A Tool for Navigating and Editing 360 Video of Social Conversations into Shareable Highlights

Anh Truong*
Stanford University, Adobe Research

Maneesh Agrawala†
Stanford University

Abstract

We present ConvCut, an interactive tool for efficiently navigating and editing 360 video of social conversations into shareable video highlights of the memorable moments. ConvCut starts by obtaining a high-quality transcript of the conversation and uses it to segment the video into one 360 video clip per line of speech. It then applies audio, video and text analysis to label the clips with information including the spatial location of faces, the current speaker, the topics of conversation, instances of laughter and extreme changes in volume, facial expression, or gestural motion. The resulting structure lets users navigate the video using keyword search over the transcript and labels to quickly find memorable moments. Users can mark the lines corresponding to these moments and ConvCut edits together the corresponding video clips, automatically choosing a regular field of view (RFOV) framing that emphasizes the speaker of each line. If desired, users can modify the automatic edit to include alternative framings or reactions from others in the group. We demonstrate that with ConvCut , first-time users can easily edit long social conversations (25-60 min) into short highlight videos (0.27-2 min) and share them with others. The resulting highlights include jokes, reactions to pranks, funny stories and interactions with children.

Keywords: 360 Video; Video Editing; Social Conversations; Video Highlights.

Index Terms: H.5.2 [User Interfaces]: User Interfaces—Graphical user interfaces (GUI); H.5.m [Information Interfaces and Presentation]: Miscellaneous

1 Introduction

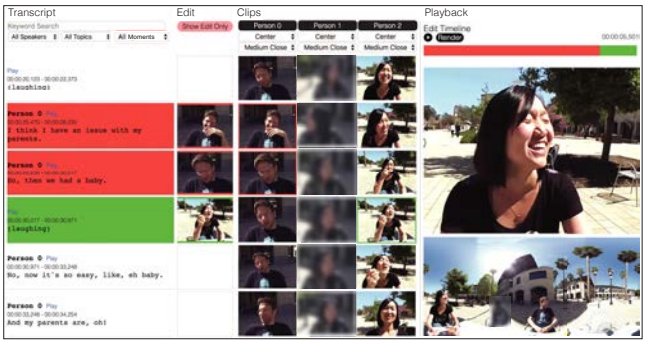
People partake in social conversations with friends and family every day. These conversations enable participants to connect with each other, and often include memorable moments such as, jokes and physical humor, emotional accounts of life events, riveting gossip, and intellectual discussions. Video clips of such conversations provide an increasingly popular way to share and relive the memorable moments with friends and family via media sharing tools or posts to social media. Twitch contains thousands of channels dedicated to live-streaming conversations and YouTube contains many compilations of memorable clips from these conversations.

But capturing such video clips is challenging. People typically forget to capture events in the moment and only pull out their cameras reactively, after something memorable happens. For example, parents often cannot anticipate when a child will say or do something cute. Moreover, even if the camera is recording the moment, the person holding the camera is usually outside the frame turning her into an observer rather than a participant. It changes the nature of the conversation.

Consumer 360 cameras offer an alternative for capturing social conversations that alleviates these two difficulties. Such cameras are small enough to place on a central table (e.g. dining table, coffee table, countertop) and can be set to record the entire conversation between all of the surrounding participants. In this setting, the challenge is to find memorable moments within the resulting 360 video and edit them into short clips that convey the back and forth pattern of the conversation.

We present ConvCut, a transcript-based video editing tool that facilitates navigating and editing shareable highlights from social conversations. Given a 360 conversation video as input, we obtain a high-quality transcript of the speech and time-align it with the video. ConvCut then breaks the video into segments corresponding to each line of speech in the transcript. It uses audio, video and text analysis techniques to label each segment with structural information, such as the spatial location of the faces in the frame, whether each face is the current speaker and the topics covered in the conversation.

(1) capture setup



(2) editing interface

(3) final video

Figure 1: A group of friends go to lunch and place a 360 camera at the center of the table between them (1). They use the camera to record the entire interaction and feed the resulting footage into ConvCut (2). In the editing phase, they use ConvCut to quickly navigate and select a subset of lines from the conversation. As they select lines, the system automatically generates an edit (3) by choosing a framing that emphasizes the speaker of each line.

*e-mail: anhlt92@cs.stanford.edu
†e-mail: maneesh@cs.stanford.edu

It also labels social indicators of memorable moments including instances of laughter as well as extreme changes in speaking volume, facial expression or gestural motion. Most of this information is not available in previous video editing tools [14, 15] and we show that it lets users quickly navigate raw video using keyword search over the transcript and labels. Users can mark these moments and our tool edits them together, automatically choosing a regular field of view (RFOV) framing that emphasizes the speaker of each line of speech. If desired, users can further modify the automatic edit to include alternative framings or reactions from the other people in the group.

The resulting highlight video condenses time by eliminating unmemorable moment and it condenses space by using a RFOV framing to focus viewers' attention on the most important region of the 360 frame. By default, the highlight is edited like a dialogue-driven film scene, cutting between the speakers and ConvCut allows users to add in reactions shots as desired. Thus, it is designed for viewing outside of headset and sharing via social media. We demonstrate the effectiveness of ConvCut by having users produce 14 video highlights (0.27-2 min) extracted from a variety of social conversations (25-60 min) involving 2 or 3 participants. The clips include jokes, intellectual discussions, funny stories and endearing interactions with children.

2 Related Work

Our tool is related to four areas of prior work in video editing.

Automatic video editing of live events. Researchers have designed automatic tools to edit together footage of unstaged, live events in a number of domains. Ranjan et al. [22] describe a multi-camera control system for framing and cutting shots of group meetings using audio and motion capture data. Heck et al. [9] propose an automatic system for editing lecture videos captured by unattended and stationary video cameras. Lu and Grauman [17] condense a long egocentric video into summary of key events. Arev et al. [1] automatically edit together multiple egocentric videos of the same social event. As with these tools, we aim to reduce the overhead of capturing and editing unstaged content by leveraging domain specific audio and visual cues. However, we focus specifically on social conversations and use 360 footage as input. Moreover, while these tools are designed to produce a single edit over all the input footage as comprehensive summary of the event, ConvCut lets users control extraction of individual highlight moments.

Converting 360 video to RFOV video. Researchers have also investigated techniques for converting 360 videos into RFOV videos for viewing outside of the headset. Su and Grauman [28] automatically learn a "capture-worthiness" metric to guide a virtual RFOV camera through a 360 video scene. Lai et al. [13] propose a saliency driven approach to summarize moving 360 footage into a RFOV hyperlapse. Hu et al. [10] use deep learning to pilot a virtual RFOV camera through 360 sports videos. These methods focus on condensing a 360 video in the spatial domain but keep the video temporally intact. In contrast, our work aims to extract short RFOV highlights from a 360 video. Closer to our approach is the work of Truong et al. [30], which distills guidelines for extracting RFOV shots from 360 social event footage. While we leverage Truong et al.'s methods, we go further to provide users with an interface to easily search for meaningful shots in the footage and edit such shots together. Moreover, we specifically focus on dialogue based social conversations and leverage the audio content of the video to provide additional structure for editing.

Transcript based video searching and editing. Several researchers have developed high level tools for searching and editing video and audio content using time aligned transcripts. For example, Berthouzoz et al. [2] present tools for editing talking- head style interview video. Rubin et al. [23] and Shin et al. [27] enable editing and rearranging of speech recordings for audio podcasts and voiceovers. Pavel et al. [21] generate structured summaries of lecture videos to facilitate browsing and skimming. Cour et al. [6] and Pavel et al. [19] align film scripts with corresponding films to enable text based search on visual content. Pavel et al. [20] present a system for reviewing and annotating video with feedback. Truong et al. [29] focus on tools for annotating and aligning b-roll footage to voiceover narrations. Leake et al. [14] develop a system to cut together multiple takes of pre-scripted dialogue driven scenes using film-editing idioms. ConvCut similarly uses time-aligned transcripts to facilitate searching and editing of speech driven scenes. However, unlike the previous tools, ConvCut works with 360 recordings of unscripted, social conversations and supports editing these recordings into shareable RFOV highlight videos. ConvCut leverages labels specifically designed to help users find memorable moments in the conversation. Almost none of these labels were present in the previous tools.

Meeting capture and browsing. A number of research groups have developed systems for capturing face-to-face meetings and browsing them using a transcript-based interface [5, 7, 8, 11, 31, 33]. Most of these systems require building specialized meeting rooms where multiple cameras are placed in a fixed configuration around a table to capture all of the meeting participants (e.g. one camera/mic per seat). This setup ensures that the system can identify the participants and the speaker at each moment. But unlike our approach, these systems cannot be used outside the room they are installed in, and are often very expensive. One notable exception is Lee et al.'s [15] work on a portable meeting recorder, which like our approach uses a portable omnidirectional camera to capture a meeting. However, none of these systems provide tools for finding social highlights (e.g. laughter, expression change) or for producing a shareable highlight that cuts back and forth between the participants.

3 Interface Workflow

Consider the following usage scenario. Alice and a group of her friends go to lunch and with the consent of everyone at the table she places a 360 camera on the table to record the interaction (Figure 1a). She positions the camera so that it sits at about eye level and captures all of the friends[1]. After the conversation, Alex feeds the 360 recording into ConvCut for editing into a set of highlight videos that she can privately share amongst the group or post to social media.

The ConvCut interface (Figure 2) includes four main components: the *Transcript View* lets users quickly find and select a subset of lines from the conversation; as the user selects a line, the system automatically generates an edit by choosing a framing that emphasizes the speaker of each line and displays the selected framings in the *Edit View*; Users can preview the edit in the *Playback View*; if users wants to change the framing, they can select from alternatives provided in the *Clips View*.

3.1 Transcript and Edit Views

The Transcript View on the left side of the interface displays a transcript of the recorded conversation. It is time-aligned with the recorded audio track and is segmented into sentences. Each such line also shows the name of the speaker and the time it was said in the recording. Selecting a line corresponds to selecting the portion of the recording where that line occurred. Clicking on the "Play" button next to the speaker's name, plays a video clip of the line in the Playback View so that users can see and hear the line in context. The transcript contains both spoken words from the conversation and non-speech sounds annotated in parenthesis such as "(laughter)". Presenting the conversation audio as a text transcript in this manner lets users find relevant segments of the conversation by reading the

[1]To capture our datasets, we used a Garmin VIRB 360, which is about the size of a small Rubik's Cube. We expect that in the future, as these camera become more popular, they will become even smaller and less conspicuous.

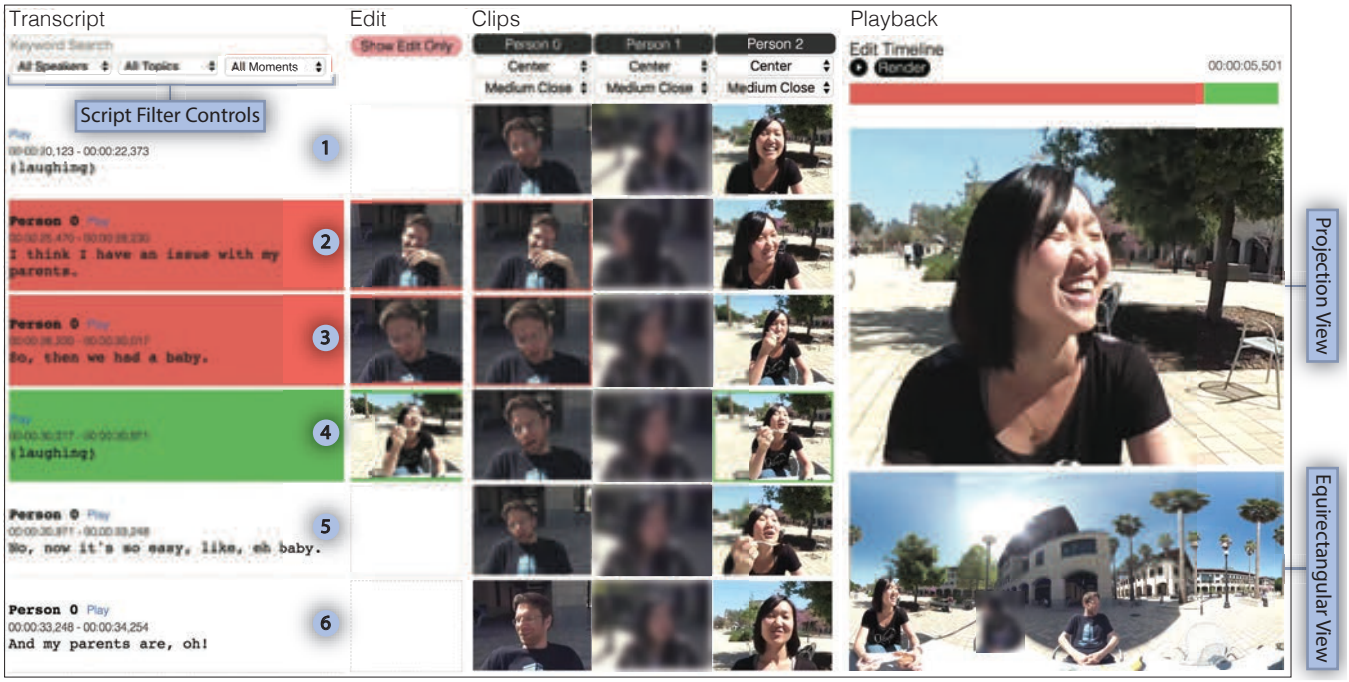


Figure 2: Creating an edit using ConvCut . The Transcript View (left) lets the user quickly find and select a subset of the conversation. She has identified three lines, (2), (3) and (4), from the transcript to add to her edit. ConvCut automatically chooses a RFOV video clip with a medium-close, centered framing of the speaker for lines (2) and (3), and adds it to the corresponding point in the Edit View. Line (4) is not associated with any one speaker because it is a '(laughter)' annotation. So the user selects a clip of Person 2 from the Clips View. ConvCut assigns each line in the edit a color based on that line's speaker. It changes the background of the line to this color and adds a bar of this color to the Edit Timeline to visualize how long each person speaks and where their lines lie with respect to the entire edit sequence. The user clicks on the play button in the Edit Timeline to preview the edit in the Playback View.

text. This is significantly faster than scrubbing through the timeline of a conversation video that may be over an hour long.

However, an hour of transcript text can still be lot for users to skim through. We provide Script Filter Controls located above the transcript to facilitate even faster searching for memorable moments. The search box lets users query by keywords they may remember from the conversation and our interface filters the Transcript View to show only the lines containing those keywords. The "Speaker" drop down lets users see only the lines spoken by the selected person. The "Moments" drop down lets users only see lines containing "Laughter", extreme "Volume Change", "Intense Expressions" or strong "Gestural Motion" as these are often indicators of memorable moments. Finally, the "Topics" drop down menu displays the higher level topics the group talked about during the conversation. Selecting one of these topics filters the Transcript View to show only the lines relevant to that topic. In practice we have found that the list of topics also serves as a reminder to the user about the variety of subjects discussed in the conversation.

Clicking a transcript line adds it to the edit sequence. ConvCut automatically adds a RFOV video clip with a medium-close, centered framing of the speaker to the corresponding point in the Edit View, (Figure 2, lines 2 and 3). Clicking the play button under the Edit Timeline lets users preview the sequence of clips that have been added to the Edit. Thus, ConvCut provides a fast path for creating a highlight video, in which users simply select a set of lines from the Transcript View and ConvCut does all of the low-level work necessary to extract the appropriate RFOV video clips from the 360 video and sequence them together into a result that jumps back and forth between the speakers of each line.

3.2 Clips View

The Clips View lets users change the framing of the video clip associated with any line in the edit sequence. For each line, it provides a variety of RFOV framings for each person captured in the 360 video. The clips are organized so that each person appears in a single column and the user can set the 'Position' (i.e. left, center, right) and 'Size' (i.e. medium-close, medium, long) of their face in the frame using drop-down menus. ConvCut similarly provides establishing shots showing two or more people in the conversation whenever they are sitting close enough together. Initially the people are named Person 1, Person 2, etc, but users can click on a name label at the top of a column to modify the name. This change propagates through the rest of the interface. User can preview the clip in the Playback View by clicking the play button in the bottom right corner of the clip thumbnail that appears on hover. Clicking the plus button in the bottom left corner of the thumbnail adds the clip to the edit sequence. Thus users can adjust the automatic edit to change the framing of the speaker or to show the reactions of others for each selected line in the transcript.

If users want to modify the framing of a clip or frame a non-human object, they can click on the clip to activate *framing specification mode* which lets them adjust the shot orientation by dragging on the frame in the Projection View (Figure 3). They can also click on any location in the Equirectangular View to navigate directly to that viewpoint (Figure 3). To adjust the zoom level users can press the W button to zoom out and create a wider shot or the T button to zoom in and create a tighter shot. Once users are satisfied with their framing, they can click the Save button to add it as another framing option for all lines.

3.3 Playback View

The Playback View consists of two video players. The Projection Player shows only the RFOV projected clip, while the Equirectangular Player shows the complete 360 scene in equirectangular projection to give users spatial context for the location of the RFOV projection. Once users finalize an edit, they click on the Render

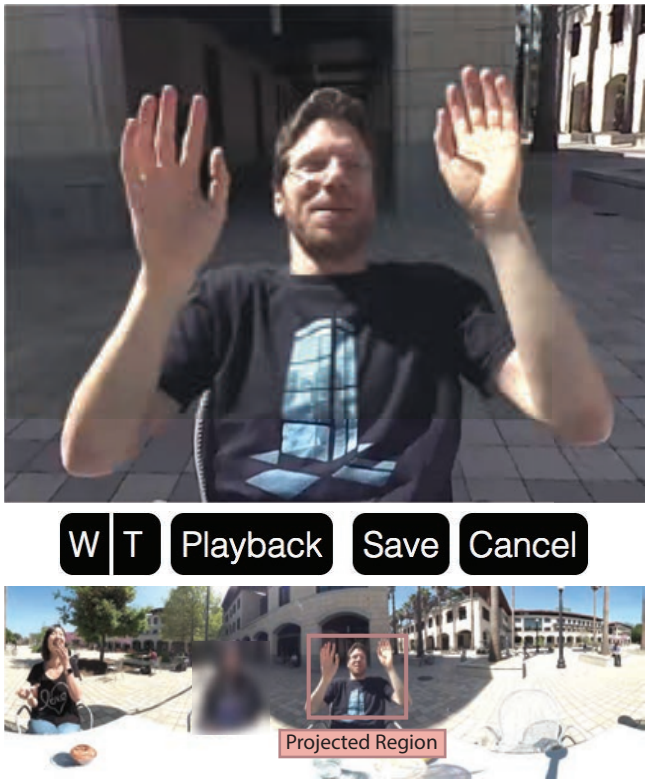


Figure 3: Our shot specification interface. The user modifies a clip of Person 0 by dragging on a frame in the Projection View (top). As she changes the framing in the Projection View, a box appears over the projected region in the Equirectangular View (bottom). The W and T buttons let her zoom out and zoom in respectively. The Playback button lets her preview the modified clip in the Playback View.

button under the Edit Timeline to generate a high-quality rendering of the highlight video. In practice users often generate multiple edits for a conversation to capture different highlights.

4 Algorithmic Methods

ConvCut relies on three main processing steps to support navigation and editing of 360 footage. (1) In the transcription step it obtains a transcript of the video, time-aligns it to the speech and segments the video into one segment per sentence in the transcript. (2) In the RFOV generation step it detects and tracks the faces visible in each segment and generates regular field of view video clips with a variety of cinematic framings for each person in the conversation. (3) In the labeling step it analyzes the video clips and transcript text to extract labels (e.g. speaker of each line, topics covered, laughter) that let users efficiently search through the captured content.

4.1 Step 1: Transcription

Given a 360 video recording of a conversation, we use `rev.com` to obtain a text transcript for recording audio. The verbatim transcript costs about $1 per minute, takes about 24 hours of turnaround time, and contains speaker change indicators and approximate timestamps for each spoken sentence, as well as annotations of non-speech sounds like laughter.

We then use the phoneme-mapping method of Rubin et al. [23] to time align the transcript to the audio track of the video recording. While this approach gives us finer, word-level alignment accuracy than the sentence-level timestamps returned by `rev.com`, we found that it often struggles when people talk over each other or when there is loud background noise. In such cases, Rubin et al.'s algorithm only provides a coarse alignment in which a long multi-line sequence of text corresponds with a long segment of audio, but it does not provide word-level alignment. In these cases we consider the sentence-level timestamps provided by `rev.com`. Finally, we combine the word- and sentence-level timings to split the raw 360 video into one video segment per sentence.

4.2 Step 2: RFOV Generation

To generate the RFOV video clips of each person in the 360 video, we extend the method of Truong et al. [30] to more robustly track faces. Since modern face detectors are designed for perspective views, we follow Truong et al.'s approach and first split the input video into 8 overlapping perspective projections with horizontal and vertical fields of view of 60°, spaced 45° apart. We then run the face detection method of Li et al. [16] to obtain a face bounding box and use Saragih et al. [25] to obtain facial landmark points for all faces in each frame of each projection. Unlike Truong et al., we next eliminate small faces of people in the background, to retain only the faces of people most likely to be part of the conversation. Specifically we normalize the face bounding box heights by the frame height and then filter out bounding boxes with normalized height less than a threshold τ, that we empirically set to 0.08. We have found that this filtering step also improves the accuracy of our face tracking in environments containing people in the background.

Truong et al. [30] track faces across frames by grouping face bounding boxes that overlap on adjacent frames – by any non-zero overlap amount – as representing the same face. This approach cannot robustly handle three common issues; (1) When people are sitting close to one another their bounding boxes often overlap by a small amount and they are grouped together. (2) When the face detector misses detecting a face for a few frames (e.g. if the person turns their head too far or image noise throws off the detector), the face is treated as belonging to two different people before and after the missing detections. (3) Similarly when a person leaves and then re-enter the scene they are treated as two different people.

To robustly handle the first issue we only group together face bounding boxes that overlap only by 50% or more. In practice we have found that even when people sit very close together there is less than 50% overlap between their face bounding boxes. To handle the second issue of missed detections, instead of considering overlap between faces in adjacent frames, for each face we consider overlap between the nearest face within a 2 second window of frames. Thus, our tracking approach can bridge up to two consecutive seconds of missed face detections as long as the person hasn't moved very far and the bounding box overlap is high when the face is detected again. In practice we have found that our face detector misses faces for a few frames here and there and rarely misses them for 2 consecutive seconds. After this grouping step we interpolate the face bounding boxes for any frame in which the face was not detected.

To handle the third issue, where people leave (e.g. to use the restroom) and return to the table, we detect whenever the number of faces in the scene changes. When a person "exits" the scene, our face count decreases by one and we add the exiting face to a list of exiting faces. When a person "enters" the scene we increase the face count by one and we compare the entering face to each face in our exiting faces list using an appearance-based distance measure. Specifically we use the χ^2-distance over the normalized color histograms of the faces as developed in previous work on face matching [2, 12]. If the χ^2-distance is less a threshold α, (empirically set to 5) we group the faces as the same person and remove the face from the exiting faces list. If no such match is found we treat the entering face as a new face. Note that this facial appearance matching approach also lets us handle the case when our face detector fails for more than 2 consecutive seconds.

To validate our approach, we manually produced ground truth face tracks for 5 social conversations (Cafe, Research, Baby, Tea-

house and Courtyard) shown in Table 1. Our face tracking method achieved 98.9% accuracy for these conversations. This is a significant improvement over the method of Truong et al. [30] where the average accuracy was 79.5%.

Finally, we use the methods of Truong et al. [30] to generate RFOV framings for each resulting face track. Specifically we generate nine perspective view framings of each face; we place the face at three different horizontal locations in the frame (left, center, right) and three distances from the camera (medium-close, medium, wide-angle). We also generate establishing shots containing the bounding boxes of two or more faces if the angular distance between the outermost edges of the bounding boxes is 60° or less.

4.3 Step 3: Labeling

We analyze the video, audio and text transcript of the clips to determine the face speaking each line in the transcript and to extract the topics covered in the conversation. We also identify social indicators of memorable moments by finding laughter as well as extreme changes in speaking volume, facial expression and gestural motions. These per-line labels are designed to further help users search the 360 video for memorable content.

4.3.1 Speaking Face Identification

To identify the speaker of each line we leverage the audio-visual sound source analysis algorithm of Owens et al. [18]. Given a video as input this algorithm computes a heatmap indicating the pixels most likely to be the source of the sound. Applying this algorithm to each clip of our 360 video generally highlights the mouth of the speaker. Suppose we have tracked m faces $\{f_1,...,f_m\}$ in a video clip. For each tracked face f_i, in the clip, we compute a heatmap score $\mathbf{H}(f_i)$ as the sum of the heatmap magnitudes that fall in the mouth region (based on the facial landmark points) and average this sum across all the frames in the clip. We can then treat the face f generating the maximum heatmap score as the speaker of the transcript line associated with the clip. However, in practice we have found that the heatmap generated by Owens et al. often highlights mouth regions of non-speakers as audio sources, especially when there is some background noise, the non-speaker is changing facial expressions and/or moving their head.

To increase the robustness of our speaker identification algorithm we have designed a dynamic programming algorithm that combines the sound source analysis of Owens et al. with information about speaker changes from the `rev.com` transcript. Suppose $S_n = (s_1,\ldots,s_n)$ is a sequence of faces assigned to the first n lines of the transcript and their associated video clips. That is, each s_j is assigned one of the faces $f \in \{f_1,...,f_m\}$. We define a speaker assignment score $E(S_n)$ for the speaking face sequence S_n as

$$\mathbf{E}(S_n) = \mathbf{E}(s_{n-1}) + \mathbf{H}(s_n)\mathbf{T}(s_{n-1},s_n) \quad (1)$$

where the heatmap score $\mathbf{H}$ measures how well the assignment reflects the Owens et al. sound source analysis and the speaker transition score $\mathbf{T}$ measures how well the assignment captures the speaker changes from line to line in the `rev.com` transcript.

For each sequential pair of lines $i-1$ and i we set a Boolean variable $c_{i-1,i}$ to 1 if the transcript says the speaker changed between the lines. Otherwise we set it to 0. We then define the speaker transition score as

$$\mathbf{T}(s_{n-1},s_n) = \begin{cases} 1, & \text{if } c_{n-1,n} \text{ and } s_{n-1} \neq s_n \\ 1, & \text{if not } c_{n-1,n} \text{ and } s_{n-1} = s_n \\ 0, & \text{otherwise} \end{cases}$$

The recursive definition of Equation 1 allows us to apply dynamic programming to efficiently solve for the optimal speaker assignment, while accounting for information from the video, audio and the transcript.

We compared our automatically generated speaker labels against manually extracted ground truth for 5 scenes (Cafe, Research, Baby, Teahouse and Courtyard) shown in Table 1. We found that our approach correctly labeled the speaker for 100% of the lines in conversations with 2 people and 83% of the lines for conversations with 3 people. The original Owens et al. algorithm without our dynamic programming approach,correctly labeled only 74% of the 2 person conversations and 65% of the 3 person conversations in our dataset. The errors in our speaker labels often resulted from multiple people speaking at the same time as is often the case in 3 person conversations. Even ground truth can be ambiguous in such cases and we leave it as future work to improve speaker detection when there are multiple simultaneous speakers.

4.3.2 Topic Modeling

We use the non-negative matrix factorization (NMF) based topic modeling method of Shahnaz et al. [26] to extract topics from our transcript. The input to the NMF algorithm takes in a list of the unique words from the transcript and their TF-IDF frequencies. The output is a set of word clusters where each cluster represents a different topic.

To apply this algorithm to our transcripts we first lemmatize each word so that the same word with different endings (e.g, run and running) are considered the same. We then remove high frequency stop words (e.g., 'the', 'be', 'and', etc.) as contained in the Natural Language Toolkit (NLTK) [3]. We also filter out all instances of the words "mhmm", "like", "eh", "um", "oh", "mmm", and "okay". While these are not traditional stop words for written English, they are filler words for conversational English and add little semantic value. Finally, because there are individual differences in filler words e.g., a few people often used the word "literally" with high frequency), we also filter out words that occur in more than 70% of sentences for an individual person.

After applying NMF topic clustering , we associate each sentence in the transcript to a resulting topic if at least one words in that cluster is also present in the sentence.

4.3.3 Laughter

Laughter in a conversation often indicates that something funny occurred. The `rev.com` transcript provides annotations of laughter. We mark segments where these annotations occur as "interesting". While these laughter labels from the transcript have worked well in practice, it may also be possible to use automatic laughter detection to identify laughter segments automatically from the audio, using for example the approach of Ryokai et al. [24]

4.3.4 Volume Increase

We observe that when people get excited in conversation, they tend to speak more loudly. To detect these moments, we compute a loudness score for each video clip as the average root mean square energy of the audio signal. We then compute the interquartile range between the 25% quartile and the 75% quartile of the loudness scores across all clips. Finally we consider any loudness score that is greater than the 75% quartile by 1.5 times the interquartile range to be an outlier. We label the clips with such outlier loudness scores compared to the other clips in the video to be "volume increase" clips.

4.3.5 Strong Gestural Motion

People often communicate not just with words, but also with gestures and other body language. When someone moves their body more than normal, it can indicate that they are saying something noteworthy. To detect such movement, we apply the OpenPose [32] [4], a 2D pose detection algorithm, to each frame of the video. For each person in a clip, we assign a gesture score as the average euclidean distance between joint locations for each sequential pair of their poses. We then apply the quartile-based outlier detection we used

Edits Produced with ConvCut

Conversation (# participants) Highlight	Video Length (min)	Edit Time (min)
Cafe (3 people):	45:00	
Advice	00:27	02:00
Human child interaction	00:40	03:00
Human cat interaction	00:16	04:00
Research (3 people):	25:00	
River	00:44	07:00
Advisor	00:19	02:00
Baby (2 people):	60:00	
Mom	00:21	02:00
Jokes	00:34	02:00
Dinosaur	01:59	10:00
Art (2 people):	30:00	
Turrell	01:52	10:00
Daughter	01:26	08:00
Teahouse (2 people):	30:00	
White privilege	00:17	02:00
Friend	00:57	05:00
Courtyard (2 people):	25:00	
Biking	00:42	03:00
Office (2 people):	25:00	
Wedding	01:28	05:00

Table 1: ConvCut has been used to produce highlights for 7 conversations ranging in length from 25 to 60 min (top of each row). The resulting highlights range in length from 16 sec to 2 min (Video Length) and required 2 to 10 min to create (Edit Time).

for volume increase labeling to the gesture scores and label any resulting outlier clip as containing"strong gestural motion".

4.3.6 Intense Change in Facial Expression

An intense change in expression from a listener (e.g., raising eyebrows quickly or opening their mouths widely) in reaction to something that has just been said is a good indicator of a noteworthy moment. We detect such changes in expression for each person by computing an expression change score as the average Euclidean distance between the positions of facial landmarks in sequential pairs of frames, for all clips in which a participant in not speaking. We then apply the interquartile outlier detection we used for volume increase labeling to the expression scores and label any resulting outlier clip as containing "intense expression".

5 RESULTS

ConvCut has been used to edit 14 shareable highlight videos from seven 360 video recordings of social conversations captured in a variety of different environments (Table 1). Figure 4, shows examples of edited highlights for 5 of our 7 conversations. We obtained the raw video by asking volunteers, including both authors, to use a Garmin 360 camera to record conversations taken with friends or family. We instructed these volunteers on proper camera setup before their conversations. We also encouraged them to talk naturally, and did not suggest conversation topics. Co-authors participated in 2 of the 7 conversations (Baby and Teahouse). The volunteers reported that although some people were initially a bit unnatural around the camera, all of them stopped noticing it as they became absorbed in the conversation. The resulting raw 360 videos were each 25-60 minutes long, 5K in resolution and 20-40 GB in size. Each 10 minutes of video required approximately 2 hours to process through our algorithmic pipeline on a cloud compute cluster with 80 cores. Face detection and tracking, as well as pose detection, accounted for the majority of this processing time.

We asked nine of the volunteers, none of whom were authors, to use ConvCut to extract a highlight from a conversation in which they participated. In some cases we asked them to extract another highlight from a conversation they did not participate in. All were first-time users. Their edits ranged from 16-119 seconds in length and required between 2-10 minutes to create. Most of this time was spent choosing which moments to extract and previewing the edit.

We observed that the first-time users often started by browsing the "topic" filters to refresh their memory of the conversation. After this refresher, some users identified specific parts of the conversation that they wanted to extract and used the "topic" filters and "keyword" search to navigate to those moments. Other users who were still undecided used one of the "Laughter", "Volume Change", "Intense Expressions" or "Gestural Motion" filters to find a noteworthy segments. Similarly when they were not involved in the conversation they relied on these labels as well as directly reading the transcript and performing keyword search to browse the conversation and identify memorable moments.

As these users added lines to the edit they often accepted the automatically selected framing of the speaker. However, there were instances where they selected an alternative clip from the Clips View to introduce establishing shots or reaction shots and to resolve jump cuts between non-contiguous lines. The process of selecting an alternative usually took a few seconds. The manual framing feature was used less often.

The resulting highlights consisted of a variety of different types of moments. Many of them were jovial, such as jokes, dramatic accounts of recent events, funny stories, and endearing interactions with children. Though many of the conversations were lighthearted, ConvCut can also be useful in more informative settings. For example, one user created a highlight of his opinions about the art at a museum. Such highlights are potentially useful as memory refreshers for repeat visits or recommendations to others.

Using ConvCut to create video highlights. We asked our first-time users to rate how difficult/easy it was to create video highlights using ConvCut . The average score was 6 on a 7 point Likert scale (1 = very difficult to use, 7 = very easy to use). Seven of the nine users gave scores of 6 or above while the remaining two users gave middle scores of 4 and 5. Many of these users commented on how the segmentation and organization of the 360 video into clips based on transcript lines made it easy for them to browse the conversation and quickly generate an edit that jumped back and forth between different speakers and framings. All of the users rated the tool as useful in helping them to identify memorable moments (average 5.88 on a 7 point Likert scale). They particularly appreciated the topic filters and keyword search. Seven of the nine users rated the topics as being informative in helping them to remember things that came up in the conversation (average 5.86 on a 7 point Likert scale). The other two users didn't rate the topics because they did not use them. Instead they remembered the conversation and used keyword search to find the highlights they wanted to extract.

Everyone appreciated the videos as a means of remembering and sharing moments. One user commented that *"you feel like you're almost there."* Another user expressed, *"I've rewatched them many times and still am not tired of them at all, and in fact can't help but laugh every time."* Another user, who participated in a serious conversation about art, appreciated having the video to help him remember how the art scene in a particular city stood out to him. He commented that the video would be useful to share with other friends who were interested in visiting that art scene to help inform their decisions. Parents who used the tool liked being able to keep track of memories with or of their children. Eight out of nine users said they would use the tool again in various settings such as get-togethers with a close group of friends, events including birthday parties or weddings, car conversations during long drives and roundtable discussions such as reading groups or other meetings.

Sharing and viewing video highlights. All of our users said they would be comfortable sharing the video highlights they created with

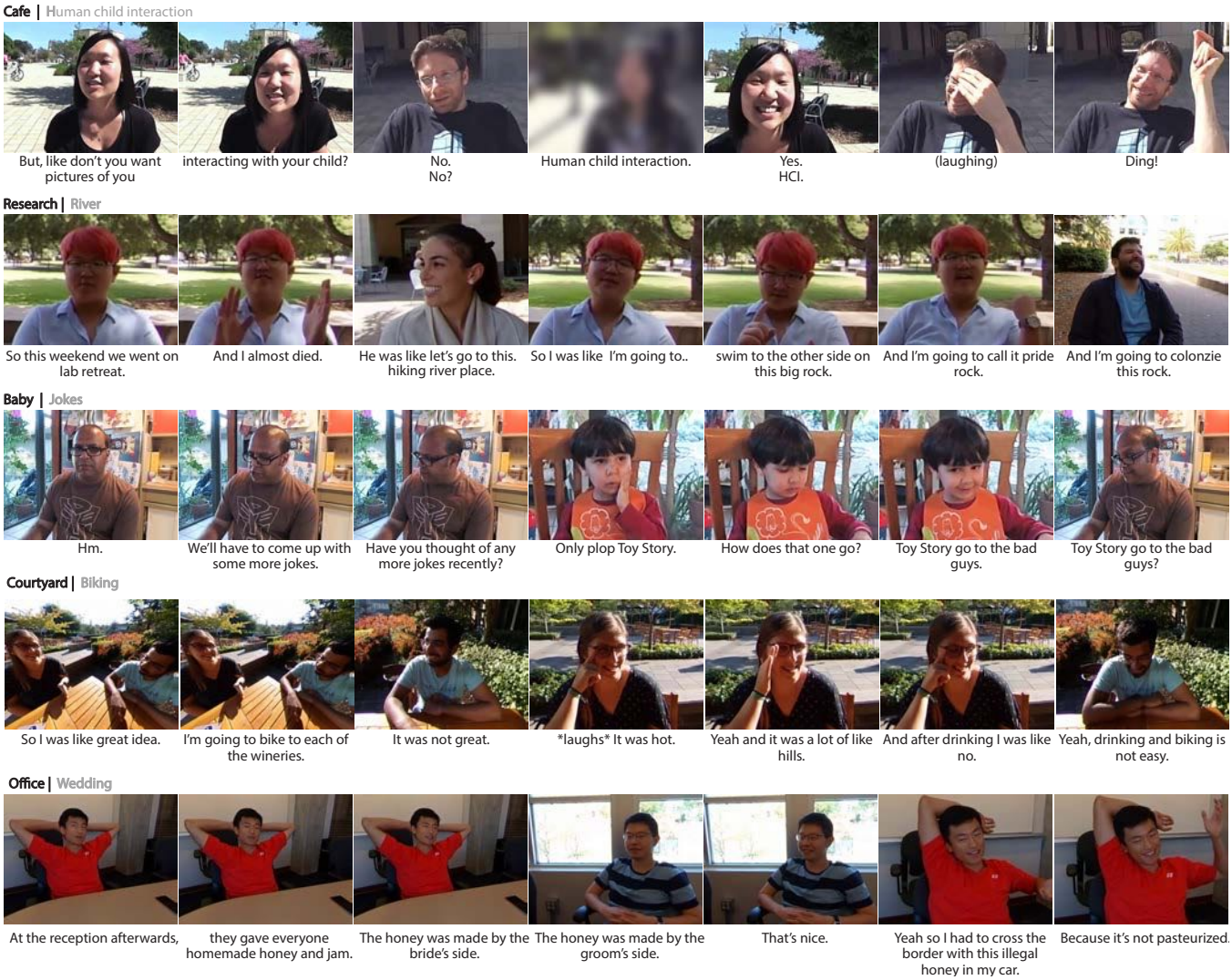


Figure 4: Example edits of memorable highlights for five of our conversations. The Human child interaction moment from the Cafe scene (first row) captures a witty back and forth between friends. The RFOV framings were all selected automatically. The River moment from the Research conversation (second row) is a funny story about a lab retreat and was identified using the topic clusters and the "Volume Change" label. The Jokes moment from the Baby scene (third row) captures a cute moment of a child. The user (not an author) added a reaction shot for the fifth line. The rest of the edit was automatically generated. The Biking moment from the Courtyard scene (fourth row) captures two friends bonding over biking experiences and was identified by searching for the keyword "Bay Bridge". The Wedding moment in the Office conversation (fifth row) captures a friend sharing a story about an uncommon part of a wedding he recently attended. Please see the video and supplemental materials for complete video results.

others, particularly those who participated in the conversation. Five of them actually did shared their highlight with other people; two of them users shared their edits with others from the same conversation, another two shared with close friends and family, and the last user shared with a wider social circle.

We also informally interviewed viewers who were not involved in the highlight creation. They typically found the resulting videos to be funny or endearing. Common responses to the highlights included *"cool"*, *"lollll"*, and one viewer even commented *"I'm jealous I want to be in a video like that"*. Many viewers were impressed with the back and forth editing between the speakers which made the conversation "easy to follow". They often asked how the edit was created and wanted to learn more about our system. Viewers who had taken part in the original conversation frequently asked if they could make their own highlights of other parts of the conversation. Viewers who hadn't been part of the original conversation asked how the video was captured as they were not used to seeing everyday conversations filmed up close.

To further evaluate the editing quality of highlights created with ConvCut, we generated fixed, wide-FOV versions of the three of the highlights. Such a fixed wide-angle shot is often the easiest way to capture a social conversation today as it ensures that all participants are always visible. We then asked seven people to compare these wide-FOV highlights to ConvCut generated highlights, which use shot, reverse-shot dialogue editing. Viewers strongly preferred the shot, reverse-shot editing of ConvCut in all cases. They liked being able to see the speakers and listeners' facial expressions more closely in the ConvCut highlights and said that the wide-FOV edits felt "amateur or homemade". One viewer commented that the closer shots *"helped [him] to focus on the conversation"*. In contrast, one of the fixed wide-FOV videos contained distracting objects in the background that caught his attention. Many viewers also pointed out that jump cuts were much less obvious in the shot-reverse shot edits compared to the fixed FOV edits. One viewer said that she *"didn't even realize that there were jumps in the conversation"* until she saw the fixed wide-FOV edit.

6 Limitations and Future Work

While ConvCut facilitates the creations of shareable highlights from 360 video of social conversations, it does have a few limitations that suggest directions for future work.

Improving speech alignment. At exciting moments in social conversation, people often talk or laugh over each other, making it difficult for our algorithms to align a text transcript to the audio. In such instances, ConvCut currently falls back to using the sentence-level timings generated by `rev.com`'s human transcribers. However, this is not ideal because the human generated timings can be inaccurate. Resolving overlaps in audio from multiple sources is an open area of research.

Automatically transcribing conversations. While our current implementation relies on manual transcription, we have seen rapid improvements in the quality and decreases in cost for automatic transcription. Comparing automatic transcripts from Google Cloud Speech ($0.024/min) to manual transcripts from `rev.com` ($1/min) we find that the automatic transcripts have an average accuracy of 72.4%. We anticipate that automatic transcription will soon yield transcripts accurate enough to replace the manual transcription step of our pipeline. As such, we designed ConvCut to be agnostic to how the transcript is generated so that it can take either manually or automatically transcribed transcripts as input.

Social norms, privacy and consent. An issue underlying our work is that participants in a social conversation cannot always anticipate where the conversation might go and could be concerned about privacy if the conversation covers sensitive or intimate topics. Nevertheless, in our approach, because the 360 camera remains large enough to be visible, the conversation cannot be recorded without the knowledge and consent of the participants. The social norms around ownership and consent to sharing of the resulting raw footage as well as edited highlights need to be established. The concerns are similar to those around taking photos with friends at a party and deciding together which ones are shared to which social channels. While the norms around acceptability of posting such photos do vary from group to group, these have developed and evolved over time. We expect that as systems like ours become more common, people will similarly start establishing implicit guidelines around privacy and sharing the raw footage and highlights within the group of participants and with larger groups of friends, acquaintances and the public at-large. One direction for future work is to consider technologies that allow all of the participants in the conversation to vote on how the footage is edited and shared. For some editing and sharing actions only a single vote in favor may be necessary, while others might require a simple majority while others still might require a unanimous vote.

7 Conclusion

Social conversations are an opportunity for people to connect with friends and family every day. They're often filled with memorable moments such as jokes, riveting stories, or intellectual debates. Video highlights are a natural way to relive and share these moments with others. However, such moments are difficult to anticipate and capture using traditional RFOV cameras. Our proposed workflow offloads the task of recording to a static 360 camera. Our system then takes in the resulting footage, segments it, and analyzes each segment to generate informational labels. The user uses these labels to easily navigate the footage in our ConvCut interface and generate RFOV highlights within a couple of minutes. This approach lets users focus on the conversation rather than the act of recording, with the comfort of being able to tractably extract memorable moments afterwards. As 360 cameras get increasingly smaller and cheaper, we believe that these kinds of systems are essential in enabling users to explore new methods of video capture while helping them to manage the large amounts of data that come with it.

References

[1] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)*, 33(4):81, 2014.

[2] F. Berthouzoz, W. Li, and M. Agrawala. Tools for placing cuts and transitions in interview video. *ACM Trans. Graph.*, 31(4):67–1, 2012.

[3] S. Bird. NLTK: The natural language toolkit. In *Proc. of COLING/ACL*, pp. 69–72, 2006.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[5] P. Chiu, A. Kapuskar, L. Wilcox, and S. Reitmeier. Meeting capture in a media enriched conference room. In *International Workshop on Cooperative Buildings*, pp. 79–88. Springer, 1999.

[6] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *European Conference on Computer Vision*, pp. 158–171, 2008.

[7] R. Cutler, Y. Rui, A. Gupta, J. J. Cadiz, I. Tashev, L.-w. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *Proc. ACM Multimedia*, pp. 503–512, 2002.

[8] R. Gross, M. Bett, H. Yu, X. Zhu, Y. Pan, J. Yang, and A. Waibel. Towards a multimodal meeting record. In *IEEE International Conference on Multimedia (III)*, pp. 1593–1596, 2000.

[9] R. Heck, M. Wallick, and M. Gleicher. Virtual videography. *ACM Trans. on Multimedia Computing (TOMM)*, 3(1):4, 2007.

[10] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. *CVPR*, 2017.

[11] A. Jaimes, K. Omura, T. Nagamine, and K. Hirata. Memory cues for meeting video retrieval. In *Proc. of CARPE*, pp. 74–85, 2004.

[12] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz. Exploring photobios. In *ACM Transactions on Graphics (TOG)*, vol. 30, p. 61, 2011.

[13] W.-S. Lai, Y. Huang, N. Joshi, C. Buehler, M.-H. Yang, and S. B. Kang. Semantic-driven generation of hyperlapse from 360° video. *CVPR*, 2017.

[14] M. Leake, A. Davis, A. Truong, and M. Agrawala. Computational video editing for dialogue-driven scenes. *ACM Transactions on Graphics (TOG)*, 36(130), 2017.

[15] D.-S. Lee, B. Erol, J. Graham, J. J. Hull, and N. Murata. Portable meeting recorder. In *Proc.of ACM Multimedia*, pp. 493–502, 2002.

[16] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, pp. 5325–5334, 2015.

[17] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Proc. of CVPR*, pp. 2714–2721, 2013.

[18] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. *arXiv preprint:1804.03641*, 2018.

[19] A. Pavel, D. B. Goldman, B. Hartmann, and M. Agrawala. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proc.of UIST*, pp. 181–190, 2015.

[20] A. Pavel, D. B. Goldman, B. Hartmann, and M. Agrawala. Vidcrit: Video-based asynchronous video review. In *Proc. of UIST*, pp. 517–528, 2016.

[21] A. Pavel, C. Reed, B. Hartmann, and M. Agrawala. Video digests: a browsable, skimmable format for informational lecture videos. In *UIST*, pp. 573–582, 2014.

[22] A. Ranjan, J. Birnholtz, and R. Balakrishnan. Improving meeting capture by applying television production principles with audio and motion detection. In *Proc. of SIGCHI*, pp. 227–236, 2008.

[23] S. Rubin, F. Berthouzoz, G. J. Mysore, W. Li, and M. Agrawala. Content-based tools for editing audio stories. In *Proc. of UIST*, pp. 113–122, 2013.

[24] K. Ryokai, E. Durán López, N. Howell, J. Gillick, and D. Bamman. Capturing, representing, and interacting with laughter. In *Proc. of SIGCHI*, pp. 358:1–358:12, 2018.

[25] J. M. Saragih, S. Lucey, and J. F. Cohn. Face alignment through subspace constrained mean-shifts. In *ICCV*, pp. 1034–1041, 2009.

[26] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information*

Processing & Management, 42(2):373–386, 2006.

[27] H. V. Shin, W. Li, and F. Durand. Dynamic authoring of audio with linked scripts. In *Proc. of UIST*, pp. 509–516, 2016.

[28] Y.-C. Su and K. Grauman. Making 360° video watchable in 2d: Learning videography for click free viewing. *CVPR*, 2017.

[29] A. Truong, F. Berthouzoz, W. Li, and M. Agrawala. Quickcut: An interactive tool for editing narrated video. In *UIST*, pp. 497–507, 2016.

[30] A. Truong, S. Chen, E. Yumer, D. Salesin, and W. Li. Extracting regular fov shots from 360 event footage. In *Proc. of SIGCHI*, 2018.

[31] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. Meeting browser: Tracking and summarizing meetings. In *Proc. of DARPA broadcast news workshop*, pp. 281–286, 1998.

[32] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[33] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with ferret. In *International Workshop on Machine Learning for Multimodal Interaction*, pp. 12–21. Springer, 2004.