish, Robert E. Kraut, and Mary DP Leland. 1988. Quilt: a collaborative tool for cooperative writing. In *ACM SIGOIS Bulletin*, Vol. 9. ACM, 30–37.
- 795 Benjamin Fonseca and Eurico Carrapatoso. 2006. Coview: A Cooperative Architecture for Digital Video Editing. *SMPTE Motion Imaging Journal* 115,
796 11-12 (Nov. 2006), 482–491. <https://doi.org/10.5594/J16132> Conference Name: SMPTE Motion Imaging Journal.
- 797 Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox. 2000. A
798 Semi-automatic Approach to Home Video Editing. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST*
799 *'00)*. ACM, New York, NY, USA, 81–89. <https://doi.org/10.1145/354401.354415>
- 800 Dan B Goldman, Brian Curless, David Salesin, and Steven M. Seitz. 2006. Schematic Storyboarding for Video Visualization and Editing. In *ACM SIGGRAPH*
801 *2006 Papers (SIGGRAPH '06)*. ACM, New York, NY, USA, 862–871. <https://doi.org/10.1145/1179352.1141967>
- 802 Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. 2008. Video object annotation, navigation, and composition. In
803 *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 3–12.
- 804 Jonathan Grudin. 1988. Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. In *Proceedings of the 1988 ACM*
805 *conference on Computer-supported cooperative work*. 85–93.
- 806 Jonathan Grudin. 1994. Groupware and social dynamics: Eight challenges for developers. *Commun. ACM* 37, 1 (1994), 92–105. Publisher: ACM New York,
807 NY, USA.
- 808 Carl Gutwin and Saul Greenberg. 2002. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative*
809 *Work (CSCW)* 11, 3-4 (Sept. 2002), 411–446. <https://doi.org/10.1023/A:1021271517844>
- 810 Claudia-Lavinia Ignat and Moira C. Norrie. 2006. Draw-together: Graphical Editor for Collaborative Drawing. In *Proceedings of the 2006 20th Anniversary*
811 *Conference on Computer Supported Cooperative Work (CSCW '06)*. ACM, New York, NY, USA, 269–278. <https://doi.org/10.1145/1180875.1180917>
- 812 Oskar Juhlin, Goranka Zoric, Arvid Engström, and Erika Reponen. 2014. Video interaction: a research agenda. *Personal and Ubiquitous Computing* 18, 3
813 (March 2014), 685–692. <https://doi.org/10.1007/s00779-013-0705-8>
- 814 Demetrios Karis, Daniel Wildman, and Amir Mané. 2016. Improving Remote Collaboration With Video Conferencing and Video Portals.
815 *Human-Computer Interaction* 31, 1 (Jan. 2016), 1–58. <https://doi.org/10.1080/07370024.2014.921506> Publisher: Taylor & Francis _eprint:
816 <https://doi.org/10.1080/07370024.2014.921506>.
- 817 Seungwon Kim, Gun A. Lee, Sangtae Ha, Nobuchika Sakata, and Mark Billinghurst. 2015. Automatically Freezing Live Video for Annotation During
818 Remote Collaboration. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*.
819 ACM, New York, NY, USA, 1669–1674. <https://doi.org/10.1145/2702613.2732838> event-place: Seoul, Republic of Korea.
- 820 Clemens N. Klokmoose, Christian Remy, Janus Bager Kristensen, Rolf Bagge, Michel Beaudouin-Lafon, and Wendy Mackay. 2019. Videostrates: Collaborative,
821 Distributed and Programmable Video Manipulation. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*
822 *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 233–247. <https://doi.org/10.1145/3332165.3347912>
- 823 Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.
- 824 Ida Larsen-Ledet and Henrik Korsgaard. 2019. Territorial Functioning in Collaborative Writing. *Computer Supported Cooperative Work (CSCW)* 28, 3 (June
825 2019), 391–433. <https://doi.org/10.1007/s10606-019-09359-8>
- 826 Wendy Mackay and Daniele Pagani. 1994. Video Mosaic: Laying out time in a physical space. In *Proceedings of the second ACM international conference on*
827 *Multimedia*. 165–172.
- 828 Allison Merz, Annie Hu, and Tracey Lin. 2018. ClipWorks: A Tangible Interface for Collaborative Video Editing. In *Proceedings of the 17th ACM Conference*
829 *on Interaction Design and Children (IDC '18)*. ACM, New York, NY, USA, 497–500. <https://doi.org/10.1145/3202185.3210758>
- 830 Boris Novikov and Oleg Proskurnin. 2003. Towards Collaborative Video Authoring. In *Advances in Databases and Information Systems (Lecture Notes in*
831 *Computer Science)*, Leonid Kalinichenko, Rainer Manthey, Bernhard Thalheim, and Uwe Wloka (Eds.). Springer Berlin Heidelberg, 370–384.
- 832 Pavel Okopnyi, Frode Guribye, Valentina Caruso, and Oskar Juhlin. 2021. Automation and redistribution of work: the impact of social distancing on live
833 TV production. *Human-Computer Interaction* 0, 0 (Nov. 2021), 1–23. <https://doi.org/10.1080/07370024.2021.1984917> Publisher: Taylor & Francis
_eprint: <https://doi.org/10.1080/07370024.2021.1984917>.

- 833 Pavel Okopnyi, Oskar Juhlin, and Frode Guribye. 2020. Unpacking Editorial Agreements in Collaborative Video Production. In *ACM International Conference*
834 *on Interactive Media Experiences (IMX '20)*. Association for Computing Machinery, New York, NY, USA, 117–126. <https://doi.org/10.1145/3391614.3393652>
- 835 Michael Quinn Patton. 2014. *Qualitative research & evaluation methods: Integrating theory and practice*. Sage publications.
- 836 Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2016. VidCrit: Video-based Asynchronous Video Review. In *Proceedings of the 29th*
837 *Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 517–528. <https://doi.org/10.1145/2984511.2984552>
event-place: Tokyo, Japan.
- 838 Charles Sanders Peirce and Justus Buchler. 1902. Logic as semiotic: The theory of signs.
- 839 Mark Roseman and Saul Greenberg. 1996. Building real-time groupware with GroupKit, a groupware toolkit. *ACM Transactions on Computer-Human*
840 *Interaction* 3, 1 (March 1996), 66–106. <https://doi.org/10.1145/226159.226162>
- 841 Stephanie Santosa, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. 2013. Direct space-time trajectory control for visual media editing. In
842 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1149–1158.
- 843 Kjeld Schmidt and Liam Bannon. 1992. Taking CSCW seriously. *Computer Supported Cooperative Work (CSCW)* 1, 1 (March 1992), 7–40. <https://doi.org/10.1007/BF00752449>
- 844 Naimdjon Takhirov and Fabien Duchateau. 2011. A cloud-based and social authoring tool for video. In *Proceedings of the 11th ACM symposium on*
845 *Document engineering (DocEng '11)*. Association for Computing Machinery, New York, NY, USA, 65–68. <https://doi.org/10.1145/2034691.2034705>
- 846 John C. Tang and Scott L. Minneman. 1991. VideoDraw: a video interface for collaborative drawing. *ACM Transactions on Information Systems (TOIS)* 9, 2
847 (1991), 170–184.
- 848 Stuart Taylor, Shahram Izadi, Kursat Ozenc, and Richard Harper. 2007. VideoPlay: Playful and Social Editing of Video using Tangible Objects and
849 Multi-touch Interaction. [https://www.microsoft.com/en-us/research/publication/videoplay-playful-and-social-editing-of-video-using-tangible-
850 objects-and-multi-touch-interaction/](https://www.microsoft.com/en-us/research/publication/videoplay-playful-and-social-editing-of-video-using-tangible-objects-and-multi-touch-interaction/)
- 851 Lucia Terrenghi, Torsten Fritsche, and Andreas Butz. 2008. Designing environments for collaborative video editing. (2008).
- 852 Jamie Zigelbaum, Michael S. Horn, Orit Shaer, and Robert JK Jacob. 2007. The tangible video editor: collaborative video editing with active tokens. In
853 *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM, 43–46.
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863
- 864
- 865
- 866
- 867
- 868
- 869
- 870
- 871
- 872
- 873
- 874
- 875
- 876
- 877
- 878
- 879
- 880
- 881
- 882
- 883
- 884