Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos

Juho Kim[1] **Phu Nguyen**[1] **Sarah Weir**[1] **Philip J. Guo**[2] **Robert C. Miller**[1] **Krzysztof Z. Gajos**[3]

[1]MIT CSAIL
Cambridge, MA USA
{juhokim, phun, sweir, rcm}@mit.edu

[2]University of Rochester
Rochester, NY USA
pg@cs.rochester.edu

[3]Harvard SEAS
Cambridge, MA USA
kgajos@eecs.harvard.edu

ABSTRACT

Millions of learners today use how-to videos to master new skills in a variety of domains. But browsing such videos is often tedious and inefficient because video player interfaces are not optimized for the unique step-by-step structure of such videos. This research aims to improve the learning experience of existing how-to videos with *step-by-step annotations*.

We first performed a formative study to verify that annotations are actually useful to learners. We created ToolScape, an interactive video player that displays step descriptions and intermediate result thumbnails in the video timeline. Learners in our study performed better and gained more self-efficacy using ToolScape versus a traditional video player.

To add the needed step annotations to existing how-to videos at scale, we introduce a novel crowdsourcing workflow. It extracts step-by-step structure from an existing video, including step times, descriptions, and before and after images. We introduce the Find-Verify-Expand design pattern for temporal and visual annotation, which applies clustering, text processing, and visual analysis algorithms to merge crowd output. The workflow does not rely on domain-specific customization, works on top of existing videos, and recruits untrained crowd workers. We evaluated the workflow with Mechanical Turk, using 75 cooking, makeup, and Photoshop videos on YouTube. Results show that our workflow can extract steps with a quality comparable to that of trained annotators across all three domains with 77% precision and 81% recall.

Author Keywords

Crowdsourcing; how-to videos; video annotation.

ACM Classification Keywords

H.5.2 User Interfaces: Graphical User Interfaces

INTRODUCTION

How-to videos on the web have enabled millions of learners to acquire new skills in procedural tasks such as folding origami, cooking, applying makeup, and using computer software. These videos have a unique step-by-step structure, which encourages learners to sequentially process and perform steps in the procedure [29]. While most text- and image-based tutorials (e.g., webpages) are naturally segmented into distinct steps, how-to video tutorials often contain a single continuous stream of demonstration. Because comprehensive and accurate step-by-step information about the procedure is often missing, accessing specific parts within a video becomes frustrating for learners. Prior research shows that higher interactivity with the instructional content aids learning [13, 28], and that the completeness and detail of step-by-step instructions are integral to task performance [11].

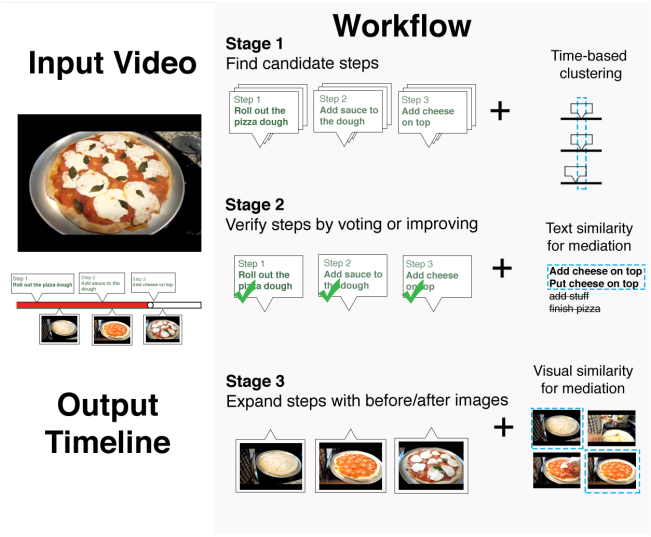

Figure 1. Our crowdsourcing workflow extracts step-by-step information from a how-to video with their descriptions and before/after images. It features the Find-Verify-Expand design pattern, time-based clustering, and text/visual analysis techniques. Extracted step information can be used to help learners navigate how-to videos with higher interactivity.

To better understand the role of step-by-step information in how-to videos, we ran a formative study where learners performed graphical design tasks with how-to videos. For this study, we designed ToolScape, an interactive how-to video player that adds step descriptions and intermediate result thumbnails to the video timeline. Learners using ToolScape showed a higher gain in self-efficacy and rated the quality of their own work higher, as compared to those using an ordinary video player. Moreover, external judges gave higher ratings to the designs produced by learners using ToolScape.

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.

ACM 978-1-4503-2473-1/14/04..$15.00.
http://dx.doi.org/10.1145/2556288.2556986

Providing such navigation support for how-to videos requires extracting step-by-step information from them. One solution is to ask instructors to include this information at tutorial generation time, but this adds overhead for instructors and does not solve the problem for existing videos. Another approach uses automatic methods such as computer vision. Previous research [3, 26] has shown success in limited domains with extensive domain-specific customization. When working with "videos in the wild", however, vision-based algorithms often suffer from low-resolution frames and a lack of training data. A scalable solution applicable beyond limited task domains and presentation formats is not yet available.

To address the issues of high cost or limited scalability with existing methods, we introduce a crowdsourcing workflow for annotating how-to videos, which includes the Find-Verify-Expand design pattern shown in Figure 1. It collects step-by-step information from a how-to video in three stages: (1) find candidate steps with timestamps and text descriptions, (2) verify time and description for all steps, and (3) expand a verified step with before and after images. The workflow does not rely on domain-specific knowledge, works on top of existing videos, and recruits untrained, non-expert crowd workers. For quality control, the workflow uses time-based clustering, text processing, and visual analysis to merge results and deal with noisy and diverse output from crowd workers.

To validate the workflow with existing how-to videos, we asked crowd workers on Mechanical Turk to annotate 75 YouTube how-to videos spanning three domains: cooking, makeup, and graphics editing software. Results show that the crowd workflow can extract steps with 77% precision and 81% recall relative to trained annotators. Successfully extracted steps were on average 2.7 seconds away from ground truth steps, and external evaluators found 60% of before and after images to be accurately representing steps.

The contributions of this paper are as follows:

- A how-to video player interface and experimental results showing that increased interactivity in a video player improves learners' task performance and self-efficacy.
- A domain-independent crowd video annotation method and the Find-Verify-Expand design pattern for extracting step-by-step task information from existing how-to videos.
- A novel combination of time-based clustering, text processing, and visual analysis algorithms for merging crowd output consisting of time points, text labels, and images.
- Experimental results that validate the workflow, which fully extracted steps from 75 readily available videos on the web across three distinct domains with a quality comparable to that of trained annotators.

RELATED WORK

We review related work in crowdsourcing workflows, video annotation, and tutorials.

Crowdsourcing Workflows

Research on multi-stage crowd workflows inspired the design of our method. Soylent [5] has shown that splitting tasks into the Find-Fix-Verify stages improves the quality and accuracy of crowd workers' results. Other multi-stage crowdsourcing workflows were designed for nutrition information retrieval from food photos [24], activity recognition from streaming videos [21], and search engine answer generation [6]. These applications demonstrated that crowdsourcing can yield results comparable to those of experts at lower cost. Our work contributes to this line of research a novel domain, video annotation, by extending [18] and [23].

Video Annotation Methods

This work focuses on providing a scalable and generalizable video annotation solution without relying on trained annotators, experts, or video authors. Video annotation tools capture moments of interest and add labels to them. Many existing tools are designed for dedicated annotators or experts in limited context. Domain-specific plug-ins [8, 14, 15] automatically capture task information, but require direct access to internal application context (e.g., Photoshop plug-ins accessing operation history). But plug-ins do not exist for most procedural tasks outside of software applications (e.g., makeup), which limits the applicability of this method.

Crowdsourcing video annotation has recently gained interest as a cost-effective method without relying on experts while keeping humans in the loop. Existing systems were designed mostly to collect training data for object recognition [31], motion tracking [30], or behavior detection [25]. Rather than use crowdsourcing to support qualitative researchers, this work supports end users learning from videos. Adrenaline [4] uses crowdsourcing to find the best frame from a video in near real-time. While [4] and our work both aim to detect a time-specific event from a video stream, our work additionally labels the event and expands to capture surrounding context.

Interactive Tutorials

In designing user interfaces for instructional videos, higher interactivity with the content has been shown to aid learning [13, 28]. Tversky et al. [28] state that "stopping, starting and replaying an animation can allow reinspection", which in turn can mitigate challenges in perception and comprehension, and further facilitate learning. Semantic indices and random access have been shown to be valuable in video navigation [32, 22], and the lack of interactivity has been deemed a major problem with instructional videos [16]. This work introduces a user interface for giving learners more interactivity in video navigation, and a crowdsourcing method for acquiring metadata handles to create such an interface at scale.

Recent systems create interactive tutorials by either automatically generating them by demonstration [8, 14], connecting to examples [26], or enhancing the tutorial format with annotated information [8, 9, 18, 20]. Our crowdsourcing workflow can provide annotations required to create these interfaces and further enable new ways to learn from tutorials.

EFFICACY AND CHALLENGES OF VIDEO ANNOTATIONS

To motivate the design of our crowdsourcing workflow for how-to video annotations, we first performed a formative study to (1) verify that annotations are actually useful to

Figure 2. Progress in many how-to videos is visually trackable, as shown in screenshots from this Photoshop how-to video. Adding step annotations to videos enables learners to quickly scan through the procedure.

Figure 3. How-to videos often contain a series of task steps with visually distinct before and after states. Here the author applied the "Gradient map" tool in Photoshop to desaturate the image colors.

learners, and (2) reveal the challenges of manually annotating videos and show the need for a more scalable technique.

Annotations on How-To Videos

How-to videos often have a well-defined step-by-step structure [15]. A *step* refers to a low-level action in performing a procedural task. Literature on procedural tasks suggests that step-by-step instructions encourage learners to sequentially process and perform steps in the workflow [29] and improve task performance [11]. Annotations can make such structure more explicit. In this paper, we define *annotation* as the process of adding step-by-step information to a how-to video. In determining which information to annotate, we note two properties of procedural tasks. First, for many domains, task states are visually distinct in nature, so progress can be visually tracked by browsing through a video (Figure 2). Examples include food in cooking videos, a model's face in makeup videos, and an image being edited in Photoshop videos. Second, how-to videos contain a sequence of discrete steps that each advance the state of the task (Figure 3). Our annotation method uses these two properties to accurately capture a sequence of steps, extracting timestamps, textual descriptions, and before and after images for each step.

We manually created a corpus of annotations for 75 how-to videos in three procedural task domains: cooking, applying makeup, and using Photoshop. We used this corpus to create our interface in the formative study, and ground truth data for evaluating our crowdsourcing workflow. We collected videos from YouTube's top search results for "[domain] [task name]" (e.g., "cooking samosa", "Photoshop motion blur").

Annotation-Aware Video Player: ToolScape

To display step annotations, we created a prototype video player named ToolScape. ToolScape augments an ordinary web-based video player with a rich timeline containing links to each annotated step and its respective before and after thumbnail images (Figure 4). ToolScape is a Javascript library that manages a timestamped list of steps and before/after images, which can connect to any embedded video player with a "play from this time point" Javascript API call.

In the timeline, the top and bottom streams represent annotated steps and thumbnail images from the video, respectively (Figure 4(a), (c)). Clicking on a step or image moves the

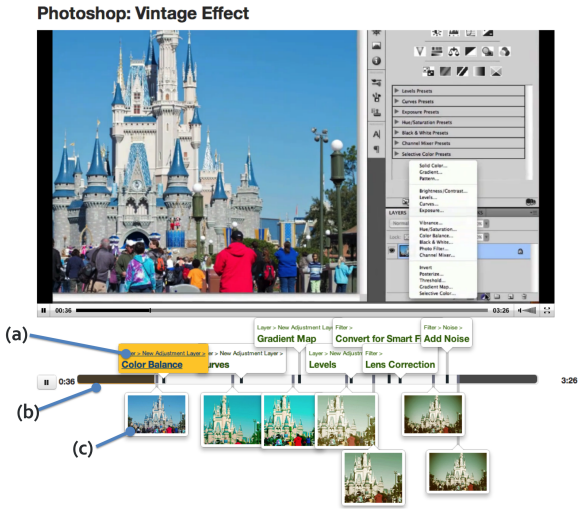

Figure 4. ToolScape augments a web-based video player with an interactive timeline. Annotations are shown above the timeline (a), screenshots of intermediate states are shown below the timeline (c), and the gray regions at both ends (b) show "dead times" with no meaningful progress (e.g., waiting for Photoshop to launch).

video player's slider to 5 seconds before the moment it occurred. The 5-second buffer, determined from pilot testing, helps learners catch up with the context preceding the indicated moment. Finally, ToolScape supports annotations of "dead times" at the beginning and end of videos (Figure 4(b)), which often contain introductory or concluding remarks. Pilot user observations showed that learners often skip to the main part of the tutorial. In our manually annotated video corpus, on average, 13.7% of time at the beginning and 9.9% at the end were "dead times" with no task progress.

Formative Study Design

To assess the effects of step annotations, we ran a formative study on novice Photoshop learners watching how-to videos on image manipulation tasks. We compared the experiences of learners using ToolScape and a baseline video player without the interactive timeline. We hypothesized that interacting with step annotations provided by ToolScape improves both task performance and learner satisfaction. Specifically:

H1 Learners complete design tasks with a higher self-efficacy gain when watching how-to videos with ToolScape.

H2 Learners' self-rating of the quality of their work is higher when watching with ToolScape.

H3 Learners' designs when watching with ToolScape are rated higher by external judges.

H4 Learners show higher satisfaction with ToolScape.

H5 Learners perceive design tasks to be easier when watching with ToolScape.

In addition to external ratings (H3), our measures of success include self-efficacy (H1) and self-rating (H2). In the context of how-to videos, these measures are more significant than

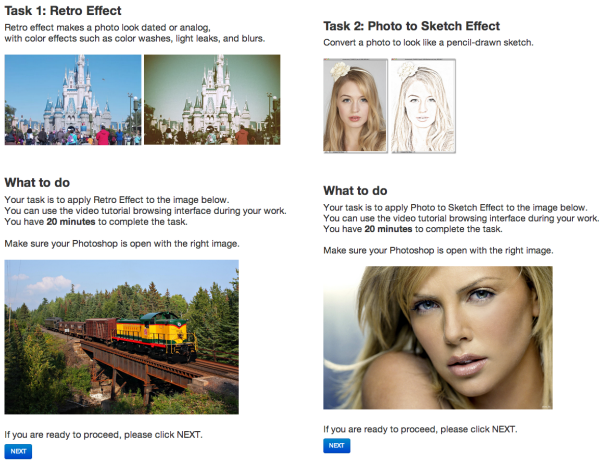

Figure 5. These task instructions are shown before the participant starts working on their image manipulation task. It includes a description of the effect to be implemented and a before-and-after example image pair.

just user preference. Educational psychology research shows that self-efficacy, or confidence in application of skills, is an effective predictor of motivation and learning [2, 33]. Positive self-rating has also been shown to accurately predict learning gains [27]. Finally, we chose not to count errors made in repeating tutorial steps as in [8], because our goal was to help users explore and learn new skills in open-ended design tasks.

Participants: We recruited twelve participants through university mailing lists and online community postings. Their mean age was 25.2 (σ = 3.2), with 8 males and 4 females. Most rated themselves as novice Photoshop users, but all had at least some experience with Photoshop. They received $30 for up to two hours of participation, on either a Mac or PC.

Tasks and Procedures: Our study had 2 x 2 conditions: two tasks each using ToolScape and baseline video players. We used a within-subject design with interface, task, and order counterbalanced. Each participant performed two image manipulation tasks in Photoshop: applying retro effect and transforming a photo to look like a sketch. In both interface conditions, we provided participants with the same set of how-to videos; the interface was the only difference. In addition, we disallowed searching for other web tutorials to ensure that any effect found in the study comes from the interaction method, not the content.

After a tutorial task covering all features of the video player interface, we asked participants self-efficacy questions adapted from Dow et al. [10], whose study also measured participants' self-efficacy changes in a design task. The questions asked: On a scale of 1 (not confident at all) to 7 (very confident), how confident are you with...

- solving graphic design problems?
- understanding graphic design problems?
- applying design skills in practice?
- incorporating skills from video tutorials in your design?

Next, participants attempted two 20-minute image manipulation tasks in Photoshop, with instructions shown in Figure 5. Participants could freely browse and watch the 10 how-to videos we provided (with annotations in the ToolScape condition). After each task, we asked questions on task difficulty, self-rating, and interface satisfaction. We also asked the self-efficacy questions again to observe any difference, followed by a 15-minute open-ended interview.

Finally, we asked four external judges to evaluate the quality of all transformed images by ranking them, blind to user and condition. They ranked the images from best to worst, based on how well each participant accomplished the given task.

Formative Study Results

H1 (higher self-efficacy for ToolScape) is supported by our study. For the four self-efficacy questions, we take the mean of the 7-point Likert scale ratings as the self-efficacy score. The participants' mean initial score was 3.8; with the baseline video player, the score after the task was 3.9 (+0.1) whereas with ToolScape the score was 5.2 (+1.4), which meant that learners felt more confident in their graphical design skills after completing tasks with ToolScape. (For H1, H2, and H4, differences between interfaces were significant at $p<0.05$ using a Mann-Whitney U test.)

H2 (higher self-rating for ToolScape) is supported. Participants rated their own work quality higher when using ToolScape (mean rating of 5.3) versus baseline (mean of 3.5).

H3 (higher external judge rating for ToolScape) is supported. The overall ranking was computed by taking the mean of the four judges' ranks. The mean rankings (lower is better) for output images in the ToolScape and Baseline conditions were 5.7 and 7.3, respectively. A Wilcoxon Signed-rank test indicates a significant effect of interface (W=317, Z=-2.79, $p<0.01$, r=0.29). Furthermore, nine of the twelve participants produced higher-rated images with ToolScape. The ranking method yielded high inter-rater reliability (Krippendorff's alpha=0.753) for ordinal data.

H4 (higher satisfaction with ToolScape) is supported. Mean ratings for ToolScape and Baseline were 6.1 and 4.5, respectively.

H5 (easier task difficulty perception for ToolScape) is not supported: The mean ratings for ToolScape and Baseline were 4.0 and 3.7, respectively. Combined with H2 and H3, this might indicate that participants did not find the tasks easier yet still produced better designs with greater confidence.

In conclusion, ToolScape had a significant effect on learners' belief in their graphical design skills and output quality. They also produced better designs as rated by external judges. Note that participants were watching the same video content in both conditions. Thus, the video annotation browsing interface affected design outcomes. Participants especially enjoyed being able to freely navigate between steps within a video by clicking on annotations.

Lessons for Video Browsing Interfaces

The features of ToolScape that provided higher interactivity and non-sequential access were highly rated and frequently used. In participants' responses to the 7-point Likert scale

questions on the usability of interface features, the time-marked image thumbnails (6.4) and step links (6.3) were among the highest rated, as well as the graying out of "dead times" with no workflow progress (6.5). Participants noted, "It was also easier to go back to parts I missed.", "I know what to expect to get to the final result.", and "It is great for skipping straight to relevant portions of the tutorial."

All participants frequently used the ability to click on timeline links to navigate directly to specific images and steps. They clicked the interactive timeline links 8.9 times on average ($\sigma = 6.7$) in a single task. We also analyzed the tracking log, which records an event when the user clicks on an interactive link or a pause button, or drags the playhead to another position. The learners watched videos less linearly with ToolScape: The ToolScape condition recorded 150 such events, versus only 96 in the Baseline condition. In ToolScape, 107 out of 150 events were interactive link clicks and 43 were pause button clicks or direct scrubbing on the player. These findings indicate that interactive links largely replaced the need for pause or scrubbing, and encouraged the stepwise navigation of the procedure.

Lessons for How-To Video Annotation

The study results suggest that annotated step information makes how-to videos much more effective for learners. However, the bottleneck is in obtaining the annotations. Here are some lessons from our experience annotating videos by hand:

- Extracting step information from how-to videos involves detecting timing, generating a natural language description of a step, and capturing before and after states.
- It often requires multi-pass watching, which adds to task complexity. Before knowing what each step is, the annotator cannot extract before and after thumbnail images. This experience supports a design choice to split the work into multiple stages so that in each stage, the annotator's attention is focused on a single, simple task.
- Hand annotation is time-consuming. Roughly three times the original video length was required by trained annotators to annotate each how-to video.
- Timing detection is difficult. Sometimes there is an interval between when a step is spoken and demonstrated. Also, if the goal is to find a starting time of a step, the annotator has to watch, verify, and scroll back to mark as a valid step.

These lessons informed the design of our crowdsourced how-to video annotation method, which we now present.

CROWDSOURCING WORKFLOW: FIND-VERIFY-EXPAND

Using lessons from our formative study, we designed a three-stage crowdsourcing workflow for annotating how-to videos with procedural steps, timings, textual descriptions, and before and after thumbnail images. This workflow works with any how-to video regardless of its domain, instructional style, and presentation. It also collects annotations with untrained crowd workers (e.g., workers on Mechanical Turk).

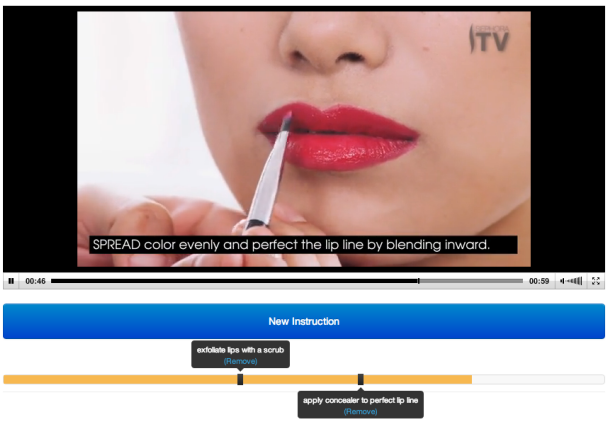

Figure 6. In the Find stage, the crowd worker adds new steps to the timeline by clicking on the "New Instruction" button.

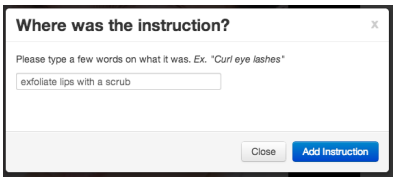

Figure 7. Upon clicking on the "New Instruction" button, a popup window asks the worker to describe what the step is about in free-form text.

Inspired by crowd design patterns that segment a bigger task into smaller micro-tasks [5], our workflow decomposes the annotation task into three stages and each video into shorter segments. This design addresses the task complexity and multi-pass overhead problems of manual annotation.

We developed a generalizable crowd workflow pattern called **Find-Verify-Expand** (Figure 1) for detecting temporal and visual state changes in videos, such as steps in a how-to video, highlights from a sports game, or suspicious incidents from a surveillance video. The unique *Expand* stage captures surrounding context and causal relationships (e.g., before/after images for a step in a how-to video) by expanding on the detected event (e.g., a step in a how-to video). To better handle crowd output coming from timing detection and image selection, we apply clustering algorithms and text and visual analysis techniques to intelligently merge results from workers.

Stage 1: FIND candidate steps

This crowd task collects timestamps and text descriptions for possible steps from a video segment. While watching the video, the worker adds a step by clicking on the "New Instruction" button every time the instructor demonstrates a step (Figure 6). Each time the worker clicks on the button, the task prompts the worker to describe the step in free-form text (Figure 7). The same segment is assigned to three workers, whose results get merged to create candidate steps.

Pre-processing: A video is segmented into one-minute chunks. We learned from pilot runs that longer video segments lead to lower annotation accuracy toward the end and slower responses on Mechanical Turk. However, a drawback in using segmented video is the possibility of missing steps

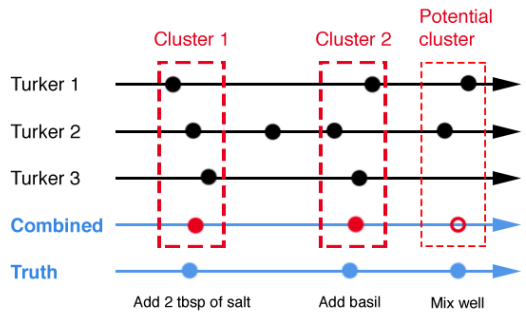

Figure 8. Our clustering algorithm groups adjacent time points into a candidate step. It further adds a potential cluster as a candidate, which might turn out to be a proper step once checked in the Verify stage. This inclusive strategy mitigates the effect of clustering errors.

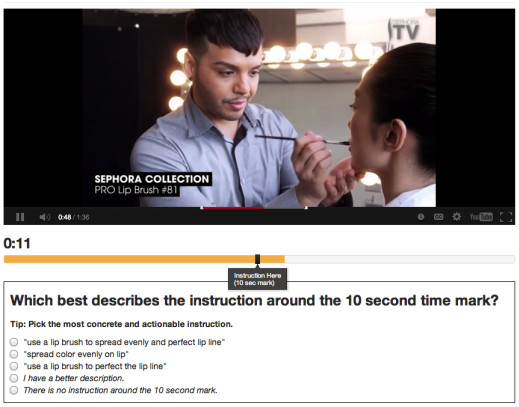

Figure 9. The Verify stage asks the worker to choose the best description of a candidate step. The options come from workers in the Find stage. Additional default options allow the worker to either suggest a better description or mark the step as invalid.

near segment borders. We address this issue by including a five-second overlap between segments, and attaching the final segment to the prior one if it is shorter than 30 seconds.

Task Design: For quality control, the task first ensures that the user has audio by giving a test that asks the worker to type in a word spoken from an audio file. Our pilot runs showed that labeling accuracy drops significantly when the worker does not listen to audio. Secondly, we disable the Submit button until the video playhead reaches the end to ensure that the worker watches the entire segment. Finally, when the worker clicks on the "New Instruction" button, the video pauses and a dialog box pops up to ask what the step was. Our initial version simply added a tick on the timeline and continued playing without pausing or asking for a label. But this resulted in workers clicking too many times (as many as 100 for a 60-second chunk) without thinking. The prompt adds self-verification to the task, which encourages the worker to process the workflow by each step. The prompt also includes an example label to show the format and level of detail they are expected to provide (Figure 7).

Post-Processing: The workflow intelligently merges results from multiple workers to generate step candidates. To cluster nearby time points given by different workers into a single step, we use the DBSCAN clustering algorithm [12] with a timestamp difference as the distance metric. The clustering idea is shown in Clusters 1 and 2 in Figure 8. The algorithm takes ϵ as a parameter, which is defined by the maximum distance between two points that can be in a cluster relative to the distance between farthest points. We train ϵ once initially on a small set of pilot worker data and ground truth labels. Our tests show that the values between 0.05 and 0.1 yield high accuracy, regardless of domain or video. We configured the algorithm to require at least two labels in every cluster, similar to majority voting among the three workers who watched the segment. We considered other clustering algorithms such as K-Means, but many require the number of clusters as an input parameter. In video annotation, the number of steps is neither known a priori nor consistent across videos.

Depending on videos and parameters, the DBSCAN algorithm might over-generate (false positive) or under-generate (false negative) clusters. We bias the algorithm to over-generate candidate steps ('potential cluster' in Figure 8) and aim for high recall over high precision, because the first stage is the only time the workflow generates new clusters. We improve the initial clusters in three ways, with the goal of higher recall than precision. First, we take into account the textual labels to complement timing information. The clustering initially relies on workers' time input, but using only time might result in incorrect clusters because steps are distributed unevenly time-wise. Sometimes there are steps every few seconds, and other times there might be no step for a minute. We run a string similarity algorithm between text labels in border points in clusters, to rearrange them to the closer cluster. Second, we break down clusters that are too large by disallowing multiple labels from one worker to be in a cluster. Finally, if there are multiple unclustered points within ϵ between clusters, we group them into a candidate cluster. For each cluster, we take a mean timestamp as the representative time to advance to the Verify stage.

Stage 2: VERIFY steps

Here the worker's verification task is to watch a 20-second clip that includes a candidate step and textual descriptions generated from the prior stage, and vote on the best description for the step (Figure 9). The workflow assigns three workers to each candidate step, whose votes are later merged.

Pre-processing: For each of the candidate steps from Stage 1, the workflow segments videos into 20-second clips around each step (10 seconds before and after).

Task Design: To prevent workers from selecting the first result without reviewing all options, we randomize the order of options presented each time. We also lowercase all labels to prevent capitalized descriptions from affecting the decision. Also, the Submit button becomes clickable only after the worker finishes watching the 20-second clip.

In addition to candidate text descriptions, two additional options are presented to workers: "I have a better description",

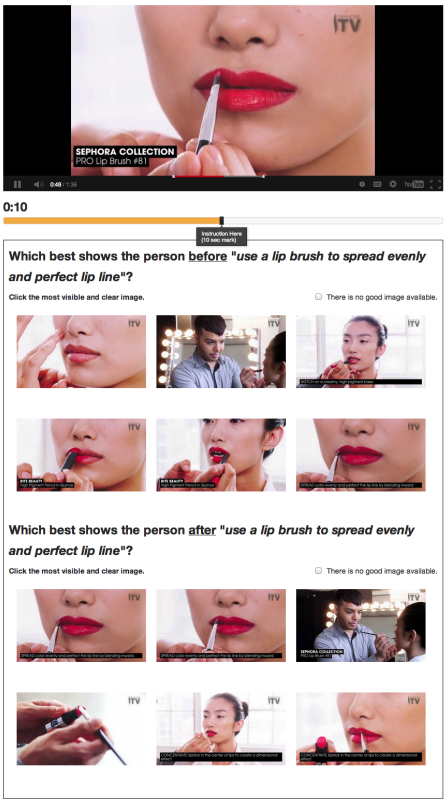

Figure 10. The Expand stage asks the worker to choose the best before and after images for a step. The worker visually reviews the thumbnail options and clicks on images to decide.

which improves the step label, and "There is no instruction", which filters out false positives from Stage 1.

Post-Processing: Two possible outputs of this stage are 1) finalizing the timestamp and description for a valid step, or 2) removing a false step. The workflow uses majority voting to make the final decision: If two or more workers agreed on a description, it becomes the final choice. If workers are split between three different options, it checks if some of the selected text descriptions are similar enough to be combined. We first remove stop words for more accurate comparisons, and then apply the Jaro-Winkler string matching algorithm [17]. If the similarity score is above a threshold we configured with initial data, we combine the two descriptions with a longer one. If not, it simply picks the longest one from the three. The decision to pick longer description for tie-breaking comes from a pilot observation that longer descriptions tend to be more concrete and actionable (e.g., "grate three cups of cheese" over "grate cheese").

Stage 3: EXPAND with before and after images for steps

This final stage collects the before and after images of a step, which visually summarize its effect. This stage captures surrounding context and causal relationships by expanding on what is already identified in Find and Verify. The worker's task here is to watch a 20-second video clip of a step, and select a thumbnail that best shows the work in progress (e.g., food, face, or Photoshop image) before and after the step (Figure 10). The workflow assigns three workers to each step.

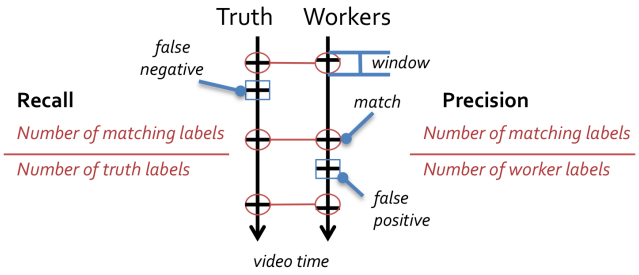

Figure 11. Our evaluation uses the Hungarian method to match extracted and ground truth steps with closest possible timestamp within a 10-second window size. Then we compute precision and recall, which indicate if our workflow over- or under-extracted steps from ground truth.

Pre-processing: This stage uses a 20-second video clip of a step verified in Stage 2, and uses its final text label to describe the step. It creates thumbnails at two-second intervals to present as options, 10 seconds before and after the step.

Task Design: Our initial design asked workers to click when they see good before and after images, but this resulted in low accuracy due to variable response time and the lack of visual verification. We then simplified the task to a multiple choice question. Selecting from static thumbnail images makes the task easier than picking a video frame.

Post-Processing: Similar to the Verify stage, we apply majority voting to determine the final before and after images. For merging and tie breaking, we use Manhattan distance, an image similarity metric that computes pixel differences between two images.

EVALUATION

We deployed our annotation workflow on Mechanical Turk and evaluated on:

- **Generalizability**: Does the workflow successfully generate labels for different types of how-to tasks in different domains with diverse video production styles?
- **Accuracy**: Do collected annotations include all steps in the original video, and avoid capturing too many (false positive) or too few (false negative)? How do textual descriptions generated by crowd workers compare to those generated by trained annotators?

Methodology

We used our workflow and Mechanical Turk to fully extract step information from the 75 how-to videos in our annotation corpus, with 25 videos each in cooking, makeup, and graphics editing software (Photoshop). We did not filter out videos based on use of subtitles, transitions, or audio, to see if our annotation workflow is agnostic to presentation styles. Out of 75 videos in our set, 7 did not have audio, and 27 contained text overlays. For each domain, we picked five tasks to cover diverse types of tasks: Cooking – pizza margherita, mac and cheese, guacamole, samosa, and bulgogi; Makeup – bronze look, reducing redness, smokey eyes, bright lips, and summer

glow; Photoshop: motion blur, background removal, photo to sketch, retro effect, and lomo effect. The mean video length was 272 seconds, summing to over 5 hours of videos.

Results

Our evaluation focuses on comparing the quality of step information produced by our crowdsourcing workflow against ground truth annotations from our corpus.

The Turk crowd and trained annotators (two co-authors with educational video research experience) generated similar numbers of steps (Table 1). In Stage 1, 361 one-minute video segments were assigned to Turkers, who generated 3.7 candidate steps per segment, or 53.6 per video. Clustering reduced that to 16.7 steps per video. Stage 2 further removed overgenerated steps, resulting in 15.7 per video, which is nearly equivalent to the ground truth of 15 steps per video.

Precision indicates how accurate extracted steps are compared to ground truth, while *recall* shows how comprehensively the workflow extracted ground truth steps (Figure 11). We present precision and recall results considering only the timing of steps (Stage 1), and both the timing and the textual description accuracy (Stage 2). For matching crowd-extracted steps to ground truth steps, we use the Hungarian method [19] whose cost matrix is filled with a time distance between steps.

Evaluating Stage 1: FIND

We consider only precision and recall of times in the Stage 1 evaluation because final textual descriptions are not yet determined. Detecting the exact timing of a step is not straightforward, because most steps take place over a time period, verbal and physical steps are commonly given with a time gap.

To more accurately account for the timing issue, we set a highest threshold in time difference that accepts a Turker-marked point as correct. We set the threshold to 10 seconds, which indicates that a step annotation more than 10 seconds off is discarded. This threshold was based on heuristics from step intervals in our corpus: We hand-annotated, on average, one step every 17.3 seconds in our video corpus (mean video length / number of steps in ground truth = 272/15.7), so a maximum 10-second difference seems reasonable.

The mean distance between ground truth steps and extracted steps (ones within 10 seconds of the ground truth) was only 2.7 seconds. This suggests that for matched steps, the time-based clustering successfully detected the timing information around this distance. When considering only time accuracy, our workflow shows 0.76 precision and 0.84 recall (Table 2).

Evaluating Stage 2: VERIFY

Here we combine the accuracy of both timing and text descriptions. Precision for this stage captures what fraction of steps identified by the workflow are both placed correctly on the time line and whose description reasonably matches the ground truth. The analysis shows 0.77 precision and 0.81 recall over all the videos (Table 2).

For text accuracy measurement, we use the string similarity algorithm to see if a suggested description is similar enough to a description from ground truth. We apply the same threshold as what we configured in the workflow for tie breaking in the Verify stage. The precision and recall both go down when the text similarity condition is added, but precision recovers from the post-processing of steps in this stage. Two enhancements contribute to this recovery: removing steps that workers indicated as "no instruction" from the task, and merging nearby steps that have identical descriptions.

By Turkers	After Stage1	After Stage2	Ground Truth
53.6	16.7	**15.7**	**15.0**

Table 1. The mean number of steps generated by the workflow in each stage. At the end the workflow extracted 15.7 steps per video, which is roughly equivalent to 15.0 from ground truth. Stage 1 clusters the original Turker time points, and Stage 2 merges or removes some.

Stage 1. Time only		Stage 2. Time + Text	
Precision	Recall	Precision	Recall
0.76	0.84	0.77	0.81

Table 2. When considering time information only, recall tends to be higher. When considering both time and text descriptions, incorrect text labels lower both precision and recall, but removing unnecessary steps in Stage 2 recovers precision.

In 76% of the steps, two or more Turkers agreed on a single description. For the rest, the tie breaking process determined the final description. For 13% of the steps, Turkers provided their own description.

Evaluating Stage 3: EXPAND

This evaluation should judge if crowd-selected before and after images correctly capture the effect of a step. Because this judgment is subjective, and there can be multiple correct before and after images for a step, we recruited six external human evaluators to visually verify the images. We assigned two evaluators to each domain based on their expertise and familiarity with the domain, and gave a one-hour training session on how to verify before and after images. For each workflow-generated step, we presented an extracted text description along with a before and after image pair. Their task was to make binary decisions (yes / no) on whether each image correctly represents the before or after state of the step.

We used Cohen's Kappa to measure inter-rater agreement. The values were 0.57, 0.46, 0.38, in cooking, makeup, and Photoshop, respectively, which show a moderate level of agreement [1]. Results show that on average, both raters marked 60% of before and after images as correct. At least one rater marked 81.3% as correct.

Cost, time, and tasks

We created a total of 8,355 HITs on Mechanical Turk for annotating 75 videos. With three workers on each task and a reward of $0.07, $0.03, and $0.05 for Find, Verify, and Expand, respectively, the average cost of a single video was $4.85, or $1.07 for each minute of a how-to video. The Expand stage was more costly ($2.35) than the first two; thus, time points and text descriptions can be acquired at $2.50 per video. The average task submission time was 183, 80, and 113 seconds for Find, Verify, and Expand, respectively.

Results summary
In summary, our workflow successfully extracted step information from 75 existing videos on the web, generalizing to three distinct domains. The extracted steps on average showed 77% precision and 81% recall against ground truth, and were 2.7 seconds away from ground truth. Human evaluators found 60% of before and after images to be accurate.

DISCUSSION AND LIMITATIONS

We now discuss qualitative findings from the experiment, which might have practical implications for future researchers designing crowd workflows.

Detecting precise timing of a step. We observed that Turkers add new steps with higher latency than trained annotators, resulting in Turker-labeled time points being slightly later than those by annotators for the same step. The trained annotators often rewinded a few seconds to mark the exact timing of a step after seeing the step, whereas most Turkers completed their tasks in a single pass. While this might be a limitation of the workflow, our results show that a reasonable window size mitigates such differences. We will explore time-shifting techniques to see if timing accuracy improves.

Handling domain and video differences. Extraction accuracy in our workflow was consistent across the three domains with different task properties. This finding validates our domain-agnostic approach based on the general properties of procedural tasks. Photoshop videos were often screencasts, whereas cooking and makeup videos were physical demonstrations. Cooking videos contained higher number and density of steps than makeup or Photoshop videos, while Photoshop and makeup videos often had longer steps that required fine-grained adjustments and tweaking. Also, some videos were studio-produced with multiple cameras and high-quality post-processing, while others were made at home with a webcam. Our workflow performed robustly despite the various differences in task properties and video presentation styles.

Extracting steps at different conceptual levels. Video instructors present steps at different conceptual levels, and this makes it difficult to keep consistent the level of detail in Turkers' step detection. In a makeup video, an instructor said "Now apply the bronzer to your face evenly", and shortly after applied the bronzer to her forehead, cheekbones, and jawline. While trained annotators captured this process as one step, our workflow produced four, including both the high-level instruction and the three detailed steps. Turkers generally captured steps at any level, but our current approach only constructs a linear list of steps, which sometimes led to redundancy. Previous research suggests that many procedural tasks contain a hierarchical solution structure [7], and we plan to extend this work to hierarchical annotation.

POSSIBLE APPLICATIONS AND GENERALIZATION

We list possible applications that leverage step information extracted from our workflow, and discuss ways to generalize the Find-Verify-Expand pattern beyond how-to videos.

Our scalable annotation workflow can enable a series of novel applications in addition to the ToolScape player. First, better video search can be made possible with finer-grained video indices and labels. For example, ingredient search for cooking or tool name search for Photoshop can show all videos and time points that cover a specific tutorial element. Furthermore, video players can present alternative examples to a current step. If a learner is watching how to apply the eyeliner, the interface can show just the snippets from other videos that include demonstrations of the eyeliner. This allows the learner to hop between different use cases and context for the step of interest, which can potentially improve learning outcomes.

We believe the Find-Verify-Expand pattern can generalize to annotating broader types of metadata beyond steps from how-to videos. For example, from a soccer video this pattern can extract goal moments with Find and Verify, and then use Expand to include a crucial pass that led to the goal, or a ceremony afterward. Generally, the pattern can extract metadata that is human-detectable but hard to completely automate. It is a scalable method for extracting time-sensitive metadata and annotating streaming data, which can be applied to video, audio, and time-series data.

CONCLUSION AND FUTURE WORK

This paper presents a scalable crowdsourcing workflow for annotating how-to videos. The Find-Verify-Expand pattern efficiently decomposes the complex annotation activity into micro-tasks. Step information extracted from the workflow can enable new ways to watch and learn from how-to videos. We also present ToolScape, an annotation-enabled video player supporting step-by-step interactivity, which is a potential client of this workflow. Our lab study shows the value of accessing and interacting with step-by-step information for how-to videos. Participants watching videos with ToolScape gained higher self-efficacy, rated their own work higher, and produced higher-rated designs.

Our future work will explore applying the workflow to additional procedural task domains such as origami, home DIY tasks, and Rubik's cube. We will also explore procedural tasks that require a conceptual understanding of the underlying concept, such as solving algorithm or physics problems.

Another direction for research is collecting task information using learners as crowd. We believe learners can potentially provide more advanced, higher-level, and richer information not possible with Turkers, if their learning interactions can naturally provide useful input to the system. Combining crowdsourcing with "learnersourcing" can extract rich annotations from existing resources while enhancing learning.

ACKNOWLEDGMENTS

The authors would like to thank the members of the UID Group at MIT CSAIL for their feedback. This work was funded in part by Quanta Computer. Juho Kim is supported by the Samsung Fellowship.

REFERENCES

1. Altman, D. G. *Practical statistics for medical research*, vol. 12. CRC Press, 1991.

2. Bandura, A. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review 84*, 2 (1977), 191.
3. Banovic, N., Grossman, T., Matejka, J., and Fitzmaurice, G. Waken: Reverse engineering usage information and interface structure from software videos. In *UIST '12* (2012).
4. Bernstein, M. S., Brandt, J., Miller, R. C., and Karger, D. R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *UIST '11*, ACM (2011).
5. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *UIST '10* (2010), 313–322.
6. Bernstein, M. S., Teevan, J., Dumais, S., Liebling, D., and Horvitz, E. Direct answers for search queries in the long tail. In *CHI '12'* (2012), 237–246.
7. Catrambone, R. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General 127*, 4 (1998), 355.
8. Chi, P.-Y., Ahn, S., Ren, A., Dontcheva, M., Li, W., and Hartmann, B. Mixt: Automatic generation of step-by-step mixed media tutorial. In *UIST '12* (2012).
9. Chi, P.-Y. P., Liu, J., Linder, J., Dontcheva, M., Li, W., and Hartmann, B. Democut: generating concise instructional videos for physical demonstrations. In *UIST '13*, ACM (2013).
10. Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., and Klemmer, S. R. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput.-Hum. Interact. 17*, 4 (Dec. 2010), 18:1–18:24.
11. Eiriksdottir, E., and Catrambone, R. Procedural instructions, principles, and examples: how to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Hum Factors 53*, 6 (2011), 749–70.
12. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, vol. 96 (1996).
13. Ferguson, E. L., and Hegarty, M. Learning with real machines or diagrams: application of knowledge to real-world problems. *Cognition and Instruction 13*, 1 (1995), 129–160.
14. Grabler, F., Agrawala, M., Li, W., Dontcheva, M., and Igarashi, T. Generating photo manipulation tutorials by demonstration. In *SIGGRAPH '09* (2009), 1–9.
15. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories. In *UIST '10* (2010).
16. Hadidi, R., and Sung, C.-H. Students' acceptance of web-based course offerings: an empirical assessment. *AMCIS 1998* (1998).
17. Jaro, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association 84*, 406 (1989), 414–420.
18. Kim, J. Toolscape: enhancing the learning experience of how-to videos. In *CHI EA '13* (2013), 2707–2712.
19. Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.
20. Lafreniere, B., Grossman, T., and Fitzmaurice, G. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *CHI '13* (2013), 1779–1788.
21. Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. Real-time crowd labeling for deployable activity recognition. In *CSCW '13* (2013), 1203–1212.
22. Li, F. C., Gupta, A., Sanocki, E., He, L.-w., and Rui, Y. Browsing digital video. In *CHI '00*, ACM (2000).
23. Nguyen, P., Kim, J., and Miller, R. C. Generating annotations for how-to videos using crowdsourcing. In *CHI EA '13*, ACM (2013), 835–840.
24. Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. Platemate: crowdsourcing nutritional analysis from food photographs. In *UIST '11* (2011), 1–12.
25. Park, S., Mohammadi, G., Artstein, R., and Morency, L.-P. Crowdsourcing micro-level multimedia annotations: The challenges of evaluation and interface. In *CrowdMM Workshop* (2012).
26. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. Pause-and-play: Automatically linking screencast video tutorials with applications. In *UIST 2011* (2011).
27. Schunk, D. Goal setting and self-efficacy during self-regulated learning. *Educational psychologist 25*, 1 (1990), 71–86.
28. Tversky, B., Morrison, J. B., and Betrancourt, M. Animation: can it facilitate? *International journal of human-computer studies 57*, 4 (2002), 247–262.
29. van der Meij, H., Blijleven, P., and Jansen, L. What makes up a procedure? *Content & Complexity* (2003).
30. Vondrick, C., Ramanan, D., and Patterson, D. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV 2010*. Springer, 2010, 610–623.
31. Yuen, J., Russell, B., Liu, C., and Torralba, A. Labelme video: Building a video database with human annotations. In *ICCV 2009*, IEEE (2009), 1451–1458.
32. Zhang, D., Zhou, L., Briggs, R. O., and Jr., J. F. N. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management 43*, 1 (2006), 15 – 27.
33. Zimmerman, B. J., Bandura, A., and Martinez-Pons, M. Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal 29*, 3 (1992), 663–676.