

---

# Generating Annotations for How-to Videos Using Crowdsourcing

**Phu Nguyen**

MIT CSAIL  
32 Vassar St.  
Cambridge, MA 02139  
phun@mit.edu

**Juho Kim**

MIT CSAIL  
32 Vassar St.  
Cambridge, MA 02139  
juhokim@mit.edu

**Robert C. Miller**

MIT CSAIL  
32 Vassar St.  
Cambridge, MA 02139  
rcm@mit.edu

**Abstract**

How-to videos can be valuable for learning, but searching for and following along with them can be difficult. Having labeled events such as the tools used in how-to videos could improve video indexing, searching, and browsing. We introduce a crowdsourcing annotation tool for Photoshop how-to videos with a three-stage method that consists of: (1) gathering timestamps of important events, (2) labeling each event, and (3) capturing how each event affects the task of the tutorial. Our ultimate goal is to generalize our method to be applied to other domains of how-to videos. We evaluate our annotation tool with Amazon Mechanical Turk workers to investigate the accuracy, costs, and feasibility of our three-stage method for annotating large numbers of video tutorials. Improvements can be made for stages 1 and 3, but stage 2 produces accurate labels over 90% of the time using majority voting. We have observed that changes in the instructions and interfaces of each task can improve the accuracy of the results significantly.

**Author Keywords**

Video tutorials; how-to videos; crowd workers.

**ACM Classification Keywords**

H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces [Graphical user interfaces (GUI)]

## **Introduction**

How-to videos are great resources for users to learn in a variety of domains from folding intricate origami to using professional-grade graphic editing programs. Many use the Internet to search for the video that fits their needs, but finding the right video tutorial catered to a person's task or learning goal is not always easy. Searching on Youtube for a task such as "removing an object in Photoshop" returns over 4,000 video results. Users often have trouble determining which videos are relevant without watching them. Many might inefficiently spend minutes to hours skimming multiple videos before finding the one they need.

Current search engines used for finding how-to videos rely on basic metadata such as view counts, titles, descriptions, and tags. By gathering more data about each video relevant to the domain such as tools and plugins being used at certain timestamps of a graphic design throughout the video, a better searching interface that is catered towards finding how-to videos can be created. Improved browsing interfaces for watching how-to videos would also benefit from more annotations. ToolScape has demonstrated that users who use a video-browsing interface with a storyboard summation and an interactive timeline are able to produce higher quality graphics based on external ratings [2].

In order for interfaces like ToolScape to be useful, annotations for large numbers of videos should be available. Our research introduces a crowdsourcing method that is capable of producing accurate annotations while maintaining efficiency and scalability. Our method uses three stages to gather metadata: (1)

gathering timestamps of important events, (2) labeling each event, and (3) capturing how each event affects the task of the tutorial. We decided against using trained experts as our workers because obtaining enough trained experts to annotate thousands of how-to videos would not be time or cost effective. The task of annotating videos can be broken down into easier subtasks that crowd workers can complete.

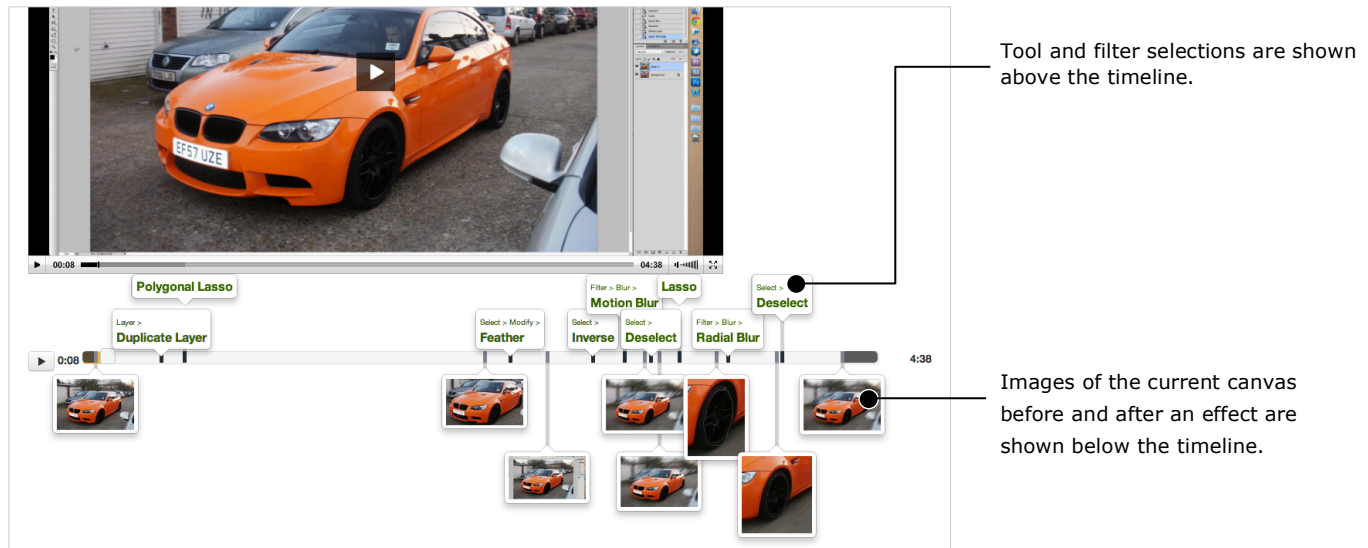
## **Related Work**

Recent studies have shown that gathering data using crowdsourcing can be accurate and highly successful. The ESP game has shown the potential of using interactive games to produce labels for images [5]. Sorokin used Amazon Mechanical Turk to generate quality data annotations at a cheap rate [4]. Soylent has shown that splitting tasks into a multi-stage process using the Find-Fix-Verify method improves the quality and accuracy of the results provided by crowd workers [1].

LabelMe has shown that by providing users web-based annotation tools, they can create and share annotations such as labels of objects seen in images [3]. The study used annotations created by LabelMe to train object recognition and detection.

## **Proposed Method**

Our method follows the Find-Fix-Verify pattern introduced by Bernstein et al. modified for generating annotations for how-to videos [1]. By breaking down generating annotations into shorter multiple tasks, the accuracy of annotations may increase. Our tasks are simple enough that workers are not required to have Photoshop experience to participate.



**Figure 1:** ToolScape uses an interactive timeline to display annotations about the tutorial. Users can click on any tool annotation to go to the part of the video where the tool is being selected and used.

In this method, each worker completes one of three tasks:

1. Get timestamps of important events
2. Label each important event
3. Capture how each event affects the end result

We chose Photoshop how-to videos as our example domain due to its popularity and abundance. In this domain, we defined important events as locations where the instructor selects and uses a new tool. We capture before and after images in task 3 to show how each tool affects the task in the tutorial.

## Workflow Design

### Task 1

The worker watches a video clip of a Photoshop how-to video and click on the "Tool Clicked" button every time the instructor selects a tool in one of the red regions (see Task 1 in Figure 1). A mandatory tutorial is shown before starting the task that helps the worker understand when and when not to click the button. The tutorial was added after we concluded that users were having trouble understanding when to click. To increase the HIT acceptance rate, each worker is paid \$0.05 per completion since this task required more time from worker than the other two tasks. We collect the timestamps of the video every time the button is clicked.

## Task 1

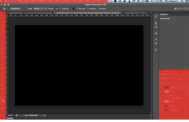
**Tutorial**

In order to start this task, you must complete short a tutorial.

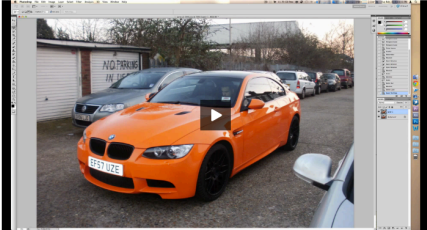
1. Copy and paste this url into a new tab or window:  
<http://www.phuaster.com/hwk2/test.html?assignmentid=1>
2. Complete the test and paste the passcode below.  
64

**Instructions**

1. **Please have your audio on!**
2. Watch the Photoshop video tutorial.
3. Click on the  button every time the instructor in the video clicks on a tool in the red regions.
  - A tick  will be added to the timeline which represents the time the tool was selected.
  - To remove a tick, double-click on the tick.



**Task**



**Finish**

When you are done hit the submit button.

## Task 2

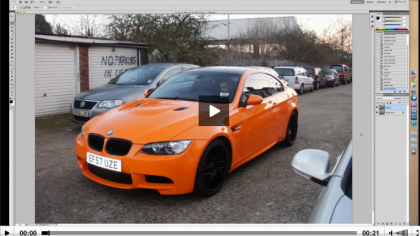
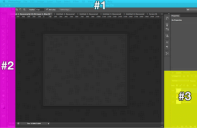
**Label Tools Used in Photoshop Videos**

**Background**

In this video, the instructor is going to use a tool. We would like to know what tool it is.

**Instructions**

1. **Please have your audio on!**
2. Watch the video clip.
3. Find the tool the instructor selects that is within one of the highlighted regions (#1, #2, #3).
4. Find the tool the instructor selects from the drop down menu.



**What was the name of the tool?**

Please Select The Tool Used

No tool was used in the video

**Finish**

When you are done hit the submit button.

## Task 3

**Find Before and After Images in Photoshop**

**Information**

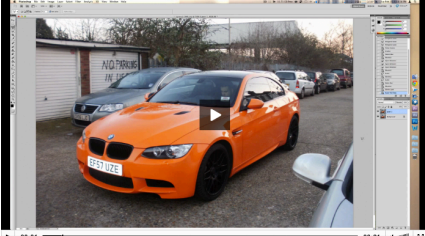
In this video, the instructor is going to use the **Polygonal Lasso** tool in Photoshop. We would like the before and after images for **Polygonal Lasso**.

**Instructions**

1. **Please have your audio on!**
2. Watch the Photoshop video tutorial.
3. **Before Image:** Click the  when you see the image during the first ten seconds.  
*Note: The image should not be blocked by any dialog or menu.*
4. Wait until the instructor is done with using **Polygonal Lasso**.
5. **After Image:** Click the  when you see the image after the instructor is finished with **Polygonal Lasso** during the last ten seconds.

• Tip: Clicking on the buttons multiple times to override the current selection.

**Task**



**Finish**

When you are done hit the submit button.

**Figure 2:** Interfaces of the three tasks from our latest implementation tested during our experiment.

### Task 2

The worker labels the tool used in the video clip by using a dropdown menu to select the tool label (see Task 2 in Figure 1). A timeline visualizer beneath the video player was added during later iterations of the design to help the worker understand when the tool is being used. We collect the tool label from the task.

### Task 3

The worker watches a video clip and click on the "Before Image" button when they can see graphic before the tool is used and the "After Image" button when they see the graphic after the tool is used (see Task 3 in Figure 1). We also conducted a few live user studies and results suggested that users might have trouble with the task because they do not read the

instructions. Therefore, the video player and buttons are hidden from the worker until the worker has successfully read the instructions. We also added a timeline visualizer identical to that in task 2. We collect the timestamps of the video when these buttons are pressed.

### Experiment

We tested each task with 90 crowd workers on Amazon Mechanical Turk. Each worker that completed task 1 was paid \$0.05 and \$0.02 for tasks 2 and 3. No qualifications from workers were required to accept the task.

Three Photoshop how-to videos were used for our experiment. We believe that using shorter video clips for each task will result in higher accuracy, so each video was spliced into one-minute chunks. In order to test the usability of our interfaces for tasks 2 and 3, we chose not to generate video clips using results from task 1. Instead, we generated twenty-second clips of the three videos by manually finding where tools were used in the tutorials and creating clips such that only a single tool was used in the video and the tool was used at middle of the clip. This is equivalent to using ground truth timestamps from task 1 to generate video clips for tasks 2 and 3.

### Experiment Results

#### Task 1: Click when tools are used in Photoshop

Workers completed 90 of the 90 HITs for task 1. On average, 1.37 timestamps were correctly submitted per video, 1.13 timestamps were missed per video, and 1.47 timestamps were added per video that were unnecessary. This results in a 44% accuracy rate calculated using the equation:

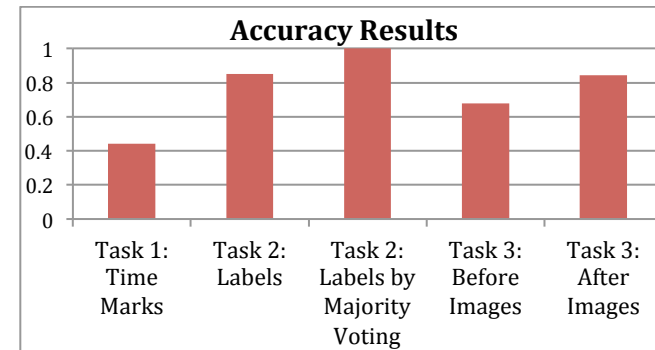
$$Accuracy = \frac{correct\ indices}{correct\ indices + missed\ indices + false\ indices}$$

#### Task 2: Label tools used in Photoshop how-to videos

Workers completed 90 of the 90 HITs for task 2. Labels of the tools produced by workers were correct 85% of the time. However, if we use majority voting out of 5 workers who label the same tool, 100% of the videos are correctly labeled.

#### Task 3: Capture before and after images

Workers completed 90 of the 90 HITs for task 3. Workers captured an acceptable before image 67% of the time and an after image 84% of the time. Overall, the mean accuracy was 76%.



**Figure 3:** These bar graphs display the accuracy of our three-stage method in our latest experiment. Task 1 shows the lowest accuracy. The accuracy of task 2 shows significant improvement with majority voting. For task 3, collecting after images was slightly more accurate than collecting before images.

### Discussion

Both of our experiments have shown that crowd workers are having the most trouble with gathering

timestamps in task 1. Workers are successful with labeling tools used and fairly successful with capturing before and after images.

The current accuracies for tasks 2 and 3 are based on video clips generated using ground truth results from task 1. However, we would like to generate video clips based on actual results from task 1 because our goal is to pipeline the three tasks together in an automated process. If results from task 1 are inaccurate, the accuracies of tasks 2 and 3 are also affected.

In our current design, task 1 is still the most difficult task for workers to complete. Considerations have been made to incorporate more of the Find-Fix-Verify method into this task to generate better results. However, that would require more workers to complete task 1 per video.

Our results have been based on work completed by nine workers. Majority voting is used for some of our tasks, so we would like to decrease the number of workers required to complete each task in the three-stage process to a number closer to two or three in order to reduce the cost to annotate a video.

### **Future Work**

We would also like to evaluate our three-stage process of generating video annotations for video tutorials using other video domains such as origami folding and

cooking. We would like our process to be successful with a wide variety of domains with little modifications to the core concepts of the method.

Our long-term goal is domain-independent metadata generation by both crowdsourcing and computer vision. Our crowdsourcing method can be used to collect data for training and evaluating computer vision approaches.

### **Acknowledgments**

The MIT EECS SuperUROP Program and Quanta Computer, Inc. have supported this work.

### **References**

- [1] Bernstein, M., Little, G., Miller, R., et al. Soylent: A Word Processor with a Crowd Inside. UIST '10, ACM Press (2010).
- [2] Kim, J. ToolScape: Enhancing the Learning Experience of How-to Videos. In Proc. of CHI 2013 Extended Abstracts, to appear.
- [3] Murphy, K. and Freeman W. LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vision 77, 1-3 (May 2008), 157-173.
- [4] Sorokin, A. and Forsyth, D. Utility data annotation with Amazon Mechanical Turk. CVPR '08, (2008).
- [5] von Ahn, L. and Dabbish, L. Labeling images with a computer game. CHI '04, ACM Press (2004).