Generating annotations for how-to videos using crowdsourcing

Citation: Nguyen, Phu. "Generating annotations for how-to videos using crowdsourcing." In Proceeding CHI EA '13 CHI '13 Extended Abstracts on Human Factors in Computing Systems, Paris, France, April 27- May 02, 2013, Pages 835-840.

As Published: 10.1145/2468356.2468506

Publisher: Association for Computing Machinery (ACM)

Persistent URL: https://hdl.handle.net/1721.1/121444

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Generating Annotations for How-to Videos Using Crowdsourcing

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

CHI'13, April 27 – May 2, 2013, Paris, France.
Copyright 2012 ACM 978-1-XXXX-XXXX-X/XX/XX...$10.00.

Abstract

How-to videos can be valuable teaching tools for users, but searching for them can be difficult. Having labeled events such as uses of tools in how-to videos would improve searching, browsing and indexing for videos. We introduce a method that uses crowdsourcing to generate video annotations for how-to videos with a three-stage process 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. We evaluate our method using Photoshop video tutorials by Amazon Mechanical Workers to investigate the accuracy, costs, and feasibility of our method for annotating large numbers of video tutorials.

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 various domains from folding intricate origami to using professional-grade graphic editing programs. Users often turn to the internet and search for the video that fits their needs, but finding the right video tutorial catered to a person’s task or learning goal isn’t always easy. Searching on Youtube for a task like “removing an object in Photoshop” returns over 4,000 video results. Often, a user cannot confidently tell if a video is useful without watching it. One might inefficiently spend minutes to hours skimming multiple videos before finding the right one.

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 frames of the Photoshop how-to video and different stages 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 interactive timeline are able to produce graphics in Photoshop that they think are higher in quality [2].

We considered both computer vision and crowdsourcing as methods to generate annotations. In order to successfully use computer vision in our method, it must be able to evaluate a broad range of how-to videos. Problems arise with computer vision because it is domain specific. It may be easy to detect tool usages in Photoshop video tutorials but it could be substantially harder to detect utensils usages in cooking how-to videos. Since the goal of the project is to make the entire process of gathering metadata automated, computer vision seemed like a plausible option.

However, we have decided to focus our work on crowdsourcing in order to collect data to train and evaluate computer vision. We propose a three-stage method to gather additional metadata that consists of gathering timestamps of important events, labeling each event, and capturing how each event affects the task of the tutorial.

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. By providing crowd workers with tools similar to LabelMe, generating annotations for how-to videos is possible.

Proposed Method

Our method follows the Find-Fix-Verify pattern introduced by Bernstein modified for generating annotations for how-to videos [1]. By breaking down generating annotations into multiple tasks, the accuracy of annotations may increase. Having shorter tasks would also make it more likely for workers to complete our tasks.

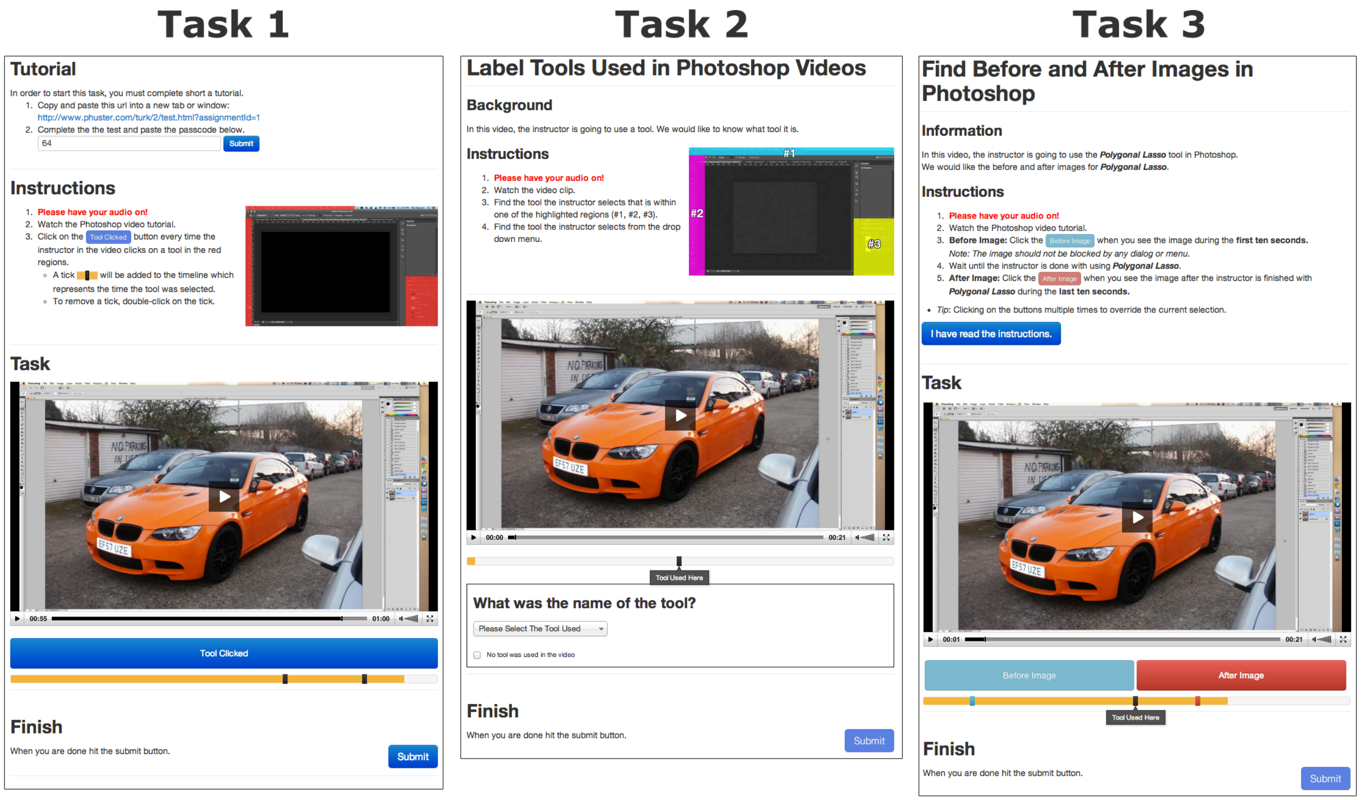
In this method, each worker will do one of three tasks:

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 its abundance, but we expect our method to be effective with any generic how-to video. In this domain, we defined important events as locations where the instructor selects and uses a new tool. We will capture before and after images in task 3 to show how each tool affects the task in the tutorial.

Workflow Design

An experiment was conducted on our first iteration of


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.phuster.com/turk/2/test.html?assignmentId=1
2. Complete the 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 Tool Clicked 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 Before Image 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 After Image 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 1: Interfaces of the three tasks from our latest implementation tested during our experiment.

our design, and revisions were made for our latest workflow design.

Task 1
Workers watch 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). They complete a mandatory tutorial before starting the task that helps them 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 workers than the other two. We collect the timestamps of the video every time the button is clicked.

Task 2
Workers label 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 region number and the tool label from the task.

Task 3
Worker watch a video clip and click on the “Before Image” button when they see the 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

In our latest 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.

We used three Photoshop how-to videos for our experiment. Each video was spliced into one-minute clips to be tested for task 1. 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 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 using all nine workers for each video clip, 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.

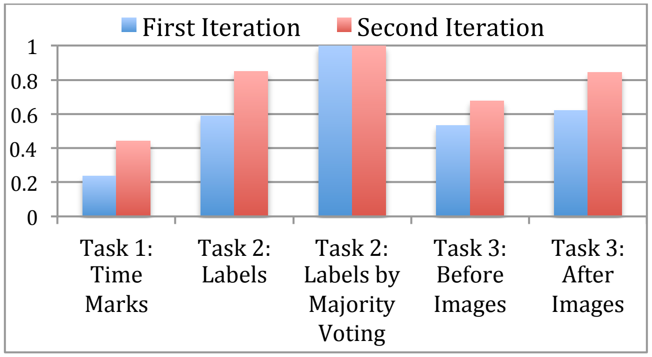


Figure 2: Bar graphs comparing accuracy results between our first experiment and our latest experiment.

Conclusions and Future Work

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 Bernstein’s 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.

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.

Acknowledgments

The MIT EECS SuperUROP Program and Quanta Computer, Inc. have supported this work. I would also like to thank Robert C. Miller and Juho Kim for their mentorship.

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., Nguyen, P., Gajos, K., and Miller, R. Summarization and Interaction Techniques for Enhancing the Browsing and Watching Experience of How-to Videos. 2012. in submission.

[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).