VidCrit: Video-Based Asynchronous Video Review

Amy Pavel[1], Dan B Goldman[2], Björn Hartmann[1], Maneesh Agrawala[3]

[1]University of California, Berkeley

{amypavel,bjoern}@cs.berkeley.edu

[2]Google

dgo@google.com

[3]Stanford University

maneesh@cs.stanford.edu

ABSTRACT

Video production is a collaborative process in which stakeholders regularly review drafts of the edited video to indicate problems and offer suggestions for improvement. Although practitioners prefer in-person feedback, most reviews are conducted asynchronously via email due to scheduling and location constraints. The use of this impoverished medium is challenging for both providers and consumers of feedback. We introduce VidCrit, a system for providing asynchronous feedback on drafts of edited video that incorporates favorable qualities of an in-person review. This system consists of two separate interfaces: (1) A feedback *recording interface* captures reviewers' spoken comments, mouse interactions, hand gestures and other physical reactions. (2) A feedback *viewing interface* transcribes and segments the recorded review into topical comments so that the video author can browse the review by either text or timelines. Our system features novel methods to automatically segment a long review session into topical text comments, and to label such comments with additional contextual information. We interviewed practitioners to inform a set of design guidelines for giving and receiving feedback, and based our system's design on these guidelines. Video reviewers using our system preferred our feedback recording interface over email for providing feedback due to the reduction in time and effort. In a fixed amount of time, reviewers provided 10.9 ($\sigma = 5.09$) more local comments than when using text. All video authors rated our feedback viewing interface preferable to receiving feedback via e-mail.

Author Keywords

Video review; video editing; feedback.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Video review is a key step of the video production pipeline in which stakeholders provide feedback on drafts of video projects. The reviewer's feedback can include comments that indicate problems in the video (e.g., "the text is too small"), offer suggestions (e.g., "crop the shot") or give compliments

(e.g., "I like the lighting in this shot"). In addition to commenting on such local issues, reviewers also give global critiques that pertain to the entire video (e.g., "In general, the music is too loud compared to the speech."). Video authors interpret and incorporate such feedback into subsequent drafts to improve the video.

In our formative study, we found smaller teams working on video projects typically give feedback informally, either verbally in-person or in asynchronously written text comments. Reviewers often prefer in-person feedback, because they can communicate their feedback more efficiently with speech than they can using text. For example, reviewers can easily communicate *temporal changes* (e.g. "move this shot here") by timing their spoken comments to events in the video and scrubbing the timeline to the locations they wish to change. Reviewers can indicate *spatial changes* (e.g. "move this text over to here") by gesturing over the source video with the mouse. In contrast, writing down temporal and spatial critiques can be tedious and time consuming, because reviewers first need to identify and transcribe timestamps, and then carefully describe proposed changes using text.

Video authors also prefer in-person critiques, because the context of comments can be more immediately understood. Authors can observe the reviewer's tone of voice and other non-verbal cues to better interpret verbal comments. For example, the comment "Come on, what is this mess?" could either be a joke or indicate a serious problem, depending on the reviewer's tone of voice. They can also engage in a dialog with the reviewer to come to a shared understanding of requested changes. However, written comments have the advantage of providing the author with a documented list of critiques. We interviewed four video authors who estimate that, even though they prefer in-person feedback, 80% of feedback occurs over e-mail because of scheduling and location constraints.

In this paper, we investigate how to preserve favorable qualities of both in-person and written text critiques, while allowing the feedback to occur asynchronously. By conducting formative interviews with practitioners, we develop a set of design guidelines for giving and receiving feedback. We then present VidCrit, a system that incorporates these guidelines into two interfaces; (1) a feedback *recording interface* (Figure 3) that lets reviewers efficiently capture their feedback, and (2) a feedback *viewing interface* (Figure 5) that lets video authors view this feedback later. The feedback recording interface captures the reviewer's spoken comments, facial expressions and hand gestures (using a video camera), the current playback time in the source video, interactions with the source video timeline (e.g. scrubbing, play, pause), and mouse gestures over the video player and transcript.

Our feedback viewing interface allows the video author to browse, skim and edit the feedback provided by the reviewer. The interface links the audio track of the reviewer's speech with the time-point the reviewer was watching in the original video, so that the video author can view the reviewer's comments in context. Authors can navigate these context-based comments using a timeline, or by browsing and skimming an automatically segmented list of transcribed comments. We automatically segment comments using a novel technique which achieves precision and recall of 0.836 and 0.835 respectively. We also automatically label each transcribed comment with several comment properties to help authors choose which comments to view. For instance, we mark which comments pertain to a short portion of the video (local comments) or the video as a whole (global comments).

We conducted a user study and found that, when reviewing 3 minute videos, reviewers provided 10.9 ($\sigma = 5.09$) more local comments using our interface than when using text. In a Likert-scale survey users compared the recording interface (5) to text (1) on a number of attributes. Overall, users preferred our interface ($\mu = 4.6, \sigma = 0.48$) for providing comments. We also asked both amateur and professional video authors to create a to-do list of edits using our feedback viewing interface. All video authors found our interface preferable to receiving feedback via e-mail.

RELATED WORK
Our VidCrit system allows video reviewers and authors to communicate feedback. It builds on prior work in three areas; (1) video logging and annotation, (2) feedback interfaces that support text comments, and (3) feedback interfaces that support speech comments.

Video annotation tools
Recording comments about a source video is closely related to video logging, the process in which practitioners watch and label a video according to its content. Video logging helps practitioners categorize video clips for later review, and cut down the amount of video they will need to re-watch. Early systems for logging videos addressed individually logging clips during the recording session [25], after the recording session [33], or as a group after the recording session [19, 20, 13]. Later work improved video logging usability and efficiency by creating fluid tablet interactions [29], allowing users to apply tags to keyframes [32], accommodating voice annotations [31], and segmenting videos at the per frame level [21]. Whereas previous work mainly addresses generalized tasks of assigning tags and writing notes on video clips, we support the specific task of video review, helping reviewers produce specialized annotations. Using the spoken word, reviewers can express nuances that are hard to convey in written text. By recording the position in the video as the user scrubs, our system allows reviewers to indicate points in the video timeline, without using timecodes. And by recording the mouse position, our system allows reviewers to indicate spatial regions of a video frame using mouse gestures instead of textual descriptions. For video authors, we support searching and browsing the feedback session using a variety of interactions.

Text-based asynchronous feedback interfaces
Marqueed [6] allows users to provide feedback on images via annotation tools and text, and provides support for discussion on each comment. Frame.io [3], ScreenLight [8] and Wipster [12] build on this idea but allow reviewers to draw and provide text comments on any single frame of a video. Users can navigate the source video by clicking a reviewer's text comments on the side of the player. However, because these interfaces link each comment to a single frame, they do not easily support delivering comments about a range of frames or about multiple discontinuous time ranges. Unlike VidCrit, these prior systems do not accommodate any spoken feedback.

Speech-based asynchronous feedback interfaces
UserTesting.com [11] and Silverback [9] are designed for practitioners conducting user studies on websites or software to record study sessions. These tools record a screencast of the task, the voice of the user, and the user's webcam video (Silverback [9] only) and present these feeds as one video segmented by task. VidCrit builds on this approach by transcribing and segmenting a reviewer's feedback by topical comments, allowing video authors to easily skim and browse comments. In the context of communicating feedback on PDF documents asynchronously, Yoon et al.'s RichReview [34] and RichReview++ [35] support voice comments supplemented with mouse gestures. The task of video review faces different design challenges than PDF review: Whereas RichReview implements spatial annotations via drawing, our system also records and replays temporal annotations such as scrubbing in the video. In addition, we consider how to allow users to view such temporal annotations and gain context for diectic comments by navigating synchronized source and feedback video timelines.

Cattelan et al. [17] also allow users to leave speech comments on a video so that non-collocated viewers can watch a TV show together. One viewer can leave comments on the video timeline while pausing the video, and another can only play back the comments in order by watching the video. Unlike Cattelan et al., our feedback recording interface lets video reviewers leave comments continously without pressing record and our feedback viewing interface lets video authors index those comments without watching the entire video.

Finally, FrameBench [2] lets reviewers and authors remotely collaborate synchronously using a shared video player. Due to scheduling constraints and bandwidth limitations, synchronous collaboration isn't always practical, so our work focuses on asynchronous collaboration instead.

FORMATIVE STUDY
Video workflows are well studied in the context of sharing and watching home videos [23], creating TV episodes [15, 14], and individuals editing videos [18, 16, 29]. However, no prior work investigates the process of small teams giving and receiving feedback for editing videos. To guide the design of our system for communicating such feedback during the editing process we asked practitioners: What are the benefits
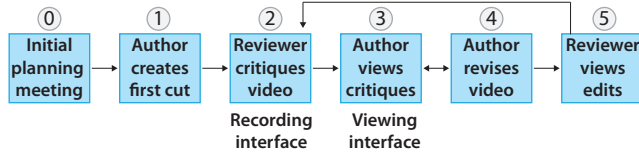
Figure 1. A typical editing workflow proceeds as follows: First, (0) the team meets synchronously to talk about the plan for the video and gathers the footage before or after this meeting. After this, (1) the video author (e.g. editor, producer/editor) creates the first cut, (2) a reviewer (e.g., client or executive producer) provides feedback, (3) the author views critiques, then (4) revises the video, and (5) the reviewer views the edits. The team then repeats steps 2-5 until both parties find the result satisfactory, or they reach a predetermined number of iterations. /danIn my experience, the termination criterion is when time runs out! Did you really find evidence that editors pick a predetermined number of iterations?

and drawbacks of the methods video editors currently use to communicate feedback?

We conducted semi-structured interviews with four video authors – three professionals and one amateur. These participants included an owner and producer for a small production company, a producer for a local TV news station, a video producer on staff for a university campus, and a university student who has created several research videos. We asked participants to describe their production workflow and compare experiences with asynchronous and synchronous feedback. All participants had received both types of feedback.

We found video editors and producers follow a common editing workflow (Figure 1). Although producers preferred to provide and receive such feedback in person, 80% of feedback occurs asynchronously through e-mail with text comments and timestamps. Note that although the practitioners we interviewed use text when communicating asynchronously, and voice when communicating synchronously, there exist methods for communicating voice feedback asynchronously (e.g. Watch-and-comment [17]), or for communicating text feedback synchronously (e.g. instant messaging). However, we compare the pros and cons of asynchronous text communication (e-mail review) and synchronous voice communication (in-person review).

Interviewees mentioned several benefits of in-person reviews:

- **Efficiency for the reviewer:** Composing a succinct and complete list of e-mail comments takes more time than watching and describing the changes in person.
- **Discussion and brainstorming:** In person, the author and reviewer discuss changes in the case of disagreements, and brainstorm alternatives to problems.
- **Body language and tone of voice**: Authors benefit from observing body language and tone of voice, as these convey affect of the reviewer's comments and initial reactions to the video draft. One author mentioned that he tries to "read between the lines" of text comments, because he cannot tell if the video works for the client through an e-mail list of critiques alone.
- **Additional comments**: Authors mentioned they received additional comments in person, including off-the-cuff re-
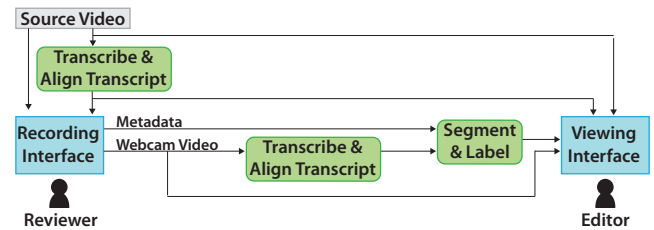

Figure 2. VidCrit takes a source video – and optionally, a source script – as input. If the reviewer does not provide a script, the system transcribes the video using rev.com [7], a crowd-based transcription service. VidCrit aligns the transcript or script to the source video. Then the reviewer uses the recording interface to record their feedback on the video, and the system transcribes and aligns the reviewer's webcam video. The system segments the session into comments by considering the interaction metadata, the transcript, and the aligned transcript timestamps. The author reviews the segmented feedback session and synced source video using the viewing interface.

actions and positive comments, that may not be conveyed in an e-mailed list of changes.

- **Trying alternatives:** Co-located editors and producers sometimes try alternative edits in person to reduce the number of iterations.

Interviewees also mentioned a few benefits of e-mail reviews:

- **Efficiency for the author:** E-mail comments from the reviewer provide an explicit to-do list of edits for the video author.
- **Accountability:** When the author finishes the list of documented changes, the author has completed their job and can resolve disputes by pointing to recorded e-mail comments. No such record exists for in-person reviews.

Through our interviews with practitioners, we identified a set of benefits of both asynchronous and synchronous feedback. Because synchronous feedback is often unfeasible, we build a system for asynchronous feedback that preserves many of the identified benefits. As we are focusing on building an asynchronous system, we do not provide features for synchronous discussion, brainstorming, and trying alternatives. Instead, we consider how to keep the process efficient for both the reviewer and the video author. In addition, we seek to let authors observe the reviewer's body language, tone of voice, and hear additional comments as they could in person.

INTERFACES

VidCrit (Figure 2) features two interfaces: one interface for reviewers to record their critiques on a video (the feedback *recording interface*) and another interface for video authors to view the critiques (the feedback *viewing interface*).

Feedback Recording interface

The recording interface (Figure 3) lets reviewers capture spoken feedback on a source video. The reviewer watches the video in the interface and delivers spoken critiques about the video much as they would in person. The interface records the reviewer's speech and their playhead location at all times, in order to capture the context of each comment. The interface also records the reviewer's webcam video (Figure 3E) so that the reviewer can supplement critiques with facial expressions and hand gestures.
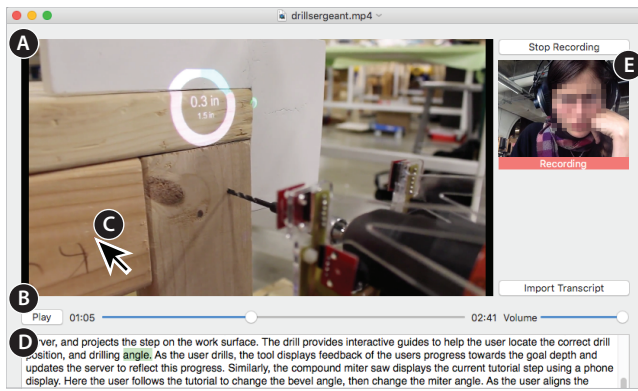
Figure 3. The feedback recording interface displays a video player for the source video (A) that the reviewers can play, pause, and scrub using the video timeline (B). Reviewers can mouse over the video to leave annotations (C). Reviewers can also import a source video transcript (D) using the "Import Transcript" button or start/stop recording using the "Start/Stop Recording" button. The recording interface also features a webcam camera view (E).

The reviewer can pause, play, and seek within the source video using the timeline (Figure 3B) to easily record temporal critiques. For example, the reviewer may watch the video until she finds a problem such as an incorrect ordering of two video segments. The reviewer can pause the video and say "Move this part from here, back to here" while scrubbing on the timeline to indicate the relevant locations in the video. The reviewer can also draw on the video player (Figure 3C) to describe spatial feedback.

For instance, if the reviewer notices distracting markings on a piece of wood, she can pause the video and say "We should remove the extra marks on the wood." while gesturing over the video player with the mouse, or scrubbing to other parts of the video to show additional points where distracting marks occur. Alternatively, the reviewer can include reactions as the video plays such as commenting "These are good images." while viewing a shot she likes.

The reviewer can optionally import a source video transcript (Figure 3D), which the system aligns to the video such that the transcript highlights the currently spoken word as the video progresses, as in prior work [16, 28]. Reviewers can use this transcript to navigate the source video by clicking on a word, and to provide feedback on wording or grammar in the source video script. For example, a reviewer can provide feedback by saying "I think instead of 'multiple tool tutorials' we should say, 'tutorials which leverage multiple tools'" while highlighting the corresponding section of the source video transcript.

Feedback Viewing Interface
After a reviewer finishes recording feedback with the recording interface, the video author can open a review session in the feedback viewing interface (Figure 5). The interface consists of two panes: (1) the *direct navigation pane* lets video authors watch and navigate the synchronized feedback session and source video, and (2) the *segmented comments pane* lets authors browse the feedback by segmented text critiques.
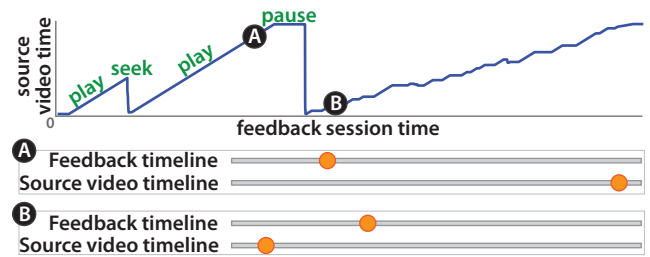

Figure 4. The source video time vs. feedback session time chart displays the reviewer's location in the source video during the feedback session. Here, the user plays the first part of the video, then seeks to the beginning to replay the whole video. Then, the user pauses the video and seeks to the beginning of the video. After this, the user slowly progresses through the video, playing, pausing, and seeking while giving comments. The timelines (A) and (B) show how the position of the source video timeline in the viewing interface updates depending on position of the feedback session timeline, according to where the reviewer positioned the source video at each time.

When reviewing feedback, the video author now has two pieces of time-based media to negotiate: In addition to the temporal order of the original source video under review, the feedback itself has its own temporal order, which may not relate to the original source video temporal order in any simple way (Figure 4). At one point, the video author may want to navigate using the temporal order of the source video, for example to address multiple comments about a single shot that may have occurred at different times in the feedback video. At another point, the video author may want to navigate using the temporal order of the feedback video, for example to understand a sequence of related comments that refer to different shots in the source video. The feedback viewing interface therefore features two timelines: the *feedback session timeline* and the *source video timeline*. Either timeline can be used to navigate, and navigation using these timelines can be linked or unlinked as described later in this section.

Direct navigation pane
The direct navigation pane shows the feedback session webcam video and the source video (Figure 5A). The author navigates the feedback session by playing, pausing, and scrubbing with the feedback session timeline (Figure 5B). When the author plays the feedback session, the reviewer's webcam video and speech comments advance linearly, while the source video and the source video timeline (Figure 5C) update to reflect the reviewer's position in the source video during the feedback session. For example, if, during the feedback session, the reviewer paused the source video to say "We should remove the extra marks on the wood", the source video will later pause at the same point when the video author replays the feedback session. Figure 4 shows an example how the synchronized timelines update as the video author plays back a feedback session.

Because the video reviewer often remains silent as they watch the video, a large portion of the feedback session does not contain feedback. To let the video author avoid silent portions of the feedback session, the feedback session timeline shows the duration of reviewer comments with colored rectangles. The color of these colored rectangles represents whether the comment addresses the beginning (light green), middle, or end (dark blue) of the source video (Figure 5B). Video au-
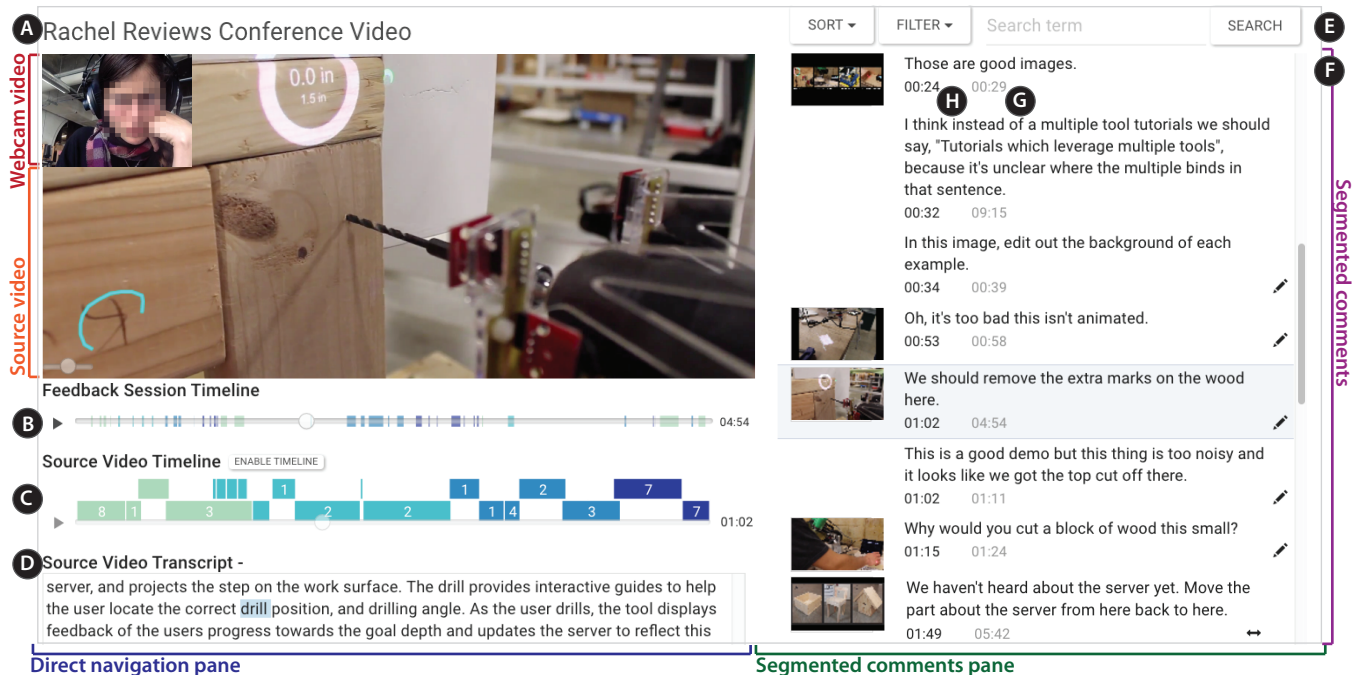
Figure 5. The VidCrit interface consists of a direct navigation pane for navigating the feedback session using the webcam and source video timelines, and a segmented comments pane for reviewing transcribed and segmented critiques. The direct navigation pane features the source and webcam videos along with a title (A), a feedback session timeline (B), a source video timeline (C) and the source video transcript (D). The segmented comments pane features sorting, filtering and search options (E), along with a list of segmented comments (F).

thors can see how the reviewer moved through the video by glancing at the comment colors. In Figure 5B, the reviewer watched the video two times while making comments then skipped around the video to make additional comments.

Video authors may also unlink these timelines with the "Unlink timelines" button and then navigate (e.g., play, pause or scrub) within the source video independently, while the feedback session video, audio and timeline remain paused. For instance, if the video author hears the reviewer say "Why would you cut a block of wood this small?" the author may need to rewatch the corresponding source video clip to gain context. The video author may view the section with the block of wood a second time by clicking "Unlink timelines" (Figure 5C), scrubbing back to the relevant location, then pressing play. Afterwards, the author can click "Link timelines" to continue browsing the synchronized feedback session and source video.

As in prior work [16, 28], the source video transcript (Figure 5D) highlights the currently spoken word as the source video progresses, enabling authors to gain context for comments. In addition, the source transcript displays the reviewer's transcript selections from the feedback session.

Segmented comments pane

The segmented comments pane allows authors to navigate the feedback session using transcribed and automatically segmented comments (Figure 5F) that each pertain to a single-issue critique. Each comment corresponds to a certain time in the feedback session (*feedback session time*) and a certain time in the reviewed source video (*source video time*). The *feedback session time* (Figure 5G) is the starting time of the

first word in the comment, and the *source video time* (Figure 5H) is the source video playhead location at the *feedback session time*. Clicking on a comment navigates to the corresponding times in the synchronized feedback session and source video. Pressing 'space' plays and pauses the synchronized feedback session and source video.

The shot-based timeline (Figure 5C) features colored bars that each represent a shot. We construct the shot-based timeline using the Edit Decision List (EDL) exported from the video author's editing software (Figure 6). The EDL file, a standard format exported from common video editors (e.g., Adobe Premiere, FinalCut), gives the start and end time of each shot within the final edit along with the filename of the footage. To build the shot-based source video timeline, editors import one EDL per video track in the project and we parse the EDL files to find each shot's in and out time, and filter out all audio or blank shots. As some video editing programs do not support EDL export, we also supply automatic shot detection at as a fallback [22, 26]. We include further discussion of the automatic shot detection in the Algorithms section. Authors can navigate to all comments about a shot of the source video by clicking on it in the shot-based timeline. This is useful when reviewers make several passes through the source video. For example, if the last shot in the video is less compelling, the video author may want to first view all critiques about that part of the video. The numbers in each colored bar (Figure 5C) represent the number of comments attributed to that shot. For instance, the last bar shows there are 7 comments on the last shot.
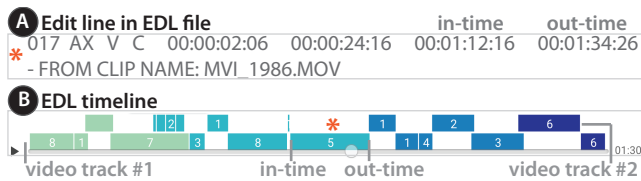
Figure 6. Each line in an EDL file (A) specifies a clip's properties along with its filename and in and out point in the final video file. Each segment in the EDL timeline (B) represents one line in an EDL file.
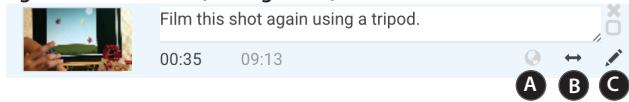

Figure 7. This figure shows a single comment in editing mode with three optional icons (A,B,C). The global icon (A) displays whether a comment relates to the entire video or just the given location. The scrubbing and annotation icons (B,C) display whether a comment has a corresponding scrubbing or annotation action.

We show one thumbnail per shot, grouping together consecutive comments that occur within the same source video shot. We show only one thumbnail per shot to avoid repetitive thumbnails. The comments (Figure 5F) also show icons that provide additional information about each comment: a pencil shows the comment contains a mouse gesture, the arrows show that the reviewer scrubbed to multiple times during the comment, and the globe shows that the comment pertains to the video as a whole rather than one specific location. As authors skim the segmented comments, the comment text and corresponding source video frame may be sufficient to understand some comments. A scrubbing or annotation icon indicates that the reviewer conveyed additional information that the author can only view by playing the comment.

As the author reviews the comments, she may hide irrelevant comments such as "Hmm, lets see." by deleting the comment, or hide comments she has already addressed by marking them as completed. When the video author hides a comment, the interface removes the corresponding mark on the feedback session timeline (Figure 5B) and decreases the comment count on the EDL timeline (Figure 6). Finally, authors can edit the text of long comments by entering edit mode (command-alt-click on a comment) (Figure 7). For example, an author might edit the second comment in Figure 5F down to *"Replace 'multiple tool tutorials' with 'tutorials which leverage multiple tools' "*, to make the comment easier to read on subsequent passes. In editing mode, the authors may also toggle the global/local icon (Figure 7A) or delete/complete a comment.

The author can sort, filter, and search through the segmented comments using the option bar (Figure 5E). The sort feature lets the author sort comments in various ways. Sorting by the feedback session time enable the author to see the comments chronologically. Sorting by the source video time allows her to see the comments grouped by shot in the source video (displayed). Or sorting by comment duration makes it possible to see the longest comments first, as they may take longer to understand and address than the shorter comments. The filter feature lets the author view comments that fit any one of the comment icon properties. For instance, the video author may

A. "Increase the brightness in this shot."
B. "Also, do not crop the speaker's forehead."
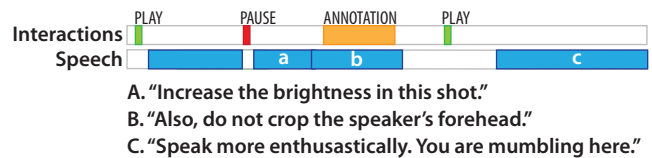C. "Speak more enthusastically. You are mumbling here."

Figure 8.   Comment (A) appears right after the reviewer pauses the video, indicating that they might begin a new comment. As the comment (B) begins with the word "also" and an annotation occurs slightly following the beginning it also may be a comment. Finally, comment (C) occurs after a long break between speech, so it may be the beginning of a new thought.

show or hide all global comments. The search feature lets authors search over the comment text, and the source video text (e.g. comments that pertain to a subsection in the source video where they say "system").

ALGORITHMS
Our system transcribes the source video and aligns the transcript to the video to enable transcript-based interactions. The system also transcribes, aligns, and segments the feedback session into single-issue comments so that video authors can browse and search the feedback session using text. We automatically assign three labels to each comment, and choose a preview thumbnail to provide authors with more information. Finally, we segment the source video into shots using Edit Decision Lists and automatic shot detection.

Transcribing and aligning videos
Aligned transcripts of the source and feedback videos are useful for reviewers and authors to skim for content of interest, index the video, and gain context for comments. Since video creators often write scripts before shooting and editing source videos, we use the original script where available. To transcribe feedback videos and unscripted source videos, we use rev.com [7] – a crowd-based transcription service that charges about $1 per minute of audio. Following prior work, we align the transcript to the source video by concatenating all text and the corresponding audio segments that are part of continuous speech [27]. We align each segment of continuous speech with its corresponding audio segment, or an existing script with the entire audio track, using forced alignment between phonemes and audio features, as in the work of Rubin et al. [30]. Reviewers can then navigate using the transcript and annotate the transcript (Figure 3), while video authors can search and browse reviewer comments using text (Figure 5F).

Segmenting comments
After transcribing and aligning the feedback session transcript, our system segments the feedback session into topical comments so that the video author can quickly skim and browse the feedback. We use the text transcript, the alignment timestamps and the interaction metadata from the feedback session as inputs to the segmentation algorithm.

Segmentation Algorithm
Our system transcribes comments into single issues so that authors can manipulate, browse, and play back comments in issue-based chunks, much as they would items in a to-do list. In the dataset we collected, almost all sentences contain a single issue (a few contain more than one issue). So our system

first generates possible issue segment boundaries as the beginning time for the first word in each sentence. Since our system's punctuation comes from a crowd transcription site, some transcribers may not include a period if a user trails off without finishing a thought. Therefore, our system adds all words that occur after a pause in speech as potential segment boundaries.

Given a set of potential segment boundaries, we classify whether each one is true issue boundary using an SVM in combination with a Random Forest Classifier. The classification relies on the video reviewer's interactions recorded during the review session, and the aligned transcript of the feedback video. Our system calculates the following feature sets for each potential segment boundary to help determine whether it is a true segment boundary:

- *Interaction proximity*: Proximity to each type of interaction (pause, play, seek, gesture) before and after the potential segment boundary
- *Word proximity*: Proximity to spoken words immediately before and after the potential segment boundary
- *Segment length*: Includes segment duration (time from beginning of first word to end of last word), and the word count of the segment
- *First word*: The first word of the potential segment
- *Punctuation*: The punctuation of the prior segment

We chose these features by observing different ways that reviewers conversationally segment their comments. For example, we include interaction proximity because reviewers often pause the video before they start a new comment, play after finishing a comment, and seek or gesture briefly after starting a comment to convey temporal or spatial critiques (Figure 8A). We include word proximity because reviewers often pause speaking between comments (Figure 8C). The segment length features help us identify very short potential segments such as "Okay.", "Yeah.", "Umm, let's see." or "Mmhmm," which often contain only verbalized pauses, and such standalone expressions represent new segments. We include the first word of the current segment because the first word may be a transition word that can be used to start a new thought (Figure 8B). Finally, we include punctuation because a pause (no punctuation) is more likely to indicate hesitation rather than finishing a thought.

We use a Linear SVM with L1 penalty, a common penalty for feature selection, to remove unimportant features (i.e. features with a coefficient of zero). The *first word* feature set generates one feature for each word that starts a potential segment. As a result, the feature selection step typically removes first word features that do not correspond to any positive examples. Then, we use a Random Forest Classifier with 3000 estimators and a maximum depth of 4 to train the classifier only using the selected features. The training and testing sets are described in the evaluation section below.

Evaluation

To evaluate our comment segmentation technique, we collected short videos in a variety of domains, including a cook-
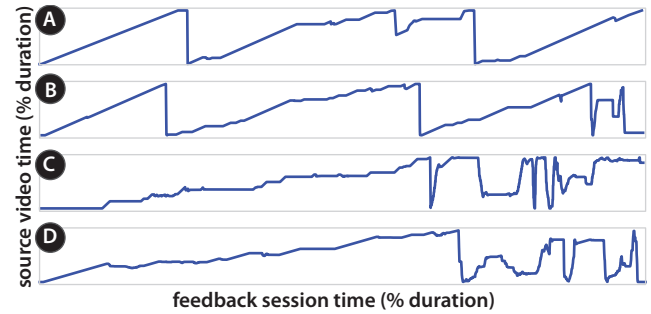

Figure 9. This figure shows the reviewer's time in the source video vs the time in the feedback session for 4 different reviewers (A,B,C,D). In the first examples (A and B), reviewers watch the video multiple times while playing, pausing, and scrubbing to leave comments after the first time. In the second examples (C and D), reviewers pause and scrub to leave comments on their first pass. At the end of the session, they give general comments while scrubbing through the video to find examples.

ing tutorial, a Kickstarter video, two technology demonstration videos, and a travel video log which were an average of 3.01 minutes long ($\sigma = 19.5s$). We invited 8 people who had prior experience creating and giving feedback on video projects to provide feedback on one video each. Recorded feedback sessions averaged 12.4 minutes long ($\sigma = 1.61min$) where the subject spent an average of 3.49 minutes speaking ($\sigma = 42.4s$) and the remainder of the time listening and watching the source video or scrubbing through it. We manually segmented the resulting feedback sessions to create ground truth segmentations.

We observed reviewers utilizing various strategies for providing feedback on the videos (Figure 9). Some watched the video all the way through silently, then watched again while playing, pausing, scrubbing, and leaving comments multiple times (Figure 9A,B). Others paused and scrubbed during the first watch-through to add comments as they went, and at the end left general comments, scrubbing through the entire video to show examples (Figure 9C,D).

Using leave-one-out cross-validation with this dataset we found that our classifier identifies true boundaries with a precision, recall and F1-score of 0.836, 0.835, and 0.834 respectively (Table 2). In practice, a missed segmentation results in a longer comment segment to watch with little effect on the authors experience. A false positive segment boundary results in two segments that are topically the same. When leaving out individual feature sets, we find that leaving out word proximity results in the largest difference in F1-score. However, even without word proximity, the classifier performs much better than random (F1-score=0.633 vs F1-score=0.475).

Determining comment attributes

VidCrit assigns each comment three attributes with corresponding icons (Figure 7A-C) which let the editor know if the reviewer left additional information with the comment, or if the comment pertains to a local or video-wide critique. VidCrit also selects a thumbnail for each comment.

Global icon: We show the global icon (Figure 7A) if the words "in general", "as a whole", or "overall" appear in a segment as reviewers in our dataset used such language to describe general comments applying to the entire video. We

Video title	Subject	Time (min)	Feedback time (min)	Total comments	# Comments labelled		
					Annotation	Scrubbing	Global
Erin's famous fruit pizza [1]	cooking tutorial	2.87	11.98	36	4	23	4
LearnKendama.com kickstarter video [5]	product promotion	3.30	11.18	32	10	15	1
Interface demo video (unpublished)	interface walkthrough	2.73	13.52	34	13	4	0
Skintillates video [10]	project teaser video	1.00	10.38	29	14	12	1
Fabrication project video (unpublished)	project demonstration	2.68	13.57	60	26	9	2
France travel log [4]	travel	3.48	10.62	24	5	8	1

Table 1. Information for result videos. We segmented and labelled all videos automatically. Total comments refers to the number automatically produced segments. The number of comments containing an annotation, scrubbing, or global critique was also determined automatically.

	Precision	Recall	F1-score
Ours	0.836	0.835	0.834
No interaction proximity	0.840	0.834	0.833
No word proximity	0.664	0.636	0.633
No segment length	0.829	0.827	0.827
No first word	0.831	0.829	0.827
No punctuation	0.836	0.833	0.832
Random	0.486	0.471	0.475

Table 2. Comparison between our segment boundary selection, leaving out feature sets, and randomly selecting boundaries. Random represents a function that randomly assigns a boundary or not weighted by the number of boundary occurrences in the training data.

created ground truth data for all of the videos in our dataset. This method achieves a much higher precision than recall for labelling global examples (0.80 and 0.33 respectively). That is, comments labelled "global" are likely correctly labelled, but this method misses many global comments. The method misses many such comments because global comments do not necessarily need to contain the specific words "in general", "as a whole" or "overall" to convey a comment about the entire video (e.g. "the audio quality is poor"). We would prefer to use a more robust n-gram classification based approach, but global comments are rare within the dataset we collected. In the future, we will collect more data to enable better classification.

Scrubbing icon: We show the scrubbing icon (Figure 7B) if the reviewer scrubbed within the video timeline at any point during the comment.

Annotation icon: The annotation icon (Figure 7C) displays whether the reviewer performed a spatial annotation within the comment. We record annotations as mouse movements over the video player, but many of these movements are unimportant or accidental. For example, users often mouse over the video player incidentally while navigating the timeline, which lies immediately below it. To filter out such unintentional movements, our system first deletes any mouse event that occurs at the lower 10% of the video player pane. Next, it groups together continuous streams of mouse movements (less than 0.1 second apart) and computes the duration of each resulting segment. Our system shows the spatial annotation icon for any comment with a total mouse-movement segment duration over 0.3 seconds. We created ground truth data for three videos by labelling comments that contained meaningful annotations as true, annotation and comments that contained no annotations or accidental annotations as false annotations. Our approach achieves an annotation-labelling accuracy of 92%, whereas counting any mouse movement as a true annotation achieves an accuracy of 80%.

Thumbnail: We choose the thumbnail time as the start time of the first word in a comment unless a spatial annotation occurs. If such an annotation occurs, we choose the starting frame of the longest annotation. We hide the comment thumbnail if the comment follows another comment in the same shot. We describe our shot segmentation methods in the "Shot-based source video timeline" section.

Shot-based source video timeline

We provide a shot-based source video timeline, and comment grouping to allow authors to browse comments by source video section. We create this timeline based on edit decision lists (as described in Interfaces) when avaiable. When editors do not provide EDL files, VidCrit relies on a automatic shot detection based on color histograms to segment the source video into shots [22, 26]. In our implementation, we calculate color histograms for every frame, then compute the Chebyshev distance between each pair of histograms. We smooth the distance signal by convolving it with a Hanning filter, then identify large color histogram changes by finding all local maxima. However, not all local maxima correspond to shot changes as some may indicate smooth changes in lighting or camera angle. So, we find the local maxima with the sharpest peaks which indicate an abrupt color change by comparing each local maxima to the point before it, and only count the local maxima as a shot change if the difference is greater than 0.5 (we determined this threshold empirically based on the videos in our dataset).

RESULTS

Figure 10 shows feedback sessions automatically segmented into critiques and labelled using our system. We generated these results using a set of videos in a variety of domains (see Table 1). We invited reviewers with experience creating videos and giving feedback on videos to use the recording interface to critique a video. To guide reviewer feedback, we gave the reviewers a set of goals for the video (e.g. convince people to financially back the project, get the general public excited about your research project). Reviewers spent an average of 11.88 ($\sigma = 1.28$) minutes delivering feedback, and produced an average of 36 ($\sigma = 11$) segmented comments. On average, 12 ($\sigma = 7$) comments contained scrubbing and 12 ($\sigma = 6$) comments contained an annotation, while only 1.5 ($\sigma = 1.3$) comments contained a global comment. These results suggest the reviewers used the annotation and scrubbing capabilities not afforded by text.

USER EVALUATIONS

We conducted a user study with our recording interface and an informal evaluation with our viewing interface.
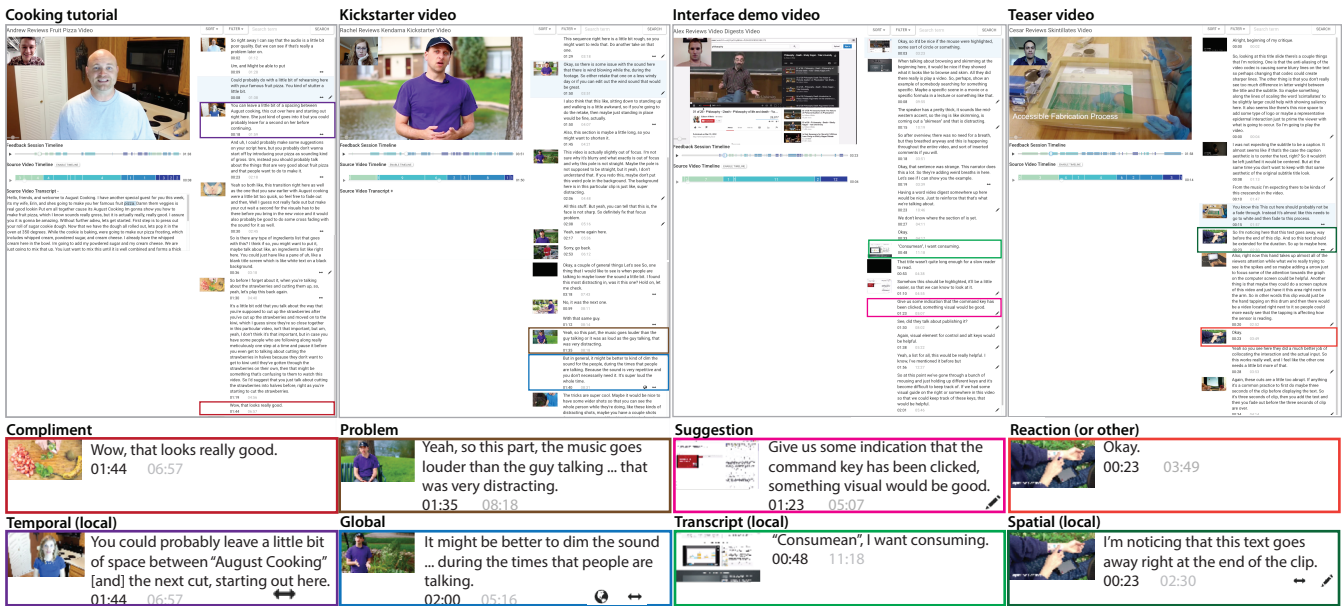
Figure 10. Users critiqued four videos on a variety of topics. We transcribed these feedback sessions using rev.com [7], then our system automatically segmented the sessions into comments and labelled them. Users were able to produce a variety of critiques using our system including different critique styles (e.g. compliment, problem, suggestion) as well as critiques with different content (e.g. temporal, spatial, transcript and global).

Recording interface user study

To find out if video reviewers communicate feedback more efficiently using our interface than using text, we conducted a user study with 8 participants who had prior experience both providing and receiving feedback on video projects. Our system focuses on asynchronous feedback, so we leave a comparison of our interface to synchronous, in-person feedback for future work.

Method

We selected two videos which contained content designed for a general audience (a Kickstarter video and a cooking tutorial), which were 3.32 minutes and 2.85 minutes long. We picked these videos because they represent different domains, production styles, and levels of formality.

We recruited 8 participants (6 males and 2 females, ages 23-31, graduate students) who had prior experience giving and receiving feedback on videos via school mailing lists. We allowed each participant 11 minutes to provide feedback on one video using text and 11 minutes to provide feedback on the other video using our recording interface. No participants had seen either video before the study. Between participants, we varied which video went with which feedback method, the order of the videos, and the order of the feedback methods. Two participants completed each of 4 possible orderings.

Before the participants provided feedback, we gave each participant the communication goals of each video, a list of types of feedback they might provide (e.g., spatial, temporal, transcript, overall), and a one minute overview of the recording interface. We concluded each study session with Likert scale questions and a semi-structured interview about advantages and disadvantages of each method.

Recording Interface Study Results

We transcribed all recorded feedback sessions using the transcription service rev.com [7], and segmented the comments into single issues by hand. Because a comment may contain specific, substantive advice, or only an off-topic anecdote, we tagged each comment with the types of critique that it contained. The first author tagged all of the response text, making sure to first randomize the responses across subjects and hide identifying information in the response text. In particular, with the conditions of the comments hidden, we labeled whether each comment contained a specific problem, an actionable suggestion, a compliment and/or a reaction to the video that did not fit under the other categories (e.g., "Wait, what?", "Interesting..."). Because our system supports efficiently referencing specific times and visuals in the video, we also labeled whether each comment discussed the video as a whole (i.e. global comment), or a single point in the video (i.e. local comment).

We found that all users produced more comments with our interface than when using text (Figure 11). On average users produced 21.9 critiques ($\sigma = 9.06$) using our interface and 10.4 ($\sigma = 4.15$) using text. A Wilcoxon Signed-rank test shows that the difference is significant ($W = 36, p < 0.05$). In addition, users of our interface produced 15.4 ($\sigma = 5.29$) comments containing specific problems compared to 7.5 ($\sigma = 4.36$) with text ($W = 28, p < 0.05$). Users also produced significantly more comments containing actionable suggestions when using our interface ($\mu = 11.9, \sigma = 4.94$) than when using text ($\mu = 5.38, \sigma = 2.64$) ($W = 28, p < 0.05$). However, users did not produce significantly more compliments or reactions. As users did not produce more global critiques, the increase in comments with our interface resulted from an increase in local critiques ($W = 36, p < 0.05$).
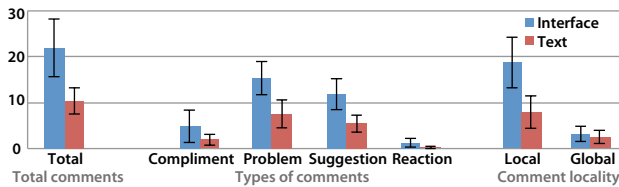
Figure 11. Average number of critiques produced using the interface and text total. Then average number of critiques produced using the interface and text which were assigned each label. Multiple labels may be assigned to one critique. Error bars show the 95% confidence interval.

Overall, users reported that they found the recording interface to be more efficient and preferable to text for recording comments (Figure 12). In an interview, users mentioned the following advantages of the interface: overall efficiency (7 users), scrubbing instead of transcribing timestamps (6 users), providing comments without pausing the video (4 users), pointing to visual changes instead of writing them down (4 users), referencing the source video transcript (3 users). The user who did not mention efficiency mentioned that using the interface felt more conversational than text. Though users preferred our interface for most types of critiques, our interface received the lowest ratings for delivering global feedback. This is unsurprising because the recording interface does not provide any specialized support for global comments. When asked about advantages of text, 5 users mentioned they could go back and edit their feedback (e.g. to rearrange their comments into categories), and 2 users mentioned they felt they had more time to think through what they were going to say when using text.

Viewing interface informal evaluation

To find out if video authors could use the viewing interface to quickly understand and review feedback on videos, we conducted an informal evaluation. Using school mailing lists, we recruited 4 participants (2 female, 2 male, ages 22-45) with experience receiving feedback on videos in-person and via e-mail. In particular, we recruited two professional video producers (U1,U2), and two participants currently working on video projects (U3, U4).

Method

Ahead of time, we collected videos from the two participants working on video projects, U3 and U4 (e.g. a 3 minute conference video, and a 1 minute project teaser). We also collected feedback on each project from one member of the project team using our feedback recording interface. During the informal evaluation, each participant used the feedback viewing interface to view, interpret, and judiciously filter the reviewer's suggestions into a list of changes to make on the next round of edits. U3 and U4 reviewed the feedback provided on their own video, while the two professional video editors (U1 and U2) each reviewed the feedback provided on one of these videos. We asked U1 and U2 to view the existing video ahead of time, and we told them the context of the feedback (e.g. a peer on the video team provided feedback on the editing and story of the video). After users completed the task, the users answered interview questions and Likert scale questions comparing receiving feedback with the viewing interface to prior experience receiving feedback over e-mail and in person.
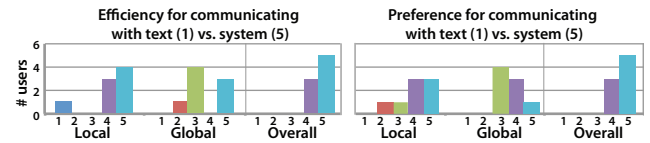

Figure 12. Users compare the interface and text for making a variety of comments.

Results: How users interacted with the feedback

All users were able to use the interface to review, interpret and filter the critiques left by the reviewer. Three users primarily navigated comments while they were sorted in the source video sort method (Table 3). All users read segmented comments, sometimes navigating to the corresponding frame in the source and feedback video by clicking on the comment. While most users (U1,U2,U3) only played comments that they didn't understand without context, U4 played all comments on the first pass, and read them on subsequent passes. When users didn't understand a comment after playing it, they either used the source timeline (U2, U3) or feedback session timeline (U4) to navigate to an earlier point.

Results: Interview and Likert scale questions

All users enthusiastically stated they would use the interface to receive feedback asynchronously. In a Likert scale survey, all users reported that they preferred our interface to e-mail, and that they found our interface to be more efficient than e-mail (Figure 13).

In the interview, all users mentioned that they preferred our interface to e-mail because comments were easier to contextualize as you could see the reviewer's playhead position while they gave the comment. Users mentioned that with e-mail, unlike our interface, they find critiques to be confusing because the context is often unclear (U1, U3, U4), and they find looking up timestamps to be tedious (U2). In addition, all users said they liked that it was easy to find more context if they needed to by navigating the source video (U1,U2,U3), viewing the source video transcript (U3), or scrubbing in the feedback timeline (U4). Users also liked being able to gauge the reviewers reaction using their tone of voice (U2, U4), the webcam video (U2), and extra reaction comments and hesitations (U2). U2, who previously taught video production, mentioned our interface was "the closest to in-person feedback I've seen", and mentioned it would be great to use in online classes. U1 mentioned he would be particularly interested in using our interface instead of e-mail in cases where there are multiple similarly-skilled creative partners (e.g. a producer and an editor), or a particularly invested client (cases where everyone "deserves" to give detailed feedback).

Unsurprisingly, some users found in-person feedback to be more efficient and easier to use than our system (Figure 13). All users mentioned that with our interface, unlike in-person feedback, they didn't get an opportunity to immediately resolve disagreements about changes. However, users identified benefits of our system compared to in-person feedback including the documentation of requested changes with their timestamps (U2, U3, U4), and the capability to sort and browse feedback (U2, U3). U3 told us that having your work

		U1	U2	U3	U4
Navigation	Feedback timeline	0	0	0	1
	Source timeline	0	1	5	0
	Shot-based timeline	0	4	3	0
	Segmented comments	2	32	84	211
Sort (% of time)	Feedback session	100%	5%	3%	1%
	Source time	0	73%	97%	98%
	Duration	0	22%	0	1%
Editing	Delete	2	8	19	39
	Edit text	0	0	0	19
	Mark complete	0	0	0	19
	Mark local/global	0	1	0	5
Search		0	0	1	0
Total time		7	12	14	19

Table 3. Interactions during the user study included users navigating the feedback session, sorting the feedback session comments, editing the comments, searching within the comments. Users primarily used the segmented comments for navigation, and most reviewed the comments sorted by source time (default sort is feedback session time). While all users deleted comments, only one user edited the original text of the comments.

criticized in person can be stressful, whereas she found viewing the feedback after the fact to be less stressful.

We asked users about what they would like to change about the interface and all users mentioned additional tagging capabilities. Specifically, users mentioned that they would like to assign tags based on priority of tasks, mark comments that need further discussion, or type of change required to fix the problem (e.g., reshoot, audio/video quality, b-roll). U1 and U2 also mentioned that they prefer to preserve reviewers comments rather than edit them. Instead, they would like the ability to write notes along with each comment.

LIMITATIONS AND FUTURE WORK

Our current implementation of VidCrit has some limitations.

Reviewers can't view or edit their comments: Like an in-person review, our current system does not support reviewers editing their own feedback video after it is recorded. In the recording interface study, reviewers pointed out that unlike in-person reviews, the feedback video can be replayed at a later time. Thus, reviewers wanted to remove mistakes, and ensure they do not repeat comments. With an accurate real-time transcription of the review session, reviewers could view their feedback and edit the review session via the text transcript, similar to prior work [30, 16, 34, 35].

Receiving comments from multiple reviewers: Our system addresses one common case in which the video author receives feedback from one reviewer at a time. However, if multiple reviewers provide feedback in parallel, video authors would need to look at each reviewers feedback in a separate interface instance. In the future, we could distinguish comments from different reviewers in the viewing interface and support multiple feedback session timelines. With more than a few reviewers, we could provide an aggregate view of categorized comments, similar to prior work [24].

Synchronous use of our system: In this paper we focus on the problem of asynchronous review as synchronous review is not always possible. However users noted that our system provides benefits over in-person feedback in that it lets au-
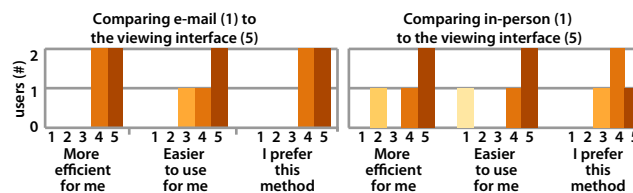

Figure 13. Users compared our viewing interface to existing methods of providing feedback. All users preferred our system overall when compared to e-mail.

thors search, browse and skim the documented feedback. In the future we will use our system to document synchronous conversations.

CONCLUSION

In this paper we present VidCrit , a new system for commenting and receiving comments on video projects. Our system allows users to deliver comments asynchronously while preserving benefits of both in-person and digital text critiques. We found users provide more actionable suggestions using our recording interface. In an informal evaluation, users successfully used our viewing interface to review editor feedback and reported they preferred the interface over existing methods.

REFERENCES

1. Erin's famous fruit pizza. **https://youtu.be/ORbZ6jzJLIA**. Accessed 2016-03-09.

2. Framebench. **http://www.framebench.com/video-collaboration/**. Accessed 2015-05-31.

3. Frame.io. **http://frame.io/**. Accessed 2015-05-31.

4. France vlog — annecy. **http://www.leoniesii.com/annecy-france/**. Accessed 2015-05-31.

5. Learnkendama.com kickstarter video. **https://youtu.be/MkkuCJ1l2K8**. Accessed 2016-03-09.

6. Marqueed. **https://www.marqueed.com/**. Accessed 2016-03-09.

7. rev.com. **https://www.rev.com/**. Accessed 2016-03-21.

8. Screenlight. **https://screenlight.tv/features/**. Accessed 2015-05-31.

9. Silverback. **https://silverbackapp.com/**. Accessed 2016-03-21.

10. Skintillates video. **https://vimeo.com/165809373**. Accessed 2016-07-20.

11. Usertesting. **https://www.usertesting.com/product/videos-and-metrics**. Accessed 2016-03-21.

12. Wipster. **http://wipster.io/**. Accessed 2015-05-31.

13. Abowd, G. D., Gauger, M., and Lachenmann, A. The family video archive: an annotation and browsing environment for home movies. In *Proc. MM'03*, ACM (2003), 1–8.

14. Bartindale, T., Schofield, G., and Wright, P. C. Tryfilm: Situated support for interactive media productions. In *Proc. CSCW'16*, ACM (2016), 1410–1420.

15. Bartindale, T., Sheikh, A., Taylor, N., Wright, P., and Olivier, P. Storycrate: tabletop storyboarding for live film production. In *Proc. CHI'12*, ACM (2012), 169–178.

16. Berthouzoz, F., Li, W., and Agrawala, M. Tools for placing cuts and transitions in interview video. *ACM TOG'12 31*, 4 (2012), 67.

17. Cattelan, R. G., Teixeira, C., Goularte, R., and Pimentel, M. D. G. C. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM TOMM 4*, 4 (2008), 28.

18. Chi, P.-Y., Liu, J., Linder, J., Dontcheva, M., Li, W., and Hartmann, B. Democut: generating concise instructional videos for physical demonstrations. In *Proc. UIST'13*, ACM (2013), 141–150.

19. Cockburn, A., and Dale, T. Ceva: a tool for collaborative video analysis. In *Proc. SIGGROUP'97*, ACM (1997), 47–55.

20. Cohen, J., Withgott, M., and Piernot, P. Logjam: a tangible multi-person interface for video logging. In *Proc. CHI'99*, ACM (1999), 128–135.

21. Diakopoulos, N., and Essa, I. Videotater: an approach for pen-based digital video segmentation and tagging. In *Proc. UIST'06*, ACM (2006), 221–224.

22. Gargi, U., Kasturi, R., and Strayer, S. H. Performance characterization of video-shot-change detection methods. *Circuits and Systems for Video Technology 10*, 1 (2000), 1–13.

23. Kirk, D., Sellen, A., Harper, R., and Wood, K. Understanding videowork. In *Proc. CHI'07*, ACM (2007), 61–70.

24. Luther, K., Tolentino, J.-L., Wu, W., Pavel, A., Bailey, B. P., Agrawala, M., Hartmann, B., and Dow, S. P. Structuring, aggregating, and evaluating crowdsourced design critique. In *Proc. CSCW'15*, ACM (2015), 473–485.

25. Mackay, W. E. Eva: An experimental video annotator for symbolic analysis of video data. *ACM SIGCHI Bulletin 21*, 2 (1989), 68–71.

26. Mas, J., and Fernandez, G. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID'03* (2003).

27. Pavel, A., Goldman, D. B., Hartmann, B., and Agrawala, M. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proc. UIST'15*, ACM (2015), 181–190.

28. Pavel, A., Reed, C., Hartmann, B., and Agrawala, M. Video digests: a browsable, skimmable format for informational lecture videos. In *Proc. UIST'14*, ACM (2014), 573–582.

29. Ramos, G., and Balakrishnan, R. Fluid interaction techniques for the control and annotation of digital video. In *Proc. UIST'03*, ACM (2003), 105–114.

30. Rubin, S., Berthouzoz, F., Mysore, G. J., Li, W., and Agrawala, M. Content-based tools for editing audio stories. In *Proc. UIST'13*, ACM (2013), 113–122.

31. Truong, A., Berthouzoz, F., Li, W., and Agrawala, M. Quickcut: An interactive tool for editing narrated video. In *Proc. UIST'16*, ACM (2016), To Appear.

32. Volkmer, T., Smith, J. R., and Natsev, A. P. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *Proc. MM'05*, ACM (2005), 892–901.

33. Weber, K., and Poon, A. Marquee: A tool for real-time video logging. In *Proc. CHI'94*, ACM (1994), 58–64.

34. Yoon, D., Chen, N., Guimbretière, F., and Sellen, A. Richreview: blending ink, speech, and gesture to support collaborative document review. In *Proc. UIST'14*, ACM (2014), 481–490.

35. Yoon, D., Chen, N., Randles, B., Cheatle, A., Löckenhoff, C. E., Jackson, S. J., Sellen, A., and Guimbretière, F. Richreview++: Deployment of a collaborative multi-modal annotation system for instructor feedback and peer discussion. In *Proc. CSCW'16*, ACM (2016), 195–205.