# DemoCut: Generating Concise Instructional Videos for Physical Demonstrations

**Pei-Yu (Peggy) Chi[†][‡], Joyce Liu[†], Jason Linder[‡], Mira Dontcheva[‡], Wilmot Li[‡], Björn Hartmann[†]**
†: Computer Science Division, UC Berkeley and ‡: Adobe Research
peggychi@cs.berkeley.edu, joyce.liu@berkeley.edu, linder@adobe.com
mirad@adobe.com, wilmotli@adobe.com, bjoern@cs.berkeley.edu

## ABSTRACT

Amateur instructional videos often show a single uninterrupted take of a recorded demonstration without any edits. While easy to produce, such videos are often too long as they include unnecessary or repetitive actions as well as mistakes. We introduce DemoCut, a semi-automatic video editing system that improves the quality of amateur instructional videos for physical tasks. DemoCut asks users to mark key moments in a recorded demonstration using a set of marker types derived from our formative study. Based on these markers, the system uses audio and video analysis to automatically organize the video into meaningful segments and apply appropriate video editing effects. To understand the effectiveness of DemoCut, we report a technical evaluation of seven video tutorials created with DemoCut. In a separate user evaluation, all eight participants successfully created a complete tutorial with a variety of video editing effects using our system.

### Author Keywords

video; instructions; tutorials; demonstrations; how-to

### ACM Classification Keywords

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

## INTRODUCTION

Do it yourself (DIY) instructional videos show viewers how to carry out physical tasks, such as craft projects, home improvement, repair, or cooking [30]. The availability of free video-sharing sites like YouTube and Vimeo has led to an explosion in user-generated video tutorials online [21]. Effective instructional videos use a range of video editing techniques, including subtitles, annotations, and temporal speed up effects, to concisely communicate physical procedures. However, producing high-quality videos requires significant time investment and expertise. In addition to recording possibly many takes, authors must review and cut the footage and then apply the appropriate editing effects [24]. Instead of investing this effort, many amateurs instead create videos that simply show a long uninterrupted recording of a demonstration. While such videos are easy to produce, they often include a lot of unnecessary footage (e.g., pauses, mistakes, long repetitive actions) that makes it difficult for viewers to focus on the most important steps and actions.

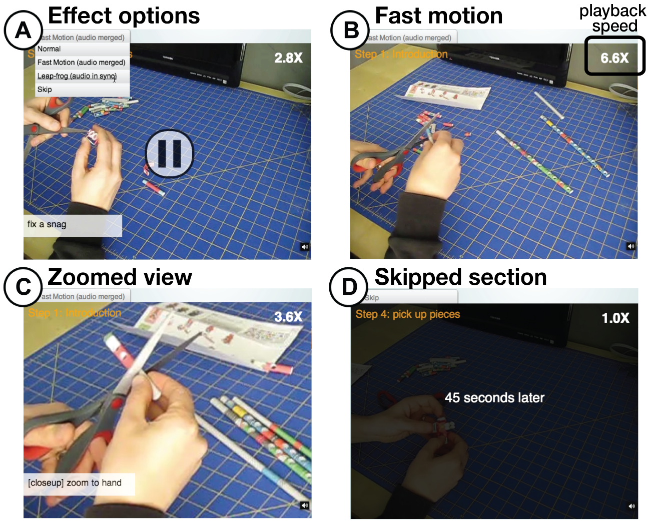

**Figure 1. DemoCut automatically segments a single-shot demonstration recording and applies video editing effects based on user markers (A), including subtitles, fast motion (B), leap frog, zoom (C), and skip (D).**

The goal of our work is to help amateur users produce effective instructional videos. We analyzed existing DIY videos and interviewed video authors to uncover key challenges in creating high-quality video tutorials: organizing long, single take recordings into meaningful steps; removing/condensing unnecessary or repetitive actions; and adding effects that emphasize important details in the demonstration. To address these challenges, we introduce DemoCut, a semi-automatic video editing system that generates concise instructional videos from recorded demonstrations (Figure 1).

With DemoCut, users record a single take of a narrated physical task demonstration and then roughly annotate the recording with markers that indicate high-level steps, important actions, supplies and mistakes. Based on these annotations, the system uses a combination of video and audio analysis to automatically organize the recording into meaningful segments and apply editing effects that make the tutorial more clear and concise. DemoCut supports both temporal effects that increase playback speed or skip segments, as well as visual effects, such as zooming, subtitles, and visual highlights. De-

UIST'13, Oct 08-11 2013, St Andrews, United Kingdom
ACM 978-1-4503-2268-3/13/10.
http://dx.doi.org/10.1145/2501988.2502052

moCut also provides an interface that allows users to quickly review and edit the automatically generated effects.

We used DemoCut to create seven video tutorials in five different DIY domains: electronics, crafts, art, repair and food. The generated videos were concise in terms of video length and descriptive instructions with low effect error rates. We also conducted a small user study where participants used our system to record and edit their own video tutorials. All participants successfully created a complete tutorial that included a variety of video editing effects, and the qualitative feedback on DemoCut was very positive. The participants felt that DemoCut enables a convenient workflow for creating concise video tutorials and that the automatic editing effects are particularly useful for speeding up repetitive actions.

In summary, the main contributions of this paper include:

- A light-weight annotation-based interface for editing instructional videos.
- A set of marker types for annotation derived from our formative work. Markers represent different types of moments that lead to different editing effects.
- A semi-automatic approach for editing DIY video that combines user annotation with audio and video analysis.
- A working implementation of this approach and a preliminary evaluation with both novice and expert video editors.

## RELATED WORK

### Current Practices around How-To Videos

The research community has been investigating the motivations of both authors and viewers of how-to videos and written tutorials. While one primary motivation is to share expertise, published videos also serve as a way to broadcast skill and as an online portfolio [30]. Authors may derive revenue through advertising or referrals [21]. Viewers, on the other hand, typically seek technical explanations, but are also searching for inspiration [29] and looking for validation of existing skills [21]. In aggregate, these studies suggest that how-to videos have a larger variety of purposes and uses than merely communicating technical content. In our work we strive to make authoring of how-to videos more accessible to amateurs while maintaining opportunities for adding individual style through control over editing effects.

### Video Capture, Annotation and Editing

*Capture.* Several research and commercial systems guide users at capture time to yield higher-quality videos. Such systems often employ templates to help users capture sequences of distinct shots (e.g., Snapguide[1]) or suggest framing of the subject or camera view as in NudgeCam [7]. Computer vision algorithms, like face tracking, can be used to offer real-time feedback during such directed actions [10, 18, 7]. Instead of relying on templates, shot suggestions can also be bootstrapped through user dialogs [1]. In contrast to these systems, we work with a single long video take and do not require the author to manipulate the camera during capture. Many leisure activities, such as home repair or cooking, require use of both hands or involve getting one's hands dirty, so camera manipulation is not possible.

*Annotation.* Researchers have investigated how to provide interactions that enable efficient, fluid annotation of video data, from the early EVA system [23] to more recent interfaces like VideoTater that leverage pen input [11]. We do not claim a contribution in the interaction techniques of our annotation interface and take inspiration from such prior work.

*Editing.* Frame-based editing of video is very time-intensive, as it forces users to operate at a very low level of detail. Editors can leverage metadata, such as transcripts [6] and shot boundaries [8], to give users higher-level editing operations at the shot level rather than the frame level. Computer vision techniques can automate certain effects, such as creating cinemagraphs [2, 20], automatically-edited lecture videos [17], zoomable tapestries [3] and synopses [27], or stabilizing shaky amateur videos [22]. When analyzing video is a matter of subjective taste, identifying salient frames can also be outsourced to crowd workers [5]. DemoCut also uses vision techniques for automatic editing. It differs from previous approaches in its focus on a particular application domain – physical demonstration videos. By focusing on a specific domain, DemoCut can make assumptions about the structure of the input and output video, such as the fact that there is a linear set of steps, and offer an interface and algorithms that make it easier to create high quality how-to videos.

### Creating Effective Tutorials

There are many ways to produce effective tutorials. One approach is to track user behavior to automate tutorial authoring [13, 14, 9]. This method also opens the door to interactive tutorials that can respond to user progress [4, 26]. However, tracking user behavior in the physical world, rather than in software, remains a challenge. DuploTrack uses a depth camera to track progress and provide guidance for block assembly tasks [15]. Augmented reality applications overlay real-time information on top of the work area, usually through a head-mounted display. Such systems can provide visual highlights (such as arrows, text, closeup views, and 3D models) for machine maintenance [19], or interactive remote tutoring for repair tasks [16].

In this work we seek to support a wide variety of how-to tasks from craft to home repair and cooking where automatically tracking user activities is not yet possible. To support these tasks we propose a semi-automatic approach where the user marks important moments and the system automatically edits the video based on the user markers. Here we focus on the authoring of how-to videos, and we leave interactive tutorials for physical tasks to future work.

## UNDERSTANDING CURRENT PRACTICE

To gain insight into the editing decisions that go into effective demonstration videos, we analyzed a set of 20 highly-rated videos on YouTube and interviewed six of the authors of these videos about their recording and editing processes.

[1] http://snapguide.com/

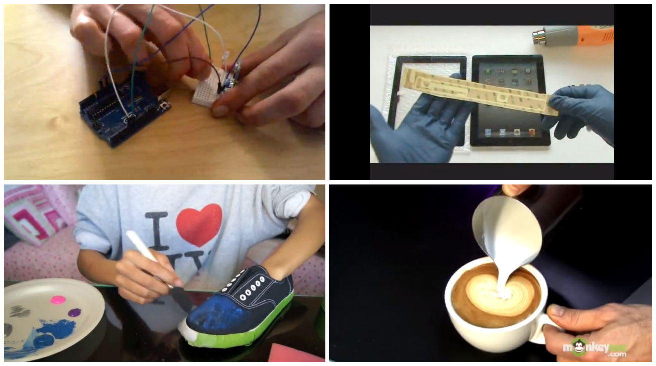

Figure 2. We analyzed 20 DIY instructional videos. Examples included (clockwise from top left): Microcontroller circuit design, tablet screen replacement, custom shoe painting, and creating latte art.

### Video Analysis

To cover a range of topics, we chose five different DIY domains (electronics/science, craft, home/repair, art, and food) from a popular DIY website[2]. We selected the first four videos on YouTube for each domain that satisfied a set of criteria chosen to ensure they were effective, including:

- Produced video: evidence of editing through cuts
- Camera angle: 1-2 static camera viewpoints
- Content: 1-2 instructors with audio narration
- Popularity: a minimum of 1000 views, with less than 10% dislikes from the total like-or-dislike ratings.
- Experience: authors with >5 published how-to videos.

20 videos from 20 distinct authors were coded in a week (Figure 2). The average length of these videos is 5 minutes and 5 seconds (*max*=9'08", *min*=1'54"), and the average view count is 269,426 (*max*=4,004,613, *min*=1,156). Although these tutorials cover various topics and tasks, we observed several common characteristics of the videos:

- **Narration.** All of the videos include narration that explains what is happening in the tutorial. Most authors seem to narrate during the demonstration (70%), while fewer authors record a separate voice-over track.
- **Speed-up effects.** Most videos (60%) include editing effects that speed up repetitive actions, such as screwing in fasteners or chopping vegetables. In many cases, authors break the sync between the audio and video tracks in these sped-up sections so that the narration plays continuously at normal speed with no long silences while the video plays at a faster speed.
- **Annotations.** Many videos (65%) include titles that add relevant information about the task, including descriptions of depicted objects or actions, measurements, elapsed time, and details that are not shown in the demonstration. Some videos also include other annotations (e.g., arrows, rectangular highlights) that emphasize important details.

This analysis suggests that authors apply a common set of editing techniques. Since the final videos convey limited information about the recording and editing processes, we interviewed several authors of the selected videos. We hope to learn what kind of footage they omitted and how much time they spent on editing. We also wanted to understand the rationale behind the edits.

| ID | Category | Experience | Videos | Sample Project |
|---|---|---|---|---|
| P1 | electronics | professional | 48* | Body-mounted camera |
| P2 | home/repair | amateur | 162 | Powder coating aluminum |
| P3 | science | professional | 45* | Develop caffenol film |
| P4 | home/repair | amateur | 717 | Snowblower repair |
| P5 | home/repair | amateur | 33 | Paintcan setup |
| P6 | electronics | amateur | 5 | On-beat disco light |

**Table 1. Background information about interview participants.** **** Numbers of videos published on personal YouTube channels, excluding those on the professional channels.***

### Interviews with Tutorial Authors

We contacted the 20 YouTube account holders of the analyzed videos, and interviewed the first six who responded (all males, ages 17 to 48). One of the YouTube user accounts corresponded to a 3-person team, which we counted as a single participant (P6). Among the participants, two were professional tutorial makers, while the rest were amateurs. For editing, three used Apple Final Cut Pro, two Corel VideoStudio, and one Adobe Premiere. Table 1 summarizes other information about the experience of the authors.

**Capture.** Except for the team (P6), all of the participants record demonstrations individually without any assistants. P2–P6 use a single video camera, while P1 uses an extra camera to capture closeup shots. To keep the recording process simple, the amateur authors tend to capture demonstrations in one uninterrupted take, narrating the action as they go. Naturally, such recordings often include mistakes (e.g., walking out of frame to retrieve a forgotten tool) and long, repetitive actions. In contrast, the professional authors create a script beforehand and record the narration separately.

**Editing.** All of the participants mentioned the importance of editing the final video tutorial down to a reasonable length (5–10 minutes). The goal is to provide enough information to understand the demonstration, but at the same time keep the video lively and interesting. P2 described his strategy as follows: *"If you can get rid of it and the video content still gets through, get rid of it"*; *"The way to make your video sizzle is to have good cuts, good points."* As a result, authors spend much of their editing time deciding on cuts, segmenting the video, removing and merging shots, and adding visual effects to speed up repetitive actions. They also take time to add subtitles and annotations. As P4 explained, *"the filming is the easiest part; it is the editing that's the challenge."* Overall, participants reported that filming time takes from one hour up to one day, and editing time typically takes 6–12 hours, depending on the the complexity of the project.

## DESIGN IMPLICATIONS

Based on our analysis of existing video tutorials and interviews with tutorial authors, we identified a few key aspects of the tutorial creation process that have important design implications for DIY video editing systems.

**Working with single take, single camera footage.** Most amateur authors record demonstrations in a single take with a a single camera. As a result, the captured footage often includes mistakes and long, repetitive actions.

[2] http://makezine.com/

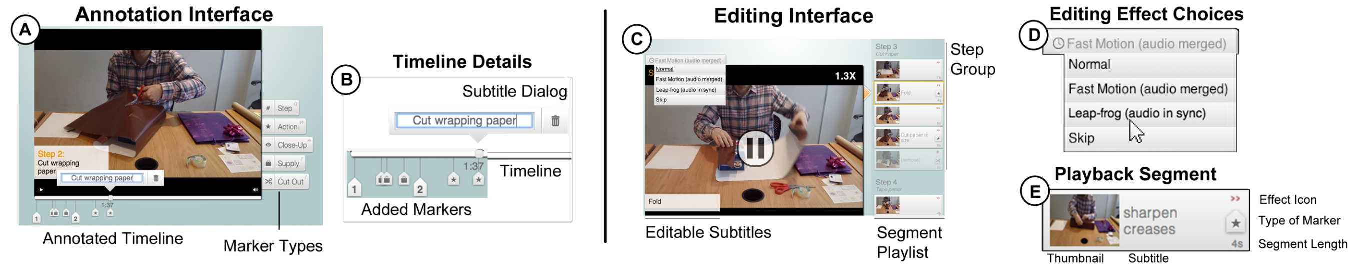

**Figure 3. Users first add markers to their recorded video in the Annotation Interface (A). Each marker can be labeled with a descriptive string (B). The Editing Interface shows automatically generated segments with effect suggestions (C). Users can change the effect (D) applied to each segment (E).**

**Making concise videos.** The most important design principle for creating effective DIY videos is to make them concise without sacrificing clarity. To this end, authors remove/condense unnecessary or repetitive actions so that the resulting video only contains salient footage.

**Retiming audio and video tracks separately.** One common technique for speeding up a video involves breaking the synchronization between the audio and video tracks so that they can be retimed separately. In cases where the narration refers to specific visual events, the tracks should remain aligned.

**Emphasizing important information.** Most effective DIY videos include titles, annotations and/or closeup views to emphasize relevant information and highlight key details.

**Focusing on high-level editing decisions.** Amateur users often struggle with low-level manipulation of cut points and timing in general-purpose video editors: A system should reduce the editing efforts and enable authors to focus on making simple choices for the final production.

We next describe how these considerations informed the design of DemoCut.

## AUTHORING VIDEOS WITH DEMOCUT

To enable amateur users to produce effective video tutorials, the DemoCut video authoring system semi-automatically edits a long, single take recording into meaningful steps. Early testing revealed that users find it easier to locate specific *moments* in the video than to mark or edit *segments*. Therefore, our Annotation Interface asks users to mark important moments. DemoCut combines the user annotations with audio and video analysis to automatically generate a segmented video with editing suggestions: It removes or condenses unnecessary/repetitive actions and enables flexible synchronization between audio and video tracks. Titles, visual annotations and closeup views are applied to enhance the content. Users can review and revise these decisions in the DemoCut Editing Interface. This section reviews DemoCut from the user's perspective (Figure 4). The following section will describe our video analysis pipeline.

### Annotating the Video

The purpose of the DemoCut Annotation UI is to collect high-level information that is difficult to extract automatically but useful in determining how to edit the video. We rely on users to distinguish important from unimportant actions and successful steps from mistakes. The user scrubs through the captured footage and adds markers for distinct moments, such as the instant when he cuts a sheet of paper (Figure 3A). DemoCut offers five types of markers for annotating a video:

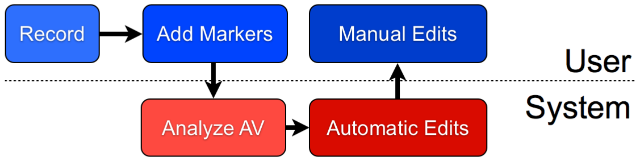

**Figure 4. DemoCut users first mark their recorded video in the Annotation Interface. DemoCut then segments their recording and suggests video edits, which users can change in the Editing Interface.**

- *Step*: indicates the start of a major part of the task
- *Action*: marks important moments
- *Closeup*: indicates moments where the action is happening in a small region of the video frame, e.g., for a detailed action such as fastening a small screw.
- *Supply*: indicates a tool or material used in the task
- *Cut-out*: indicates moments of the video that should be removed due to occlusion or a mistake in the performance.

This set of markers was derived from our observations of the structure of effective tutorial videos: actions are treated separately from supplies; zooming can direct the viewer's attention to a small area of the frame; and step divisions are used to divide actions into meaningful groups. Rather than specify start and end frames, users can place a marker on any frame of an important moment.

Users can add descriptions to markers (Figure 3B). These descriptions serve a dual purpose: they are used to generate automatic subtitles, and they are also shown as segment names in the Editing Interface to facilitate navigation. Users can also add visual highlights such as boxes and arrows to any marker.

### Automatic Video Editing

Based on the user's markers, DemoCut automatically segments the raw footage and applies editing effects.

#### *Temporal Effects*

We designed four temporal effects to shorten a video. In addition to skipping a segment or leaving it unchanged, we consider the synchronization between the audio and video tracks: People are sensitive to changes in speech playback speed, but video can often be accelerated without loss of clarity. Therefore, our temporal effects accelerate or contract video but keep audio at normal speed.

*Fast motion (with merged audio)*: When a segment includes several sections of narration with intermediate pauses, Demo-

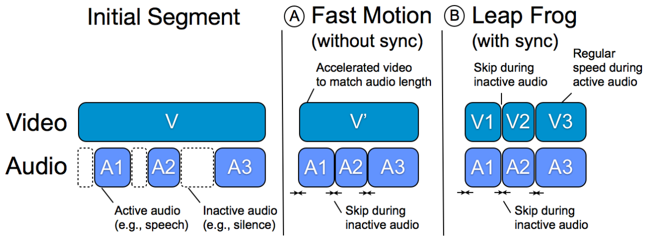

**Figure 5. DemoCut accelerates playback of video with intermittent audio narration through Fast Motion (A) and Leap Frogging (B).**

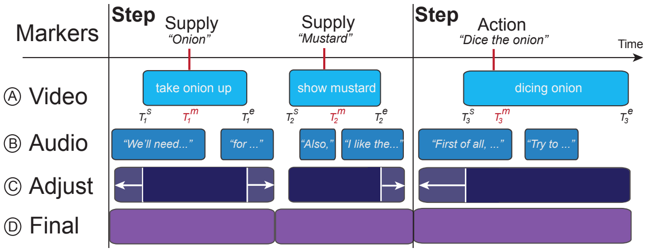

**Figure 6. Given user markers, DemoCut analyzes both video and audio to segment the demonstration video and apply editing effects.**

Cut removes the pauses and concatenates the audio segments. Then it speeds up the video so the total video length corresponds to the length of the concatenated audio (Figure 5A). This effect is appropriate if tight synchronization between audio and video is not required. For example, an author may describe general strategies for choosing supplies while measuring paper – here audio and video are independent of each other. In this case, DemoCut will accelerate the video to fit the length of the author's remarks.

*Leap frog (with synchronized audio)*: If synchronization between audio and video is necessary, this effect plays video and audio at normal speed during active audio segments, and skips video in the interstitial segments (Figure 5B). Synchronization is important if the author's face is in the shot (so lip movement and audio match), if actions produce distinct sounds (like cutting paper), or if the narration refers specifically to actions, e.g., when pointing at an object and describing its properties. Since DemoCut cannot automatically decide whether synchronization is necessary, it applies the Fast Motion effect by default but offers users control to change that effect.

*Skip:* Depending on the length of the removed segment, DemoCut either applies a fade through black (for segments up to 15 seconds); or a fade to a title that indicates how much time has passed (e.g., "2 minutes later").

If these temporal effects are not appropriate, DemoCut plays the audio and video at the captured rate. We call this the *Normal* effect.

### *Visual Effects*

In addition to manipulating time, DemoCut offers three visual effects to structure the video and to provide emphasis. These visuals appear for the duration of the segment DemoCut derived from the user's marker:

*Subtitles*: Text entered by the user in the marking phase is converted into automatic subtitles with two levels – a step heading that remains on screen for all segments within a step (e.g., "Wrapping the present"); and a subheading from individual event markers (e.g., "Sharpen creases").

*Automatic zoom*: When users create closeup markers, they also specify a rectangular region of interest. DemoCut automatically crops and enlarges this region of the segment.

*Visual annotation*: DemoCut overlays visual box or arrow annotation specified by the user in the marking stage.

### Reviewing and Editing

Since our automatic video and audio segmentation has a limited understanding of the video, it is likely that some editing decisions will be incorrect. For example, DemoCut's algorithms have no way of inferring whether audio-video synchronization will or will not be required in a given segment. In addition, automatic analysis may also lead to errors: if the narration is not correctly segmented, speech can be cut off mid-sentence. DemoCut's editor gives authors the opportunity to review and revise all editing decisions.

In the Editing Interface, the video is visualized as a set of segments (Figure 3E) flowing from top to bottom on the right side of the main video view (Figure 3C). There is no traditional timeline for two reasons: first, editing operations only apply to detected segments (we consciously prevent users from applying frame-level edits to keep with the goal of a semantic editor); second, because segments may come with labels entered by the user, a vertical layout makes it easier to read labels. Users can navigate to any segment by clicking on its thumbnail. Once selected, they can change which effect should be applied to a given segment (Figure 3D). Users can also modify any visual effects, to edit subtitles, resize the cropped region, or add/delete highlights. When satisfied with their choices, users can export a continuous video suitable for online video sharing platforms.

## AUTOMATIC EFFECT DECISION PIPELINE

DemoCut performs several automated steps to convert the user-annotated input recording into an edited video tutorial. First, the system segments the recording into regions around user-specified markers. This segmentation considers both the similarity of video frames around each marker and the presence of narration in the audio track in order to determine the appropriate segment boundaries (Figure 6). DemoCut then automatically applies an temporal and a visual effect to each segment based on the type of the corresponding user marker and the properties of the audio/video content in the segment. The rest of this section describes these steps in detail.

### Video Segmentation

Except for the step marker, all of the user-specified markers indicate important moments in the demonstration that correspond to some segment of the recording. In many cases, we can infer the duration of these segments by searching for video frames that look similar to the marked frame. For example, in Figure 7A, the similar frames before and after a supply marker show the author holding up a bottle of vinegar, and in Figure 7B, the similar frames around an action marker show the author grating cheese. For every marked frame $T^m$, DemoCut uses the following method to compute candidate start and end frames $T^s$ and $T^e$ for the corresponding segment. For the $i$-th marked frame $T_i^m$, our algorithm finds $T_i^s$ by comparing $T_i^m$ to earlier frames in the video until it reaches a previous marker at $T_{i-1}^m$, or until 5% of pixels (in grayscale) have changed by 20%. Similarly, the system finds $T_i^e$ by comparing $T_i^m$ to subsequent frames in the video. To optimize performance, DemoCut compares to frames sampled at 0.5 seconds and ignores overlaps between segments. Segment overlaps are resolved during boundary adjustment after incorporating the audio analysis.

### Adjusting Segments with Audio Analysis

Adjacent segments can have different effects that change how video and audio are processed. To prevent such changes from interfering with a video's narration, DemoCut adjusts segment boundaries to align with audio activity boundaries.

#### *Detecting non-silent sections*

Since many DIY videos include prominent non-speech sounds such as chopping noises, power tools, etc., detecting speech automatically is a challenging task. We found that even state-of-the-art speech detection algorithms produce poor results in many cases. As a result, we take a more conservative approach; DemoCut automatically detects non-silent sections in the recorded audio and treats the background sound as part of the narration.

At a high level, our algorithm for detecting non-silent sections works as follows. We compute the "loudness" of each audio window, organize the windows into a histogram based on loudness, and then analyze the histogram to determine a minimum loudness threshold for non-silent windows. We then apply this threshold to categorize all audio windows as silent or non-silent. Finally, we filter this categorization to eliminate very short sequences of silent or non-silent samples. Here, we describe these steps in more detail:

*Computing loudness.* Given an input audio waveform sampled at 44.1 kHz (Figure 8A), we estimate loudness by computing the root mean square (RMS) energy [25] across the entire waveform. The RMS energy for a window of size $n$ is $\sqrt{(\sum_n x_i^2)/n}$ where $x_i$ is the value of the $i$th audio sample in the window. We set window sizes as 0.1 second with $n = 4410$. Prior to computing RMS energy, the audio is normalized and noise-reduced with Adobe Audition.

*Computing loudness threshold.* After analyzing the RMS energy profiles of several different types of DIY videos, we found that the vast majority of recorded audio represents background sound, which tends to have similar and fairly low RMS energy values. In contrast, user narration varies from medium to high RMS values based on the speaker's distance to the microphone and the sensitivity of the recording device. Based on this observation, we first compute a histogram of RMS energy for all windows in the audio track; the windows that correspond to background sound form a large mass at the low-RMS end of the histogram (Figure 8B). To distinguish these "silent" parts of the recording from the narration, we smooth the histogram with a Gaussian kernel, find the minimum derivative point in the smoothed histogram, and set the loudness threshold $\varepsilon$ to be the RMS energy value at this elbow point. Figure 8B shows the RMS histogram and loudness threshold for one of our example videos, "How to make salad dressing."

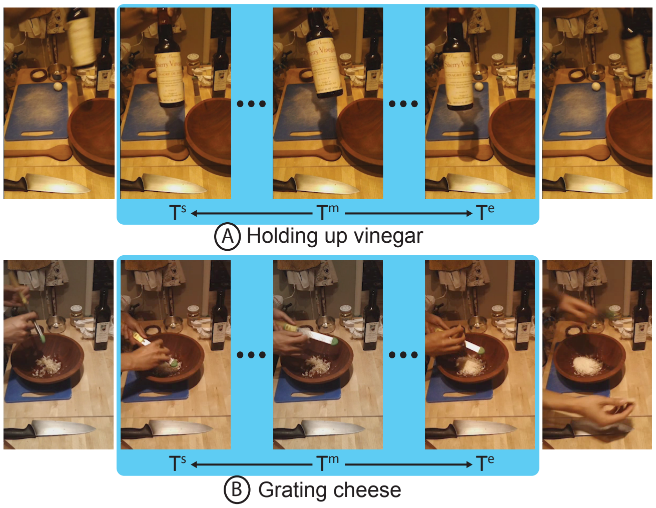

**Figure 7. DemoCut looks for similar video frames before and after a marked frame $T^m$ to find candidate start ($T^s$) and end ($T^e$) frames for the corresponding segment.**

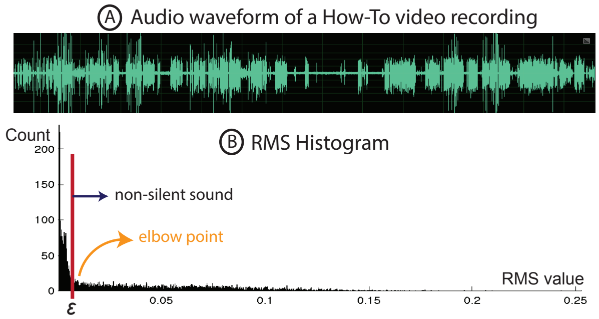

**Figure 8. We use RMS energy of the audio to find silent and non-silent regions. We determine the threshold for silence by analyzing the histogram of the RMS energy.**

*Categorizing silent/non-silent sections.* To partition the audio track into silent and non-silent sections, we first label each window as silent or non-silent based on $\varepsilon$. This initial labeling often includes some very short silent and non-silent sections. Since many short silent sections correspond to short pauses between spoken words, we turn any silent sections that are shorter than 0.4 seconds into non-silent sections. Then, we discard any non-silent sections that are shorter than 0.8 seconds to account for any clicks and pops in the recorded audio. The 0.4 and 0.8 second thresholds for silent and non-silent sections were tuned experimentally, and we used these parameter values for all of our results.

#### *Adjusting segment boundaries*

In order to avoid cutting off an author's narration, DemoCut adjusts the video segment boundaries using the non-silent sections of the audio track (Figure 6). First, for any segment we find all of the overlapping non-silent audio sections and

| Task | Category | Raw footage length | DemoCut video length | # of markers | # of seg-ments | Incorrect Effects | # of non-silent sections | Audio misses | Audio cut-off | Audio false-positives |
|---|---|---|---|---|---|---|---|---|---|---|
| A: Xbee tutorial | electronics | 7’01” | 3’27” | 16 | 30 | 0% | 79 | 5% | 0% | 0% |
| B: Paper pipe robot | craft | 10’55” | 4’40” | 18 | 30 | 20% | 77 | 21% | 12% | 0% |
| C: Ribbons for straps | craft | 10’03” | 4’23” | 39 | 46 | 7% | 72 | 15% | 7% | 0% |
| D: Fixing front light | repair | 6’32” | 2’12” | 21 | 33 | 9% | 40 | 10% | 3% | 0% |
| E: How to make grassy head | art | 9’28” | 5’29” | 29 | 44 | 5% | 86 | 8% | 2% | 0% |
| F: How to make potato stamps | art | 16’38” | 4’05 | 30 | 45 | 7% | 119 | 7% | 3% | 0% |
| G: How to make salad dressing | food | 14’46” | 5’38” | 33 | 39 | 13% | 121 | 6% | 2% | 0% |
| AVERAGE | - | 10’46” | 4’10” | 26.4 | 38.1 | 9% | 83.5 | 10.3% | 4.1% | 0% |

**Table 2. A list of how-to videos we recorded to assess the robustness of the DemoCut system.**

then grow the segment so that it completely contains all of these non-silent sections. Next, DemoCut resolves overlapping segments: If any two segments overlap, the boundaries must be readjusted. If the overlap region is silent, the region is split into two equal parts and each is assigned to the corresponding segment. If the overlap region includes a non-silent audio section, DemoCut assigns this non-silent section to the segment that has more overlap with the section. If the overlap for both video segments is the same, DemoCut assigns the section to the smaller video segment. Finally, DemoCut addresses any gaps between segments. If a gap is less than 2 seconds, it is merged to the shorter adjacent segment. Otherwise, DemoCut creates a new segment for the gap. Note that such *unmarked segments* do not have a corresponding marker, but they may still show useful details of the demonstration.

### Applying Effects

To automatically apply an effect to each computed segment, DemoCut first detects whether there is motion in the video. A segment is considered to be *static* (i.e., no motion) if less than 1% of pixels in the grayscale versions of consecutive frames have changed by more than 20%. To optimize for performance, the segment is sampled at 0.5 seconds for this comparison. DemoCut chooses effects as follows:

1. If the segment includes a *cutout* marker, apply *“Skip”*.
2. If the segment includes a *closeup* marker, apply *“Zoom”* to the entire segment.
3. If the segment includes any non-silent audio sections, apply *“Fast Motion”*.
4. If the segment is silent, static, and unmarked, apply *“Skip”*.
5. If the segment is silent but not static (either marked or unmarked), apply *“Normal”*.
6. For any marker with a text annotation, apply *“Subtitles”*.

## IMPLEMENTATION

The video and audio analysis is implemented in Matlab. The Annotation and Editing Interfaces are implemented with standard Web technologies (HTML5, CSS3, and JavaScript). An Apache web server hosts these web pages and sends the user annotations to the back-end Matlab system.

## EVALUATING AUTOMATIC EFFECT DECISION

To evaluate DemoCut’s analysis engine, we recorded seven how-to tasks from the five categories we selected in the formative user study (Table 2). The tasks were recorded by 4 people (all authors of this paper) in 7 locations using a Sony camcorder or an iPad with a video resolution of at least 640x480 pixels. We used DemoCut to annotate the recordings and then examined the automatically generated tutorials.

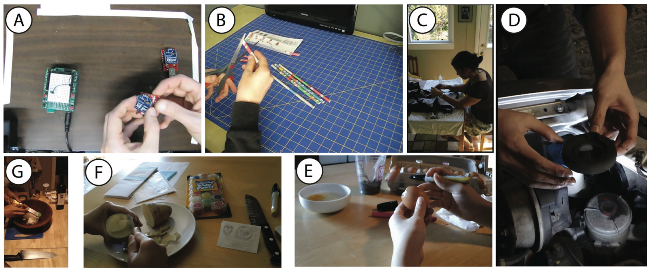

**Figure 9. Illustrative frames from the seven videos used to assess DemoCut. Labels correspond to task labels in Table 2.**

Overall, the resulting tutorials exhibit many of the desired characteristics outlined earlier in the paper. The automatically edited videos are concise: 2-5 minutes long and 2.5 times shorter than the original footage. In most cases, DemoCut successfully identified segments where the “Fast Motion” or “Skip” effects could be applied to condense the tutorial. For example, the edited salad dressing video uses “Fast Motion” to speed up repetitive actions like chopping an onion and grating cheese, and then skips the segment where the author leaves the frame to toast pine nuts. In addition, the automatically generated titles improve the clarity of the tutorials by adding valuable descriptions of steps, actions, supplies and indicating the elapsed time for skipped segments. In an electronics tutorial, titles like “sending data toggles LED” add important details that are not visible in the video.

There were some situations where the effects were not as successful. To get a more quantitative measure of DemoCut’s performance, we counted several types of errors in the automatically generated videos:

*Incorrect editing effects.* In a few cases, the “Fast Motion” effect is applied to segments where the audio track should actually be in sync with the visuals. Also, when markers are very close to one another in time, DemoCut sometimes generates very short segments where the editing effects are hard to see. We identify these cases as incorrect editing effects.

*Audio miss.* We refer to any piece of narration that is not detected as a non-silent section as a miss.

*Audio cut-off.* We refer to any detected non-silent section that cuts off narration by ending too early or starting too late as a cut-off error.

*Audio false-positive.* We refer to any non-silent section that is neither narration nor significant activity or background sound as a false-positive.

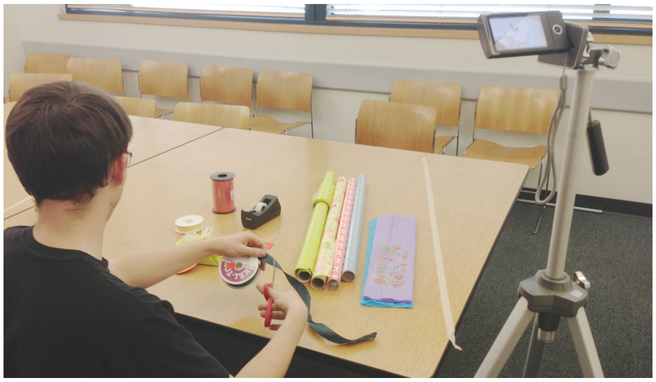

**Figure 10. Our user study setup**

We report the incorrect edits as a percentage of the total number of segments and the three audio errors as a percentage of the total number of ground-truth narration sections. Table 2 shows all of the results from our analysis. Overall, we found low average error rates (less than 11%) for all of these problems. Also, note that most of these errors can be fixed by changing the automatically applied editing effects in DemoCut's reviewing and editing interface.

## USER EVALUATION

To evaluate the usability and utility of DemoCut, we recruited 8 participants (4 males, ages 20-41) to create how-to video tutorials. We were especially interested in two questions: First, how much *effort* would participants have to invest to mark and edit their own tutorial videos with DemoCut? And second, what are the *qualities of the resulting videos* – both in terms of strengths and shortcomings?

### Task and Materials

The participants were asked to create a tutorial for wrapping and decorating a present. We chose this task because it is relatively simple but still involves multiple distinct activities and steps that can be accomplished in 5-15 minutes. Possible steps include: measuring the gift size, cutting paper and ribbons, folding and wrapping, and decorating the present with ornaments. We offered the following supplies:

- *Present*: a rectangular gift box of size $9 \times 2 \times 4.5$ inches.
- *Tools*: scissors, utility knife, ruler, pencil, double-sided tape, transparent tape, and glue.
- *Wrapping paper*: a variety of wrapping paper rolls including plain, patterned, and textured.
- *Decorations*: ribbons (curling and fabric) in multiple colors, gift bows, stickers, and message cards.

To help them understand the context of the study, the participants were asked to watch three videos before visiting our lab. The videos were selected from the formative user study.

### Procedure and Environment

The study was conducted in a quiet lab environment with static lighting. We used a tripod-mounted Sony camcorder to record the gift wrapping task (Figure 10), and a Macbook Pro running OS X and Google Chrome for DemoCut. The laptop was connected to a 30-inch monitor and external mouse and keyboard. Each study session lasted 60-90 minutes.

*Introduction (15 minutes).* The participants viewed a web-based tutorial that introduced the goal and procedure of the study. In the tutorial, the participants practiced annotating a one-minute demo video with five types of markers, reviewed a system-generated result, and modified video effects in the DemoCut Editing interface.

*Filming setup and practice (5 minutes).* The participants were asked to plan their gift-wrapping demonstration with any of the provided supplies. The camcorder was positioned either opposite the participants or behind them on the right and was angled down to capture their workspace on a conference table. The participants reviewed the camera's point of view and were told that any activities outside of the delineated workspace would not be captured. The participants were free to plan their task on paper or conduct a practice run.

*Filming demonstration (10-20 minutes).* The participants filmed their gift wrapping demonstration in a single take. The study moderator initiated and terminated the recording, but did not provide additional assistance.

*Annotating and editing (30-45 minutes).* The participants annotated their video with the DemoCut Annotation Interface and modified the generated video tutorial using the DemoCut Editing Interface.

*Review of the final video and discussion (10 minutes).* Finally, participants reviewed the final video, completed a questionnaire, and discussed the process with experimenters.

## FINDINGS AND DISCUSSION

All of the participants successfully created a complete video tutorial of their gift wrapping technique using DemoCut during the study session. The average length of the recorded demonstrations was 8 minutes, and the final generated videos were just over 5 minutes long (45% shorter than the raw footage). On average, participants spent 15 minutes annotating the recordings and added 33 markers. Table 3 summarizes several other quantitative results from the study.

Here, we summarize a few key points from the responses to the questionnaire and the end-of-study discussions:

**DemoCut interface and workflow.** We received strong positive feedback about the DemoCut interface as a whole and the editing workflow that it enables. All participants agreed or strongly agreed that it was easy to annotate a recording using the Annotation Interface, and seven of them found it easy to use the reviewing and Editing Interface. P1 explained, *"This is very simple for beginning users and takes out some of the guess work around learning how to use different layers, speed effects, etc."*, and P2 described the workflow as, *"super easy and SUPER FAST!"* P6 also appreciated the simplicity of the interface: *"I like this a lot because there aren't thousands of different buttons to work with."* In addition, several participants noted how the automated components of the system reduced the amount of effort required to create an edited video: *"I could be lazier and still have a great video cause it did everything for me"* (P8), and *"Pre-segmentation (when it worked well) made it easy to zero in on the portion I wanted to modify"* (P4).

| ID | Editing expertise (years) | Footage length | DemoCut video length | Final video length | Annotation time (mins) | # and types of markers used for annotation | | | | | | Ave text length (words) | # of segments | Review & edit time (mins) | # of effects changed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Total | Step | Action | Supply | Closeup | Cutout | | | | |
| P1 | 5 | 3’51” | 2’14” | 2’14” | 10 | 20 | 3 | 10 | 5 | 1 | 1 | 3.2 | 28 | 8 | 1 |
| P2 | 3 | 7’16” | 4’09” | 4’14” | 16 | 29 | 4 | 16 | 4 | 5 | 0 | 4 | 49 | 11 | 4 |
| P3 | 10 | 10’57” | 6’45” | 6’45” | 18 | 36 | 10 | 18 | 4 | 0 | 4 | 2.2 | 57 | 10 | 3 |
| P4 | 2 | 9’16” | 5’12” | 5’38” | 25 | 35 | 6 | 20 | 3 | 4 | 2 | 3.3 | 56 | 13 | 6 |
| P5 | 0 | 7’17” | 4’13” | 4’07” | 17 | 38 | 5 | 18 | 7 | 6 | 2 | 3 | 56 | 12 | 5 |
| P6 | 0 | 6’28” | 3’20” | 2’48” | 8 | 24 | 3 | 13 | 6 | 0 | 2 | 3.5 | 40 | 9 | 9 |
| P7 | 0 | 10’08” | 6’18” | 6’02” | 21 | 66 | 7 | 35 | 12 | 8 | 4 | 3.9 | 92 | 14 | 20 |
| P8 | 0 | 8’58” | 3’05” | 3’03” | 8 | 13 | 4 | 6 | 1 | 0 | 2 | 3.6 | 21 | 10 | 2 |
| AVE | 2.5 | 8’01” | 4’24” | 4’21” | 15 | 33 | 5 | 17 | 5 | 3 | 2 | 3.3 | 50 | 11 | 6 |

**Table 3. Quantitative analysis of the user evaluation.**

**Automatic editing effects.** In general, the participants liked how DemoCut automatically removed or condensed parts of their recordings. Their feedback suggests that the automatically generated effects were particularly useful for speeding up repetitive actions like cutting and folding and skipping extraneous actions, such as removing the adhesive sticker from a bow. As P3 noted, these effects were generally successful because DemoCut *“correctly understood parts with no speech but long actions.”* Another participant commented specifically on the fast motion with merged audio effect, and said *“(I) appreciate the automatic speeding up/slowing down of video to match speech.”*

**Reviewing and editing.** As expected, there were some cases where participants decided to modify the automatic effects. Errors in the audio analysis can cause the narration within a segment to get cut off when fast motion effects are applied. To eliminate these audio artifacts, participants changed the segment effect from fast motion to normal mode. In cases where the narration referred to specific visual events, participants switched from the default fast motion with merged audio effect to leap frog with synchronized audio. Finally, in a few situations, participants decided to skip an annotated segment that they deemed unnecessary or unclear after reviewing the rest of the tutorial.

**Quality of generated tutorials.** Five of the eight participants said they were satisfied or very satisfied with the video tutorials that they created with DemoCut during the study. The remaining three participants had significantly more video editing experience, and they wanted to further refine their tutorials by adjusting some of the timing and cut points using more traditional low-level editing tools. However, even these participants agreed that DemoCut was *“good for a first pass of editing”* and provided *“helpful “smart” suggestions”* even though the system is *“limited in manual control.”*

**Default speed-up effects.** The participants noted some limitations with the default editing effects. P5 explained that *“having the speed up of video be the default speed creates a stressful tutorial.”* Some participants pointed out that there are some obvious cases where fast motion with merged audio should not be applied; for example, the effect *“does not work well if the person’s face is showing (the speech and mouth movements would not match up).”* We agree with these comments and plan to use face detection and add an adaptive learner to improve the system.

**Annotation guidelines.** One observation from the study is that adding too many markers during the annotation phase can hurt the quality of the generated tutorial. Adding markers temporally close together leads to many short segments, and since DemoCut applies a video edit effect to each segment individually, the resulting tutorial may end up transitioning rapidly through several inconsistent effects (e.g., fast motion effects with various playback speeds). One way to address this problem is to make automatic editing decisions that span several consecutive segments. The participants offered a few other suggestions: P4 wonders *“if there are simple tips you could give to the user while recording that would make them more successful,”* and P8 suggested that seeing real-time effects while adding markers might help him understand how best to annotate the recording.

## LIMITATIONS AND FUTURE WORK

Our implementation is based on several simplifying assumptions that limit generality. We assume a single, static camera position that shows all relevant actions and a quiet indoor environment with constant lighting and little background noise. In order to detect static shots that should be skipped, our video analysis assumes a static background. Our audio analysis assumes that all non-silent sections of audio are narration, but this may not always be the case. Loud non-speech sounds, such as chopping or the sound of a sewing machine, can lead to errors in our editing effect decisions.

As was pointed out by several of our study participants, making effect decisions individually for each segment can lead to inconsistencies in playback speed as the video transitions from segment to segment. A more global approach that looks at all video effects together and enforces smooth transitions between adjacent segments would help address some of these artifacts. In addition to addressing these limitations, we see several promising directions for future work.

**Multiple camera footage.** We designed DemoCut to work with footage from a single, static camera. One interesting avenue for future work is to consider footage from multiple cameras. Prior work has compared different camera views capturing physical tasks for remote collaboration [12, 28]. Similarly, DemoCut could try to automatically select the best view for each segment based on user annotations as well as the video content (e.g., choosing a zoomed view for closeups, switching to a different view when there are occlusions).

**Support viewer's learning.** In this work, we focus on producing well-edited video tutorials. However, we could also imagine generating different output formats, including indexed videos, step-by-step instructions, or mixed media tutorials, similar to those presented by Chi et al. [9]. Another natural extension would be to develop interactive components that monitor user actions and provide realtime guidance and feedback for general DIY tasks. Follow-up studies to understand viewer's learning experience would be useful for refining the automatic editing effects and interactive design.

**Generalize to other instructional video domains.** One exciting direction is to explore other areas where our techniques could be applied, such as software learning, music instruction, and video lectures. Each domain may require slightly different analysis and segmentation rules. For example, the system could use a log of executed operations to adjust segment boundaries for software tutorials, or incorporate pitch detection when analyzing music instruction.

## CONCLUSION

In this paper, we presented DemoCut, a semi-automatic video editing system that helps users create clear and concise video tutorials of DIY tasks. The key idea behind our approach is to combine rough user annotations with simple video and audio analysis techniques in order to segment the input recording and apply appropriate editing effects. Our small user evaluation suggests that video authors are able to create effective video tutorials using DemoCut, and the qualitative feedback includes encouraging positive reactions to the annotation and editing workflow, as well as the automatic editing effects.

## ACKNOWLEDGMENTS

Work at Berkeley was supported by Adobe and a Berkeley Fellowship for Graduate Studies. We thank the YouTube users (in alphabetical order) *donyboy73, Griffin Hammond on Indy Mogul, John NYCCNC, Matt Richardson on MAKE, mjfpieters, and TheMuskokaPainter* for sharing their insights on DIY tutorials in our interviews.

## REFERENCES

1. Adams, B., and Venkatesh, S. Situated event bootstrapping and capture guidance for automated home movie authoring. In *Proceedings of MULTIMEDIA*, ACM Press (2005), 754–763.
2. Bai, J., Agarwala, A., Agrawala, M., and Ramamoorthi, R. Selectively de-animating video. *ACM Trans. Graph. 31*, 4 (2012), 66:1–66:10.
3. Barnes, C., Goldman, D. B., Shechtman, E., and Finkelstein, A. Video tapestries with continuous temporal zoom. *ACM Trans. Graph. 29* (2010), 89:1–89:9.
4. Bergman, L., Castelli, V., Lau, T., and Oblinger, D. Docwizards: a system for authoring follow-me documentation wizards. In *Proceedings of UIST*, ACM Press (2005), 191–200.
5. Bernstein, M. S., Brandt, J., and Miller, R. C. Crowds in two seconds. *Proceedings of UIST* (2011).
6. Berthouzoz, F., Li, W., and Agrawala, M. Tools for placing cuts and transitions in interview video. *ACM Trans. Graph. 31*, 4 (2012), 67:1–67:8.
7. Carter, S., Adcock, J., Doherty, J., and Branham, S. Nudgecam: toward targeted, higher quality media capture. In *Proceedings of MULTIMEDIA*, ACM Press (2010), 615–618.
8. Casares, J., Long, A. C., Myers, B. A., Bhatnagar, R., Stevens, S. M., Dabbish, L., Yocum, D., and Corbett, A. Simplifying video editing using metadata. In *Proceedings of DIS*, ACM Press (2002), 157.
9. Chi, P.-y., Ahn, S., Ren, A., Dontcheva, M., Li, W., and Hartmann, B. MixT: automatic generation of step-by-step mixed media tutorials. In *Proceedings of UIST*, ACM Press (2012), 93.
10. Davis, M., Heer, J., and Ramirez, A. Active capture: automatic direction for automatic movies. In *Proceedings of MULTIMEDIA*, ACM Press (2003), 88.
11. Diakopoulos, N., and Essa, I. Videotater: an approach for pen-based digital video segmentation and tagging. In *Proceedings of UIST*, ACM Press (2006), 221–224.
12. Fussell, S. R., Setlock, L. D., and Kraut, R. E. Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proceedings of CHI*, ACM Press (2003).
13. Grabler, F., Agrawala, M., Li, W., Dontcheva, M., and Igarashi, T. Generating photo manipulation tutorials by demonstration. *SIGGRAPH* (2009).
14. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories. In *Proceedings of UIST*, ACM Press (2010).
15. Gupta, A., Fox, D., Curless, B., and Cohen, M. DuploTrack: a real-time system for authoring and guiding duplo block assembly. In *Proceedings of UIST*, ACM Press (2012), 389–402.
16. Gurevich, P., Lanir, J., Cohen, B., and Stone, R. TeleAdvisor: a versatile augmented reality tool for remote assistance. In *Proceedings of CHI*, ACM Press (2012).
17. Heck, R., Wallick, M., and Gleicher, M. Virtual videography. *ACM Trans. Multimedia Comput. Commun. Appl. 3*, 1 (2007).
18. Heer, J., Good, N. S., Ramirez, A., Davis, M., and Mankoff, J. Presiding over accidents: system direction of human action. In *Proceedings of CHI*, ACM Press (2004), 463–470.
19. Henderson, S., and Feiner, S. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Trans on Visualization and Computer Graphics 17*, 10 (2011), 1355–1368.
20. Joshi, N., Mehta, S., Drucker, S., Stollnitz, E., Hoppe, H., Uyttendaele, M., and Cohen, M. Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of UIST*, ACM Press (2012), 251–260.
21. Lafreniere, B., Bunt, A., Lount, M., Terry, M., and Cowan, D. Looks cool, I'll try this later!": Understanding the faces and uses of online tutorials. *University of Waterloo Tech Report* (2012).
22. Liu, F., Gleicher, M., Wang, J., Jin, H., and Agarwala, A. Subspace video stabilization. *ACM Trans. Graph. 30*, 1 (2011), 4:1–4:10.
23. Mackay, W. E. Eva: an experimental video annotator for symbolic analysis of video data. *SIGCHI Bull. 21*, 2 (1989), 68–71.
24. Müller, E. Where quality matters: discourses on the art of making a YouTube video. In *The YouTube Reader*, Stockholm: National Library of Sweden (2009).
25. Panagiotakis, C., and Tziritas, G. G. A speech/music discriminator based on RMS and zero-crossings. *IEEE Transactions on Multimedia 7*, 1 (2005), 155–166.
26. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. F. Pause-and-play: automatically linking screencast video tutorials with applications. In *Proceedings of UIST*, ACM Press (2011), 135–144.
27. Pritch, Y., Ratovitch, S., and Hendel, A. Clustered synopsis of surveillance video. In *Proceedings of AVSS*, IEEE Computer Society (2009).
28. Ranjan, A., Birnholtz, J. P., and Balakrishnan, R. Dynamic shared visual spaces: experimenting with automatic camera control in a remote repair task. In *Proceedings of CHI*, ACM Press (2007), 1177–1186.
29. Torrey, C., Churchill, E. F., and McDonald, D. W. Learning how: The search for craft knowledge on the Internet. In *Proceedings of CHI*, ACM Press (2009), 1371–1380.
30. Torrey, C., McDonald, D. W., Schilit, B. N., and Bly, S. How-To pages: Informal systems of expertise sharing. In *Proceedings of ECSCW*, Springer London (2007), 391–410.