Summarization and Interaction Techniques for Enhancing the Learning Experience of How-to Videos

JUHO KIM, MIT CSAIL
PHU NGUYEN, MIT CSAIL
ROBERT C. MILLER, MIT CSAIL
KRZYSZTOF Z. GAJOS, Harvard SEAS

Video tutorials on the web have gained popularity, but most video repositories do not take into account the unique structure of how-to videos. Learners face difficulties in finding and watching relevant videos due to limited support for step-by-step navigation. We introduce ToolScape, a system for browsing and watching how-to videos powered by step-by-step annotations. It features the Storyboard summarization, which allows learners to quickly scan and review multiple videos, and the interactive timeline, which enables jumping to or repeating a particular step within a workflow. In a comparative study of three summarization techniques, learners found relevant videos in least time with Storyboard. In another study where participants engaged in design tasks with ToolScape and a control interface, the participants using ToolScape gained more self-efficacy, rated their own designs higher, and produced better designs. The study demonstrates the effectiveness of interactivity and learner control enabled by step-by-step navigation of how-to videos.

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

General Terms: Experimentation; Design; Human Factors.

Additional Key Words and Phrases: Video tutorials; how-to videos; learning interface; graphic design.

ACM Reference Format:
Juho Kim, Phu Nguyen, Robert C. Miller, and Krzysztof Z. Gajos, 2013. Summarization and Interaction Techniques for Enhancing the Learning Experience of How-to Videos. *ACM Trans. Comput.-Hum. Interact.* 0, 0, Article 0 (2013), 29 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

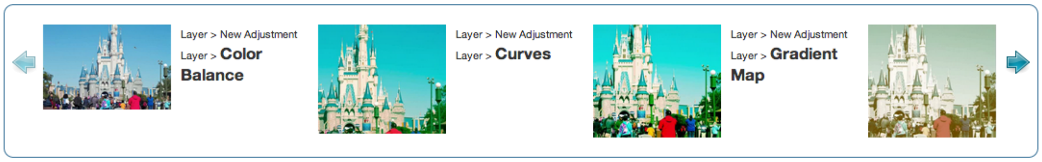
1. INTRODUCTION

Video tutorials on the web have expanded the amount and diversity of learning options available, affecting the way creative workers learn new skills. Searching on YouTube for a typical task in Photoshop, graphics editing software, such as "photoshop remove background" returns almost 200,000 video results. These videos span a variety of domains including software applications, cooking, makeup, craft, and art. A how-to video contains a specific instance of a procedural task, which is often segmented into steps that can be sequentially performed to carry out the task [van der Meij et al. 2003]. Previous research shows that the completeness and detail of stepwise instructions is integral to task performance [Eiriksdottir and Catrambone 2011]. Existing video interfaces, however, are not designed to reflect the procedural,

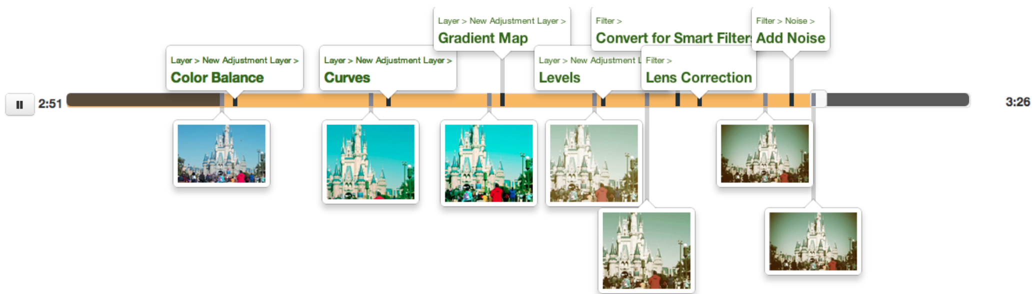
---

This work is supported in part by Adobe, and by Quanta Computer as part of the T-Party project.
Author's addresses: J. Kim, P. Nguyen and R. C. Miller, 32-G707 32 Vassar Street, Cambridge, MA 02139 USA; K. Z. Gajos, 33 Oxford Street, Cambridge, MA 02138 USA

DOI:http://dx.doi.org/10.1145/0000000.0000000



(a) ToolScape's Storyboard summarization



(b) ToolScape's interactive timeline

Fig. 1. ToolScape features the Storyboard summarization (a), which lists intermediate results and actions in a workflow as a succinct visual summary, and the interactive timeline, which adds direct access to specific moments when intermediate results and actions occur in the video.

step-by-step structure of how-to videos, and learners face difficulties when they *find* and *watch* relevant videos.

When learners attempt to find relevant videos catered to their learning goal, video search interfaces often fail to meet learners' information needs. Before deciding to watch a video, learners assess specific attributes of video search results, such as relevance, quality, context, and skill level. Search interfaces provide *metadata surrogates* [Balatsoukas et al. 2009] to help learners with the assessments, but these metadata surrogates are typically limited to title, view counts, and thumbnails, and do not directly incorporate information about the workflow. For example, there is no easy way to know before watching a clip which tools have been used, what initial and final products look like, or how complex the workflow is.

When learners watch a tutorial of their choice, video watching interfaces often fail to provide an efficient way to navigate between individual steps in the procedural workflow. This limits the learners to only sequentially navigate the workflow, while they might want to skip to a part where a specific tool is used, go back to the initial image, or repeat a step they missed. Learners have to rely on thumbnail previews and imprecise estimates for step-by-step navigation.

We introduce ToolScape (Fig. 1) , a system designed to address the challenges in finding and watching how-to videos. It features the Storyboard summarization and an interactive timeline. The Storyboard summarization (Fig. 1(a)) is a visual summary of the procedure, providing step-by-step information such as intermediate results and actions in between. It enables a quick review of multiple videos and their workflows without having to play them. The interactive timeline (Fig. 1(b)) adds direct links to intermediate results and actions in between, allowing learners to easily jump to or repeat a particular step inside a video clip. It aims to give learners more interactivity and control in workflow navigation, which is shown to encourage reinspection and improved comprehension of the workflow [Tversky et al. 2002] and to increase feelings of competence, self-determination, intrinsic interest [Lepper and Chabay 1985]. These interfaces leverage the step-by-step nature of the tasks, taking as input the annotations for each step.

To justify the choice of using the Storyboard summarization and understand the strengths and weaknesses of the method, we conducted a within-subjects laboratory study comparing three summarization methods: Storyboard, Fast Forward [Wildemuth et al. 2003], and Scene Clip [Ouyang et al. 2003]. Storyboard is a static summary of key steps in a tutorial, Fast Forward is an abbreviated version of the actual clip at 12 times the original speed, and Scene Clip is a compilation of two-second snippets of key steps. The participants spent less time in scanning videos with Storyboard and found it to be more informative, novel, and easy to use than the others, although there was no difference between the methods in how they support choosing relevant videos.

To evaluate the learning experience and usability of ToolScape, we conducted a within-subjects study with twelve participants. We asked the participants to perform graphical design tasks in Photoshop, which were to apply a given visual effect to a source image. They used either ToolScape or a baseline control interface which did not contain the Storyboard summarization and the interactive timeline. The participants showed a higher gain in self-efficacy and rated their own work higher when using ToolScape. Moreover, external judges also rated the designs produced by the participants using ToolScape higher. This result suggests that the way the participants browsed, navigated, watched, and interacted with video tutorials affected the learning outcome, even when the same learning resources were available. We attribute the success to providing more interactivity and learner control in video navigation, which adds support for the step-by-step annotation approach in ToolScape.

In the remainder of the paper we discuss related work and introduce the step-by-step annotation approach. Then we present ToolScape, a prototype system for browsing and watching tutorial videos. We report on results from two studies: the first study compares three video summarization methods, and the second study investigates the learner experience and usability of ToolScape in graphical design tasks. The paper concludes with a discussion of implications, limitations, and future work.

The contributions of this paper are summarized as follows:

— We introduce a video annotation approach that shows step-by-step information from existing videos to enhance the video instruction of procedural tasks.
— We demonstrate our approach with a prototype system that creates visual summaries and interactive timelines from step-by-step annotations.
— We report user evaluations that show the effectiveness of our system, measured by higher self-efficacy and internal/external ratings.

2. RELATED WORK

This paper builds on three bodies of previous research: video summarization (related to the *media type* of how-to videos), instructional design (related to the *presentation* of how-to videos), and tutorial enhancements (related to the *content* of how-to videos).

2.1. Video Summarization

The information foraging theory [Pirolli and Card 1999] suggests that just as food foragers follow clues about the prey's location, information seekers follow information scent to locate relevant information. In addition to the page title and URL, search interfaces often display a snippet of search results to provide the best possible information scent. For example, visual snippets [Teevan et al. 2009] offer a visual and contextual representation of search results.

Snippets for videos mostly rely on manually provided metadata such as title and description, not information directly collected from content itself. Designed to support browsing a video database, video summarization aims to provide better information scent of a video result and allows the user to assess relevance without watching the full video [Truong and Venkatesh 2007]. While researchers have attempted to identify the design space of video summarization [Truong and Venkatesh 2007; Song et al. 2010], few have focused on the design dimensions for how-to videos in particular.

The general design guidelines for video summaries might not apply to the learning context. For instance, in our pilot tests with different summary designs, users could not comprehend at all, and were frustrated by a fast forwarded tutorial video at 64x, which is a commonly used rate for generic videos [Wildemuth et al. 2003]. We attribute the difference to the unique characteristics of the domain. How-to tutorials are information-intensive and structured, where each step in a sequential workflow matters. They also consist of continuous, real-time demonstrations using a domain-specific set of vocabularies and commands that are hard to skim through.

Design dimensions we explored for tutorial video summaries are as follows, and Study 1 revisits this categorization to distill design guidelines.

— Should a summary be static (lists of keyframe images) or moving (video skims)? [Truong and Venkatesh 2007]
— Should text be used? Should audio be used? Should multiple media types be combined? [Marchionini et al. 2009]
— How to extract salient keyframes? [Girgensohn et al. 2001; Christel et al. 1998]
— How long should a summary be? [Taskiran et al. 2006]
— How much user control should be given in interacting with summaries? [Cheng et al. 2009; Wildemuth et al. 2003]

2.2. Instructional Design for Procedural Tasks

Learners unfamiliar with procedural tasks often turn to instructions. Eiriksdottir and Catrambone [Eiriksdottir and Catrambone 2011] distinguish three types of instructions based on how information is presented: principles, procedures, and examples. At the most general level, principles address how the system works by providing the theory of operation [Bibby and Payne 1993]. Procedural instructions describe and explain each step in the task [Bibby and Payne 1993]. Examples specifically demonstrate a concrete instance of the task. They differ from procedures in that they *show* the user what to do while procedures *tell* what to do.

Learners use analogical reasoning to map the example to the task at hand [LeFevre 1987]. Because learners often look for direct correspondence between the example and the task, examples with higher overlaps with the task lead to higher performance [Pirolli 1991]. Learners might suffer, however, from transfer problems that require applying knowledge to new situations [Wiedenbeck 1989]. Most video tutorials can be categorized as examples with demonstration, but many instructors combine explanations of procedure and principle, using voice. This might help learners understand the underlying process better, driving them away from analogical reasoning.

2.2.1. Interactivity of Instructions. Previous research almost unanimously supports that higher interactivity with the instructional content aids learning [Ferguson and Hegarty 1995; Tversky et al. 2002]. In the context of video instructions, Zhang et al. [Zhang et al. 2006] define interactive video as the use of computer systems to allow proactive and random access to video content based on queries or search targets. The lack of interactivity has been deemed a major problem with instructional video [Hadidi and Sung 1998]. Tversky et al. [Tversky et al. 2002] state that “stopping, starting and replaying an animation can allow reinspection”, which in turn can mitigate the challenges in perception and comprehension and further facilitate learning. In this work we add interactivity to how-to videos at the step level.

2.2.2. Worked Examples. Instructional design research has studied the value of worked examples for decades [Catrambone 1996; Atkinson et al. 2000; Renkl and Atkinson 2003]. A worked example is “a step-by-step demonstration of how to perform a task or how to solve a problem.” [Clark et al. 2005] It typically consists of a problem statement, a final solution, and stepwise actions [Atkinson et al. 2000]. ToolScape aims to make any existing video tutorial into a clear worked example by visualizing initial and final product, and retrieving and indexing solution steps.

2.2.3. Animated vs Static Instructions. There have been mixed results regarding the effect of instruction media types on learning and performance. Palmiter et al. [Palmiter and Elkerton 1991; Palmiter et al. 1991] showed that while users with animated demonstrations completed tasks faster with fewer errors, they showed weaker performance in a retention task after a week. Harrison [Harrison 1995] found that users with animated tutorials learned skills quicker and performed better than with textual tutorials. There was no significant difference between still and animated graphics, however.

Tversky et al. [Tversky et al. 2002] compiled research on animated instructions and concluded that there is no decisive evidence in favor of them. The authors suggest that the advantage is mostly from having more information or interactivity in the animated condition than the static condition. In a more recent meta-analysis of 26 studies comparing static graphics against animated instructions, animations showed higher effect size [Höffler and Leutner 2007]. In efforts to solicit when dynamic displays are more effective, the authors find that learners perform better when animated instructions are more representational, realistic, and related to procedural-motor knowledge. Recent HCI research either takes a negative position on including video instructions [Grabler et al. 2009; Kelleher and Pausch 2005], or advocates for using them in context [Grossman and Fitzmaurice 2010; Pongnumkul et al. 2011].

While the benefits of animated demonstrations are under debate, it is needless to say that a growing number of learners are using video tutorials to master new skills. Popular Photoshop tutorials on YouTube attract millions of viewers. This work focuses on improving the learning experience of existing tutorials on the web by giving the learner more interactivity and control in workflow navigation.

A recurring drawback of video instructions suggested by literature is pacing out of control [Clark and Mayer 2007; Grabler et al. 2009]. Videos can be too fast to follow, resulting in learners repeating certain parts multiple times and missing out important information. Other times they can be too slow to follow, resulting in learners losing interest in watching. ToolScape addresses this problem by providing interactive links that allow stepwise replay, skip, and jump operations.

2.2.4. Mixed Format Instructions. While a rich body of literature has studied the value of animated and static instructions, there has been little literature until recently on how multiple media formats can be combined to improve learning. In a study of mixed formats comparing static images, animations, and animations with static images, learners scored higher in questions related to a procedural task in the mixed condition [Arguel and Jamet 2009]. In the software tutorials context, Kong et al. [Kong et al. 2012] report that text+image is preferred to text only instructions or the graph representation of a workflow. Chi et al. [Chi et al. 2012] show that learners using a mixed tutorial (static + video) made fewer errors than using static or video alone. Their work attaches segmented video clips to a step-by-step tutorial. Our work is indeed an attempt at the inverse: can the step-by-step nature be incorporated into a complete, continuous video instruction?

Clark and Mayer [Clark and Mayer 2007] note that animations are good for physical procedures, while still images are good for conceptual processes. A how-to workflow often involves both types of processes. For example, in Photoshop, planning the overall design approach might be conceptual, but using selection tools to select an irregular object might be physical. We argue that video browsing and watching interfaces can benefit from incorporating more images and text.

2.3. Tutorial Enhancements

Recently, there has been active research in HCI on enhancing the tutorial experience. Systems automatically generate interactive tutorials by demonstration [Chi et al. 2012; Fernquist et al. 2011; Grabler et al. 2009] or help learners follow along examples [Fernquist et al. 2011; Lee et al. 2011; Pongnumkul et al. 2011].

Perhaps the closest tutorial enhancement system to ToolScape is MixT [Chi et al. 2012], which generates tutorials by demonstration from a Photoshop plugin. It creates a mixed media format tutorial by replacing certain steps in an HTML tutorial with short video clips. ToolScape differs from MixT in the following ways. Firstly, ToolScape does not break the instruction into segments to help learners maintain the context. It does not alter a continuous stream of videos, while MixT provides segmented video clips for some steps. Secondly, ToolScape contains audio narration if the source video does, which is almost always the case, while MixT videos do not include audio. Finally, ToolScape does not require access to the in-application context at the tutorial generation time. It works on top of readily available videos on the web, without having to install an application-specific plugin.

Waken [Banovic et al. 2012] detects cursor movement or icon clicks from video tutorials with computer vision techniques. The Waken Video Player adds tool usage events to an interactive timeline, although it does not display work-in-progress frames to summarize the workflow. ToolScape combines tools and works in progress for a richer recovery of a how-to demonstration. But Waken's event detection technique can potentially be used in ToolScape to collect command annotations.

Pause-and-Play [Pongnumkul et al. 2011] and FollowUs [Lafreniere et al. 2013] address the difficulty in following the steps from a video tutorial. Pause-and-Play detects the current workspace and compares against a video tutorial the learner is watching. It automatically pauses the video until the learner made the change in the step, and plays when the learner is done. FollowUs creates custom tutorials by matching user demonstrations with each step in the original tutorial. It allows users to contribute to the growing tutorial library by attaching their own demonstrations to the original tutorial. While these systems are similar to ToolScape in leveraging the step-by-step nature embedded in a video, ToolScape provides more flexible ways to navigate a video clip, especially when the goal is not to replicate steps but to navigate steps non-sequentially and apply partial solutions to the task at hand.

Delta [Kong et al. 2012] is an interface for comparing multiple workflows in static HTML documents. This work addresses a similar learner problem to our work: browsing through a repository of tutorials. ToolScape differs in that it focuses on summarizing video tutorials as opposed to static ones, which we believe have unique design challenges.

3. THE ANNOTATION APPROACH

ToolScape improves the learning experience of how-to videos by retrieving useful information from the videos and adding interactions based on the information. *Annotation* refers to the process of collecting information about each step in a workflow. This section introduces the annotation approach in ToolScape, which takes into account the procedural nature of how-to tasks.

3.1. Why and What to Annotate?

A design opportunity for enhancing how-to videos is that they often have a well-defined structure [Grossman et al. 2010]. First, in a step-by-step instruction, a step represents an action responsible for advancing the state of the task (Fig. 2). A step is a unit of progress in the workflow, often identified by an operation or tool usage. Scanning which operations were executed for each step allows learners to comprehend how an effect is achieved. Second, tasks are visual in nature, and progress can be visually tracked (Fig. 3). Capturing intermediate

Fig. 2. How-to videos contain steps that sequentially advance the state of the task. In this example, the Gradient map tool in Photoshop was applied to make the image more black-and-white. Annotating commands enables visualizing a change history.

Fig. 3. In many how-to videos, progress is visually trackable as in this Photoshop tutorial example. Annotating works in progress enables a quick scan of the entire workflow.

works in progress and displaying them sequentially to learners allow them to review the workflow at a glance and spot points of interest. Our annotation pipeline combines the two properties to accurately summarize an entire workflow, therefore collecting both tools used and works in progress for each step. Although this paper mainly focuses on video tutorials related to graphical design software due to its high penetration and availability of large web repositories of existing videos, the techniques discussed in this paper are applicable to other how-to videos with the discussed properties.

For Photoshop tutorials, our annotation method labels all menu and tool operations and their before and after images. All labels contain time marks for the associated occurrence in the workflow, which point to a moment an event is first made visible on the screen. Each video contains two types of labels: *command* and *image*.

The *command* label represents a tool or menu operation in Photoshop (e.g., paint bucket tool, duplicate layer menu). Parameter settings inside a dialog box (e.g., setting the foreground RGB color values to (213, 47, 53)) are not annotated. This is because they are hard to understand without the surrounding context, and the learner is likely to watch from the encapsulating step (e.g., selecting the foreground color menu) anyway. Also, these operations often involve a subtle, physical procedure, which is better learned by watching the video. The menu hierarchy information is also stored (e.g., Select > Modify > Feather) because a full path helps novices locate the tool and some tool names are too generic without ancestor menu information (e.g., Layer > New vs File > New).

The *image* label corresponds to every work in progress that is a result of a *command* and visually distinct from the previous one. For example, intermediate works in progress from parameter tuning are ignored because they are not results of a command. Also, not every command yields an *image* label because some commands do not have any effect on the current work in progress (e.g., adding a new layer). Finally, initial and final images are labeled with a special tag. Fig. 2 would translate to two *image* labels and a *command* label in between, while Fig. 3 would be five *image* labels. Fig. 6 visualizes a complete workflow on a timeline with *command* (top layer) and *image* (bottom layer) labels.

3.2. Preliminary Video Corpus with Manual Annotations

To experiment with different metadata formats and representations, we created a small corpus of hand-annotated Photoshop video tutorials. We built a video annotation interface to create the database (Fig. 4). This corpus led to a richer understanding of how video

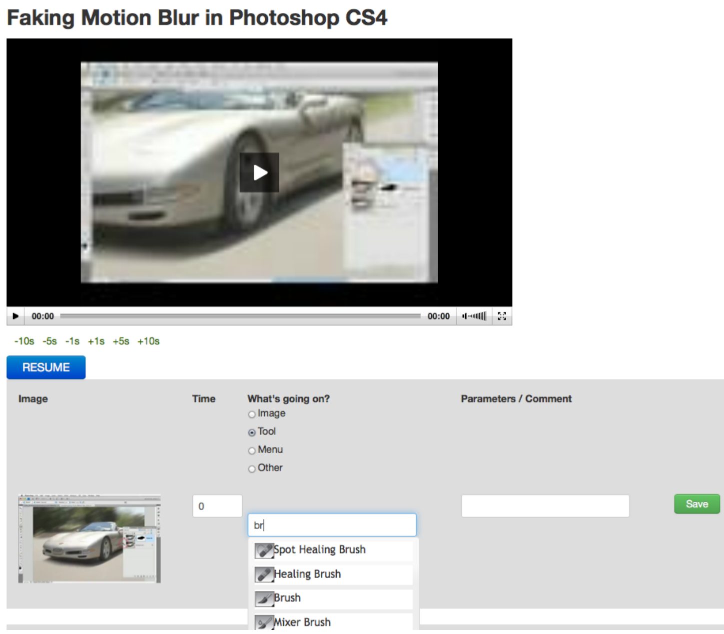


Fig. 4. A tutorial annotation interface used to generate the initial database of annotations. The interface captures a timestamp when a labeler presses the *CAPTURE* button, which opens the annotation dialog in gray and turns into *RESUME*. Then the labeler can enter the annotation type (image or tool), the tool name using auto-completion customized for Photoshop for the purpose of this work, and comments. For capturing precise event timing, it also includes fine-grained playback controls with rewinding and fast forwarding 1, 5, and 10 seconds.

Table I. Hand-annotated database of 55 Photoshop how-to videos

	length	**# of labels**	**# of images**	**# of commands**	**start point**	**end point**
mean	273.0 sec	14.2	6.4	7.8	13.7%	9.9%
stdev	100.7	7.4	3.2	4.5	8.0	7.2

We created a corpus of 55 video tutorials and hand-annotated them for commands and work-in-progress images. The videos on average contain 7.8 commands and 6.4 images, summing up to 14.2 labels. The first 13.7% and last 9.9% of the video content is non-workflow-related, which means the first image of the workflow appeared after watching 13.7% of the video.

tutorials are structured and served as a basis for interface prototypes and user studies. The corpus consists of 55 videos covering 6 common Photoshop effects: teeth whitening, motion blur, Lomo effect, retro effect, photo-to-sketch effect, and background removal. The videos were collected from YouTube's top search results by typing in "Photoshop [effect name]". We excluded videos shorter than 2 minutes or longer than 10 minutes. More information on the corpus can be found in Table I. In constructing the initial annotation database from 55 tutorials, researchers on average took 2–3 times the original video length to generate annotations for a tutorial video clip.

3.3. Annotation Methods

Although this paper focuses on enhancing the learning experience rather than collecting the annotations, the ToolScape system inherently relies on the annotations. In order for ToolScape to be of practical use, it is essential to be able to collect annotations for a large number of how-to videos in an efficient and scalable way. We briefly review existing approaches and consider several possible future methods for annotating tutorial videos at scale.

One existing annotation approach requires in-app access at the tutorial generation time, often packaged as plug-ins [Chi et al. 2012; Fernquist et al. 2011; Grabler et al. 2009; Grossman and Fitzmaurice 2010; Grossman et al. 2010]. Although the plug-in approach guarantees richer, more accurate annotations, it suffers from a limitation that tutorial instructors must install the plug-in before creating a tutorial. More fundamentally, many software applications do not support programmable in-app access and therefore are inaccessible to this approach. Furthermore, plug-ins do not exist for most how-to tasks outside of software applications, which limits the scope of the plug-in method.

An alternative approach is to annotate existing tutorials after the fact. In this work, **manual annotation** by researchers took 2–3 times the original clip duration on average. While the manual method yields consistent, high quality annotations, it is not scalable. We explore three possible methods that collect annotations after the fact: computer vision, crowdsourcing, and *learnersourcing.*

Computer vision is a cost-effective and automatic way to collect annotations, but it requires high-resolution images for high accuracy and training data to yield good results [Banovic et al. 2012]. These automatic methods are not perfect, but systems such as Sikuli [Yeh et al. 2009], Prefab [Dixon and Fogarty 2010], Pause-and-Play [Pongnumkul et al. 2011], and Waken [Banovic et al. 2012] can at least provide partial annotations.

Crowdsourcing can be a viable solution to complement computer vision by providing low-cost training data. Park et al. [Park et al. 2012] show that 3 crowdsourced workers produced video annotation quality comparable to those created by experts. We are currently experimenting with alternative task designs to reach high accuracy [Nguyen et al. 2013]. As with other crowdsourcing systems, quality control and an associated cost remain a challenge.

Our future work will mainly focus on **learnersourcing**, which leverages activities performed by learners as they naturally interact with the tutorials as useful input to the system [Kim et al. 2013]. Learners are a motivated and qualified crowd who are willing to watch how-to videos for their learning purposes. For example, we envision injecting quizzes while the learner watches videos in ToolScape, whose answers would serve as annotations and training data. We believe this approach can improve learning with quizzes based on retrieval practice [Karpicke and Blunt 2011], while the learner's input provides the system with high quality annotations at low cost.

In summary, there are potentially many viable methods to collect the annotations of commands and work-in-progress images used in tutorial videos. The contribution of the present work is exploring interaction techniques to support browsing and watching how-to videos, which can use the stepwise annotations acquired from any of these methods. The rest of the paper focuses on improving the learning experience involving how-to videos.

4. SYSTEM DESCRIPTION

A database of annotations enables us to explore video summarization methods and interaction techniques to support browsing and watching how-to videos. This section introduces ToolScape, a prototype system designed to support learning from how-to videos. ToolScape consists of the browsing interface and the watching interface. The browsing interface uses Storyboard as its video summarization method, and adds interaction techniques for reviewing and filtering the results. The watching interface adds an interactive timeline to support non-sequential and step-level in-clip navigation.

ToolScape is a web interfaces, built with HTML5, CSS3, and Javascript, with an open-source video player. It is designed to work with readily available online videos that can be found on sites such as YouTube. We followed an iterative design process with multiple rounds of pilot user feedback and refinement.

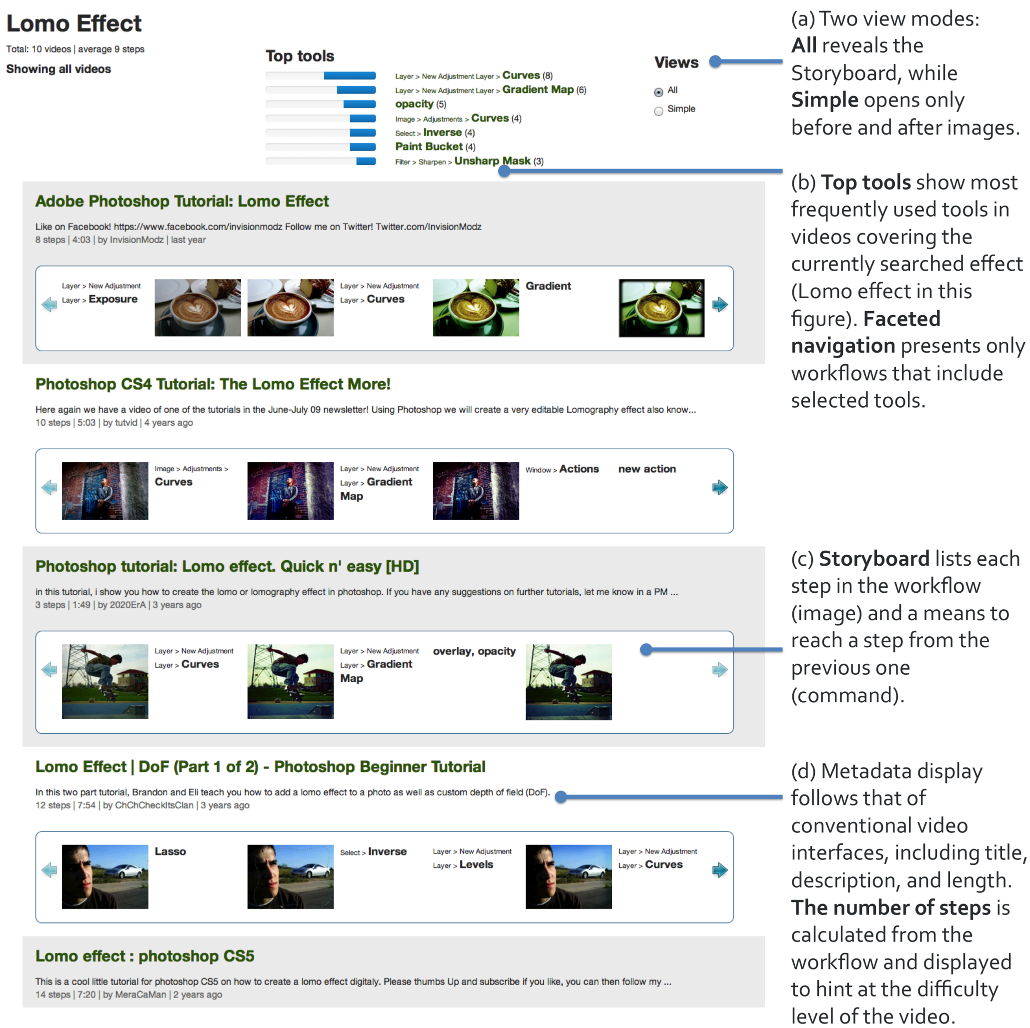


Fig. 5. ToolScape's browsing interface uses the Storyboard summary (c) to linearly present a workflow. A simple view mode (a) only shows before and after images for quicker skimming through the results. Top tools (b) are sorted by frequency, and they can be clicked to filter workflows. Metadata display (d) shows textual information about each video.

4.1. Browsing Interface

The browsing interface allows learners to quickly scan, filter, and review multiple videos without having to play one. It displays a sequential workflow for each video using the Storyboard summarization method. We chose Storyboard as our summarization approach based on results from informal pilot studies. We report on a study in the next section to evaluate this choice by comparing Storyboard to two alternative methods.

Storyboard (Fig. 5(c)) is a static video summarization method that horizontally lists keyframes. Keyframes include each intermediate result in the workflow (image) and a means to reach a result from the previous one (command). The summary generator samples every frame specified in the given image and command annotations. The summarization displays commands as text to highlight the semantic difference between image and command. This

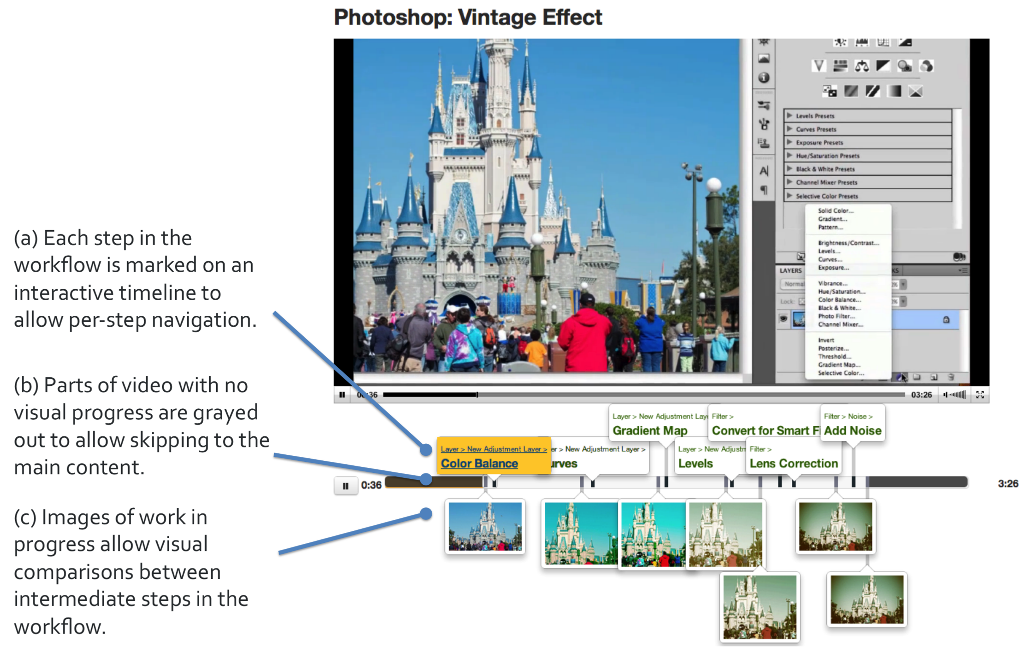


Fig. 6. ToolScape gives control to a learner when watching a how-to video with non-sequential ways to navigate a workflow. ToolScape's interactive timeline uses the top stream (a) for commands and bottom steam (c) for works in progress. Gray regions at both ends (b) help learners save time by skipping parts with no visual progress.

image+text representation visually distinguishes the two types of information, and further enables textual indexing and filtering with commands.

Two view options (Fig. 5(a)) turn the Storyboard summarization on and off. *All* reveals the Storyboard summarization, and *Simple* shows only before and after images.

Top tools (Fig. 5(b)) lists most frequently used tools for the current set of videos that are about a searched term. In addition, faceted navigation presents only workflows that include selected tools. A new filter is added by clicking on the name of a tool, and multiple filters can be applied for more fine-grained filtering.

4.2. Watching Interface

We use an interactive timeline to play annotated video clips as in many other existing systems [Banovic et al. 2012; Grossman et al. 2010; Pongnumkul et al. 2011]. The player page opens by clicking on a clip title from the ToolScape browser. The top and bottom streams represent commands and intermediate results, respectively (Fig. 6(a), (c)). The visual separation allows scanning just the command names or works in progress. Clicking on commands or images moves the player slider to 5 seconds before the moment. The 5-second buffer time, which we determined from pilot testing, helps learners catch up with the context preceding the moment. A full menu hierarchy for each tool is displayed to make it easy to activate the tool and to disambiguate between similar tool names (e.g., Layer > New vs File > New are different operations, sharing the same leaf-level tool names). To avoid occlusion between tool or image callouts, when two items are within a threshold distance, ToolScape applies a vertical space to guarantee a fuller view of each.

Table II. Three video summarization methods used in Study 1

Method	Video	Audio	Length
Storyboard	No	No	# keyframes (static)
Fast Forward	Yes	No	(original length / 12) sec
Scene Clip	Yes	Yes	(2 * keyframes) sec

Storyboard is the only type without video, and Scene Clip is the only with with audio. Storyboard and Scene Clip operate on top of manually annotated keyframes, while Fast Forward relies on uniformly sampled frames (12x).

A distinct feature in ToolScape player is the visualization of the beginning and the end of the workflow, which marks the first time the initial image and the final image appears, respectively (Fig. 6(b)). Pilot user observations suggest that learners often want to skip to the main part of the tutorial. The beginning and end of a video often include unnecessary setup instructions (e.g., opening Photoshop and locating a file) or personal comments (e.g., advertising to rate the clip), and our annotated database allows skipping to the step-by-step workflow. In the videos in our samples, 13.7% in the beginning and 9.9% at the end on average was time with no progress in the workflow. This suggests that a user can save at least 20% of their watching time if they skipped the gray regions on the timeline.

5. STUDY 1: COMPARING VIDEO SUMMARIZATIONS

We conducted a two-part study in a single session to investigate both the summarization method and the overall learning experience. The first focuses only on video summarization methods addressing learners' browsing phase, while the second study investigates the end-to-end design task experience.

One of the major design decisions in the ToolScape browsing interface is the use of the Storyboard video summarization. The goal of the first study was to compare the Storyboard technique against other summarization methods to evaluate the summarization accuracy and scanning time, understand the relative strengths and weaknesses, and draw design guidelines for video tutorial summaries.

5.1. Interfaces

Three interface conditions used in the study were Storyboard, Fast Forward, and Scene Clip.

— **Storyboard** is a static strip of keyframes, including both image and command labels. Keyframes are drawn from our annotation corpus, capturing moments of tool usage and salient visual changes. This is what ToolScape uses as its summary.
— **Fast Forward** [Wildemuth et al. 2003] is an abbreviated version of video with 12x playback rate. In the literature it is common to use a much higher rate (e.g. 64x or 128x), but we adjusted the speed after pilot testing to adapt to tutorial videos.
— **Scene Clip** [Ouyang et al. 2003] is a compilation of 2 second clips for every keyframe in the original video. As in the Storyboard condition, the keyframes are retrieved from the annotation corpus.

This selection covers both static and moving types as well as audio and video modalities (Table II). Westman [Westman 2010] conducted a similar comparative study between Storyboard, Fast Forward, and Scene Clip.

The mean number of labels in the corpus of 30 videos used in the study was 12.8 (σ = 5.7) including both images and commands, and the mean video length was 282.7 seconds (σ = 98.2). Storyboard had mean 12.8 frames statically displayed on screen, Fast Forward clips were 23.6 (= 282.7/12) seconds long, and Scene Clip 25.6 (= 12.8×2) seconds long.

5.2. Hypotheses

The study compared the three summarization methods by looking at 1) learners' estimation of relevance, interest, and skill level of a video, 2) time to task completion, and 3) subjective satisfaction. As a proxy for accuracy, we measured the difference in the estimation of relevance, interest, and skill level before (with access to only the summary) and after watching a full clip. A smaller difference in the estimation would indicate that the video summary accurately represented the actual content.

We hypothesize that in the learning context, Storyboard is more effective than Fast Forward or Scene Clip. Previous literature on video summarization does not converge on a single best method, and little research has been done for summarizing how-to videos in specific. Our design of Storyboard was guided by both the literature and multiple rounds of pilot testing with alternative designs, which revealed the following two design goals. First, Storyboard allows for efficient side-by-side comparison, which is essential for learners browsing multiple video results to find a relevant video. Also, Storyboard lets learners control the navigation at their own pace to aid comprehension by listing all important steps in the workflow.

Our hypotheses in this study are:

H1 Learners' estimation of relevance, interest, and skill level will change less after watching a full clip with Storyboard than with Fast Forward or Scene Clip.

H2 Storyboard is faster to scan than Fast Forward or Scene Clip.

H3 Learners are more satisfied with Storyboard than with Fast Forward or Scene Clip.

5.3. Participants

Twelve participants were recruited through a university mailing list and online community posting. The mean age was 25.2 ($\sigma = 3.2$), with 8 males and 4 females. Most of them rated themselves as novice Photoshop users, but all had at least some experience with Photoshop. To ensure they were novices for the specific skills need in the study, we screened applicants with a web survey. They received $30 for maximum 2 hours of participation in both Study 1 and Study 2. Participants could choose to do the study on either a Mac or a PC for natural setting.

5.4. Tasks and Procedures

The study consisted of three task and three interface conditions. We used a within-subject design and counterbalanced both the task and interface order. To limit the scope of the study to the browsing and selection behavior, we asked the participants to only imagine they needed to do them in Photoshop. Three image manipulation tasks used in the study were to (1) apply motion blur to an image, (2) apply Lomo effect to an image, and (3) remove complicated background from a photo.

A study session started with introduction and a tutorial for the current task and interface. For each interface condition (Storyboard, Fast Forward, and Scene Clip), we gave participants 10 video summaries on a single page that are related to the task (e.g., motion blur). They were asked to rate every video in 7-Likert scale after viewing the summary by answering the following three questions (Fig. 7).

— Q1. This video is relevant to my task. (relevance)
— Q2. This video contains skills I'd like to learn. (interest)
— Q3. This video is suited to my skill level. (skill level)

Once all ratings were submitted, participants watched their three highest-rated videos in full. This setup simulated the most likely selection of videos to watch in an actual search task. After they watched each full clip, they answered the same three questions again and also described what was different in the full video from their expectation. Then partici-

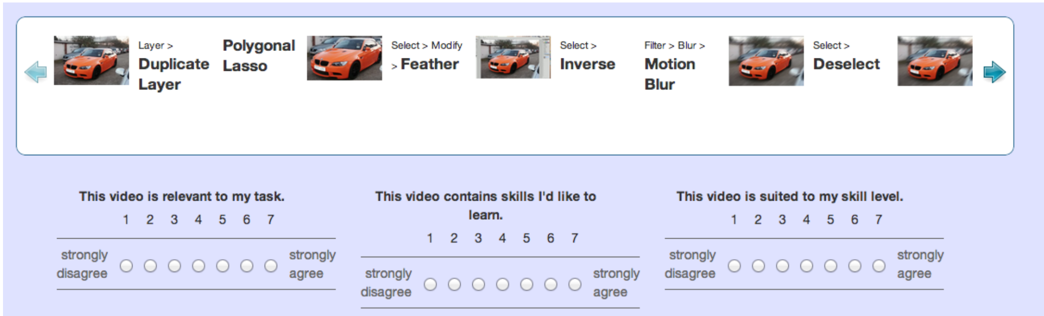


Fig. 7. The rating questions the participant answered for each video summary. This example shows a Storyboard summarization. For Scene Clip and Fast Forward, Storyboard is replaced with a video player but the same questions are displayed.

pants answered Likert questions on satisfaction and experience with the given interface, and commented on the strengths and weaknesses of the summarization method in free-form text (Appendix A.1). The whole procedure repeated for three interfaces, each time with a different task and video selection.

6. STUDY 1 RESULTS

H1, a smaller change in estimations of relevance, interest, and skill level for Storyboard, is not supported. There was no significant difference in estimations between the interfaces in a Friedman test. An explanation might be that individual differences outweigh summary differences.

H2, shorter task time for Storyboard, is supported. Storyboard was faster to comprehend than Scene Clip. The mean time to completion was 328.3 ($\sigma = 49.8$), 396.6 ($\sigma = 134.7$), and 535.4 ($\sigma = 150.7$) seconds for Storyboard, Fast Forward, and Scene Clip, respectively. In two-way repeated measures ANOVA, we found significant main effects of interface ($F(2,6)=34.091$, $p < 0.001$) on task completion time. A pairwise comparison with Bonferroni correction revealed a significant difference between Storyboard and Scene Clip ($p < 0.01$). A reason might be that it is natural to spend less than two seconds per image or command in Storyboard, and skimming is possible. Scene Clip, however, forces the user to spend two seconds on each keyframe. A user noted, “I have no control over what gets cut out, so I felt a bit disoriented.”

H3, higher satisfaction with Storyboard, is not supported. There was no significant difference in the overall user satisfaction score between the interfaces. Out of seven questions related to user satisfaction in 7-Likert scale, Storyboard (overall $\sigma = 4.6$, $\sigma = 0.9$) scored highest in five (easy to use, understandable, informative, can replace watching, and useful), Fast Forward (overall $\sigma = 4.4$, $\sigma = 0.9$) in two (enjoyable and representative of actual content), and Scene Clip (overall $\sigma = 4.4$, $\sigma = 0.8$) in none (Fig. 8). In ordinal logistic regression, interface was significant for measures of informativeness ($p<0.0001$), novelty ($p<0.0001$), and ease of use ($p<0.01$).

6.1. Design Dimensions for Video Tutorial Summarization

User observations and comments reveal relative strengths and weaknesses of each summarization method. These comments mostly fall under the design dimensions mentioned in the Related Work section. Here we provide a deeper analysis to guide the design of better video tutorial summaries.

6.1.1. Static vs moving. Static Storyboard allows skimming and skipping while moving Fast Forward and Scene Clip force watching sequentially. In a static visualization, managing

Interface	easy to use	easy to understand	enjoyable	informative	represents content well	relevant	novelty	advanced	can replace video	useful	total
Storyboard	**5.75**	**4.92**	4.08	**5.42**	4.17	5.00	**6.00**	4.33	**3.08**	**5.08**	**4.64**
Fast Forward	5.33	4.58	**4.58**	3.75	**5.17**	**5.42**	5.00	**5.08**	2.67	4.92	4.43
Scene Clip	5.08	4.42	4.42	5.17	4.58	5.00	4.58	3.83	2.67	4.92	4.46

Fig. 8. Satisfaction questionnaire results. While there was no overall statistical significant difference between the summarization methods, Storyboard scored highest in five questions, with three showing statistical significance (informative, novel, and easy to use). The highest score from each question is highlighted.

screen space can be challenging because it needs to display all summary content in one view. A static summary should avoid too small thumbnails or excessive scrolls to help learners easily comprehend the workflow with minimal effort. In a moving summary, the play rate determines the viewing experience. An optimal rate varies across domains and individuals. Our pilot testing shows that video summaries for how-to videos need slower rate than TV shows or movies due to high information density. Personalizing the pace might be necessary, as noted by the participants' conflicting comments on Fast Forward: "too fast" vs "would prefer faster fast-forward".

6.1.2. User control. Not having any control over the pace and flow might frustrate learners. Participants in the Scene Clip condition complained that the pace was not catered for them, which makes it difficult to follow as time progresses. Storyboard, on the other hand, gives navigational control to users, of which many participants took advantage to non-sequentially analyze a workflow. But some users felt inconvenienced by having to manually scroll the visualization.

6.1.3. Text vs no text. Storyboard lists the tools used in a tutorial in text, along with contextual thumbnails. Text might be effective for learners who quickly scan, review, and compare multiple workflow summaries. A participant mentioned "...the storyboards could actually replace the video since all the important functions are listed." Another learner noted that "Storyboard showed the brief history and so I could try roughly without seeing the full video. As a result, it saved my time significantly."

6.1.4. Audio vs no audio. Scene Clip audios are useful, as they can provide additional information about the video. In contrast, Fast Forward and Storyboard fared poorly when a tutorial relied on verbal explanation as these two techniques did not include audio. A participant commented, "[Scene Clip helps me] easily find the preferred circumstances (ex: teacher's voice, pronunciation, settings of the program etc..)" A challenge is in avoiding a cut-off. In the best case, tool names are spoken and how to initiate it is displayed within two seconds in Scene Clip summaries. But in other cases, sound or an important scene is cut off in the middle, which several participants found to be frustrating.

6.1.5. Retrieving keyframes. The uniform sampling used in Fast Forward covers all parts in the process but sometimes unnecessary parts are captured or important parts are missed. Users liked it for "not missing any information" and having a "great way to get the big picture". But the rapid passing of frames sometimes fails to capture salient changes in the workflow.

6.1.6. Length. Static formats take less time to comprehend, at the cost of omitting details that might be useful in some cases. This trade-off is described by a participant, "I think [Storyboard] is a great method, that gives you only the important parts of the video. Some details were lost, which meant I needed to look at the whole video sometimes, but I knew where to look from the summaries." The study results show that the shorter comprehension time did not lead to worse relevance estimations, which might indicate that Storyboard was more efficient than the others.

6.2. Why Storyboard?

While the findings do not provide conclusive evidence for a specific type of summarization, we decided to use Storyboard as our summarization method in the ToolScape system for the following reasons: First, Storyboard allows a quicker scan of a workflow without sacrificing relevance estimation performance or satisfaction. Time is an important factor especially when the learner has to review multiple results to decide which tutorial to watch. Second, Storyboard gives the learner more control. With dynamic summarization formats, it is difficult to control the pace and playback sequence, which some participants complained about.

7. STUDY 2: END-TO-END DESIGN TASK

In the second part of the study, participants were engaged in design tasks in Photoshop. The goal was to compare the skill-learning experience of ToolScape against a standard video interface. We had the same participants as Study 1, in a two-part study.

7.1. Interfaces

ToolScape and a baseline video browsing interface were used in the study. The baseline (Fig. 9) has a browsing and watching interfaces similar to ToolScape, but its browsing interface does not include the Storyboard summary, tool filtering, and view modes. It has a thumbnail for each video, along with basic metadata such as title, description, length, upload date, and uploader. The watching interface does not include the interactive timeline. In the ToolScape condition, the mean number of labels in the corpus of 20 videos was 17.35 ($\sigma = 8.9$) including both images and commands, and the mean video length was 272.35 seconds ($\sigma = 105.9$).

7.2. Hypotheses

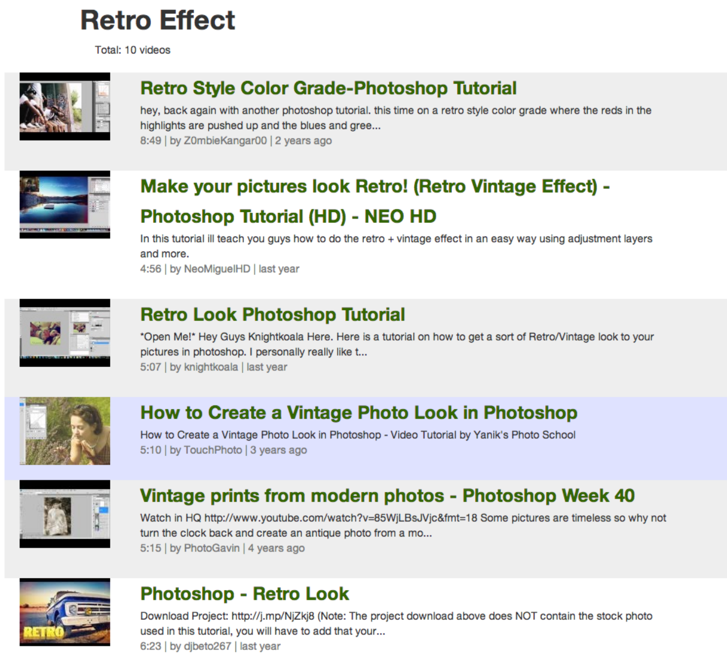
In addition to external ratings, our measures of success include self-rating and self-efficacy. These measures are more than just user preferences: educational psychology research shows that self-efficacy is an effective predictor of motivation and learning [Bandura 1977; Zimmerman et al. 1992]. Positive self assessment has also been shown to accurately predict learning gains [Schunk 1990]. We chose not to count errors made in repeating tutorial steps as in [Chi et al. 2012], because our goal was to help users explore and learn new skills in open-ended design tasks.

We hypothesize that ToolScape supports learners in performing design tasks while using video tutorials.

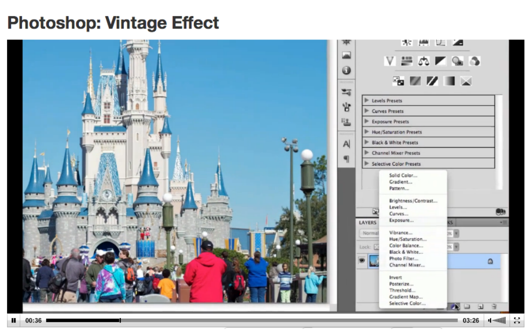
H1 Learners complete design tasks with a higher self-efficacy gain with ToolScape.
H2 Learners' self-rating on their work is higher with ToolScape.
H3 Learners' design is rated higher by external judges with ToolScape.
H4 Learners perceive design tasks to be easier with ToolScape.
H5 Learners show higher satisfaction with ToolScape.

7.3. Tasks and Procedures

The study consisted of two task and two interface conditions. 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, the participants were provided with the same set of videos. The interface for browsing and watching the videos was the only difference. In addition, we disallowed the participants to search for other tutorials on the web to ensure that any effect found in the study comes from the interaction method, not the content.



(a) Baseline browsing interface



(b) Baseline watching interface

Fig. 9. The baseline interface used in Study 2. For browsing, the interface contains a thumbnail and metadata but no information about the internal content and workflow information. For watching, the interface only presents a video player with thumbnail preview on hovering mouse on a timeline.

After a tutorial task that covered all the features of the interface, they were asked self-efficacy questions. The questions were adopted and modified from Dow et al. [Dow et al. 2010], whose study also measured participants' self-efficacy changes in a design task. In a scale of 1 (not confident at all) to 7 (very confident), the questions asked "How confident are you...":

— with solving graphic design problems?
— at understanding graphic design problems?
— with applying design skills in practice?
— with incorporating skills from video tutorials in your design?

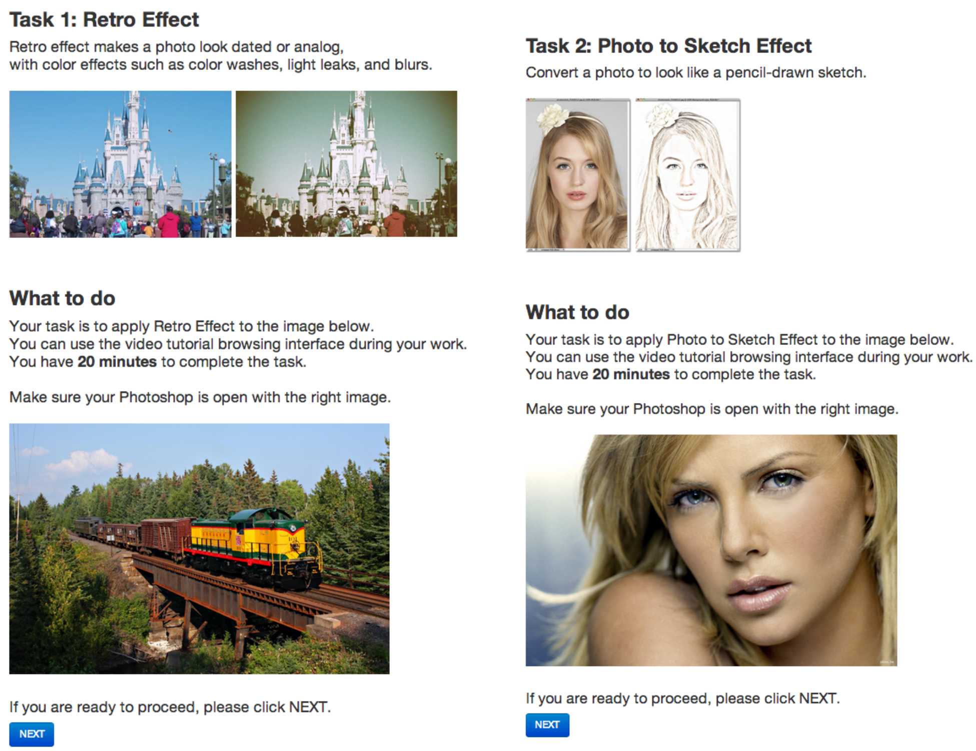


Fig. 10. The task instructions are shown before the participant starts working on their design task. This page includes a brief description of the effect to be implemented with an example featuring a before and after image pair. Then a source image is presented, which is also open in Photoshop so that the participant can make changes.

Then a 20-minute task started, and the participant could freely browse and watch the given 10 videos and work on their task in Photoshop. The task instruction is shown in Fig. 10. After the task, we asked questions on task difficulty, self-rating, and interface satisfaction (Appendix A.2). To evaluate the usability of the interface, we asked how helpful each UI component was in completing the task. To help recall the task experience, we printed the ToolScape UI screenshots on paper with annotated feature names. We also asked the self-efficacy questions again to observe any difference.

After the sessions, we asked four external judges to evaluate the quality of all submissions by ranking them in a blind setting. They ranked (1-best, 12-worst) the submissions, which encouraged them to directly compare designs. We asked the judges to rank based on how well the designs accomplished the given task. This specific instruction allowed the raters to make their judgments in the context of the study task.

8. STUDY 2 RESULTS: DESIGN TASK AND LEARNING

H1, higher self-efficacy for ToolScape, is supported (Fig. 13). To see if learners felt more confident in their design skill after completing the task with ToolScape, we look at how the self-efficacy score changed during the task in each interface condition. For the four self-efficacy questions in 7-Likert scale, we take the mean as the self-efficacy score. The participants' mean initial score was 3.8, and with Baseline the score after the task was 3.9 (+0.1) whereas with ToolScape the score was 5.2 (+1.4). A Mann-Whitney's U test shows a significant effect of interface ($Z=2.0586$, $p<0.05$).

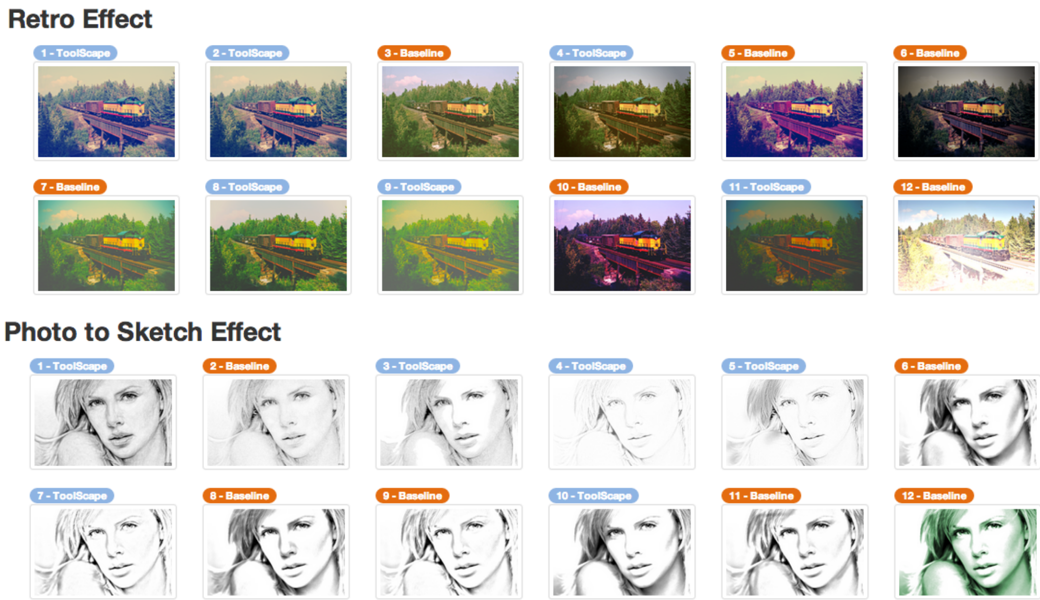


Fig. 11. Rank-ordered designs created by the participants in Study 2. For each image, the top label displays the rank of the image and the condition it was created in (either ToolScape or Baseline). The mean rankings were lower in ToolScape (5.7) than in Baseline (7.3), which maps to higher quality designs.

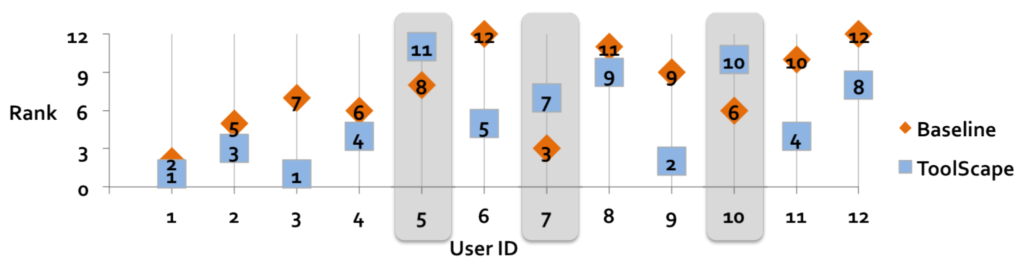


Fig. 12. Ranking information per participant. For each user, a ToolScape marker and a Baseline marker are added to show the ranking of the design produced for the conditions (lower is better). 9 of the 12 participants performed better with ToolScape than with Baseline. P5, P7, and P10 performed worse with ToolScape (gray background).

H2, higher self-rating for ToolScape, is supported (Fig. 14). The mean ratings for ToolScape and Baseline were 5.3 and 3.5, respectively. A Mann-Whitney's U test shows a significant effect of interface ($Z=2.6966$, $p<0.01$). This means participants rated their own work quality higher when using ToolScape.

H3, higher external rating for ToolScape, is also supported (Fig. 15). The overall ranking was computed by taking the mean of the four judges' ranking data. Analyzing the mean ranks suggests that designs in the ToolScape condition were ranked higher. Fig. 11 shows the designs created by the participants in each condition. The ranking method yielded high inter-rater reliability (Krippendorff's alpha=0.753) for ordinal data. The mean rankings (lower is better) for designs created in ToolScape and Baseline conditions were 5.7 and 7.3, respectively. A Wilcoxon Signed-rank test indicates that there is a significant effect of

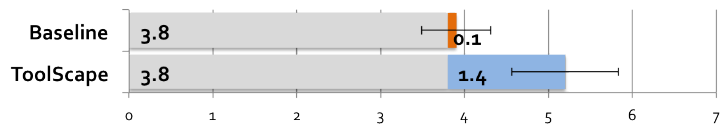


Fig. 13. H1: Higher self-efficacy gain with ToolScape is supported. Error bar: standard error.

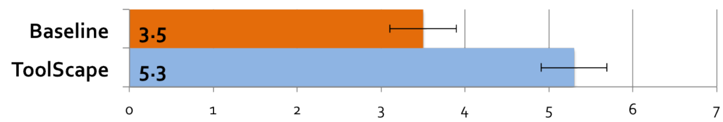


Fig. 14. H2: Higher self-rating with ToolScape is supported. Error bar: standard error.

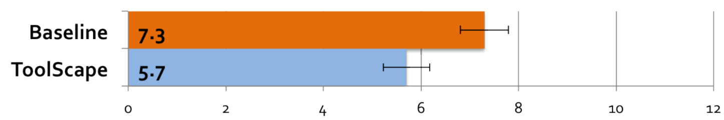


Fig. 15. H3: Higher external-rating with ToolScape is supported. Error bar: standard error.

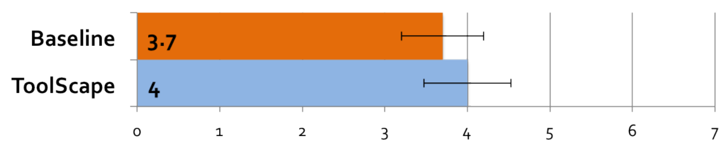


Fig. 16. H4: Task difficulty perception is not different between interfaces. Error bar: standard error.

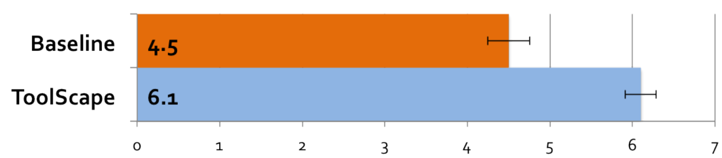


Fig. 17. H5: Higher user satisfaction with ToolScape is supported. Error bar: standard error.

interface (W=317, Z=-2.79, $p<0.01$, r=0.29). Furthermore, nine of the twelve participants performed better with ToolScape, as shown in Fig. 12.

We also looked at the order of the tasks to see if the better quality in ToolScape is confounded with learning effects. If there was a learning effect in place, the first task the learners performed would show worse ranking than the second task. The results show that this is not the case. The mean ranks for the first and second were identical (6.5) in both conditions ($\sigma = 4.2$ for the first tasks, $\sigma = 3.0$ for the second tasks), which indicates that task order had no effect on the output quality.

H4, easier task difficulty perception for ToolScape, is not supported (Fig. 16). 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 to be easier but they still produced better designs with more confidence.

H5, higher user satisfaction for ToolScape, is supported (Fig. 17). The mean ratings for ToolScape and Baseline were 6.1 and 4.5, respectively. We ran a Mann-Whitney's U test to evaluate the difference in learner satisfaction questions in 7-Likert scale, which showed a significant effect of interface (Z=2.7895, $p<0.05$).

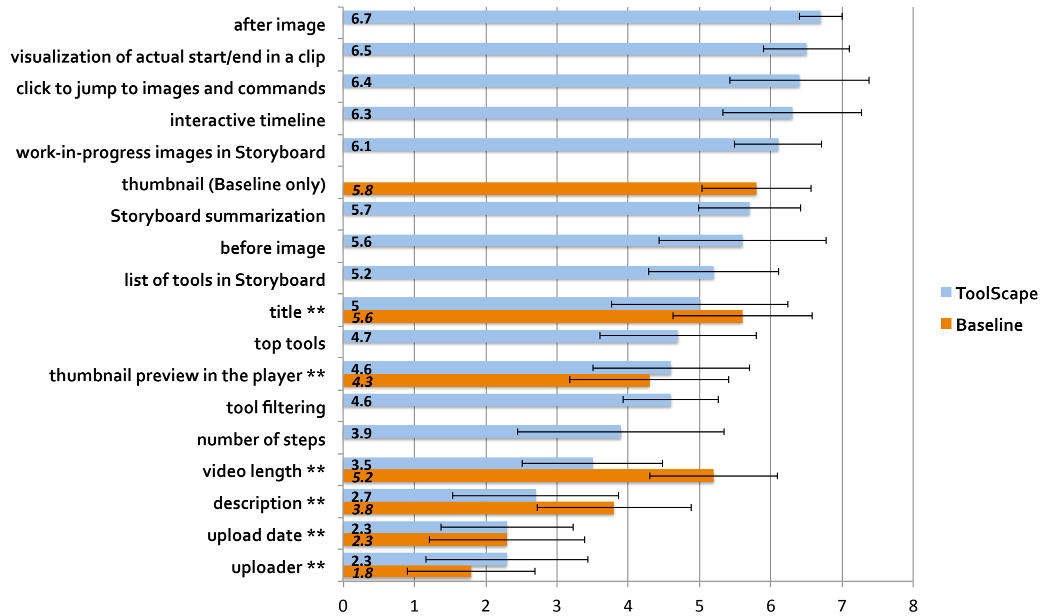


Fig. 18. Features from each interface the participants rated to be helpful, sorted by score. ** next to a label means the feature exists in both conditions. Users rated on a 1 (strongly disagree) - 7 (strongly agree) scale question whether a feature was helpful in their task. Bars indicate 95% confidence intervals. ToolScape features were highly favored across users.

We additionally looked at task completion time, but we observed no significant difference between interfaces. There are conflicting factors in play: the ability to skip unnecessary parts and higher accuracy in finding a specific moment in the video in ToolScape might shorten the task time, but users tend to navigate, interact with, and investigate workflows more with ToolScape at the same time.

In conclusion, ToolScape had a positive effect on learners' belief in their graphical design skills and quality of the outcome. More importantly, they were able to produce better designs when rated by external judges. This is a strong result considering the fact that the participants were watching the same video content in both conditions. The interface for browsing and watching the videos affected the design outcome. Most of the participants' comments were positive, especially about being able to freely navigate between steps within a video. Such positive experience, made possible by higher interactivity and control, might have led them to have increased self-efficacy, which in turn might have promoted more successful application of Photoshop skills in the design tasks.

9. STUDY 2 RESULTS: INTERFACE USABILITY

Learners' responses to the 7-point Likert scale questions on interface usability (Fig. 18) suggest that the main features of ToolScape contributed substantially to user's satisfaction and subjective ratings of success. The interactive time visualization (6.4) and tool/image links (6.3) were highly rated, as well as the graying out of parts with no visual progress (6.5). This suggests that non-sequential and stepwise access, navigation, and interactivity were perceived as important.

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."

It is interesting to note how title, video length, and description, which were available in both interfaces, were rated much lower in the ToolScape condition than in the Baseline condition. Video length, especially, was rated significantly lower in ToolScape (3.5) than in Baseline (5.2). Mann-Whitney's U test shows a significant effect of interface ($Z=-2.6028$, $p<0.01$). The reason might be that participants using ToolScape had more visual and direct cues to rely on for relevance evaluation than video length. We believe that ToolScape was successful in offering better information scent for video tutorials with information that was more directly tailored to the learners' needs.

9.1. The browsing interface

Top tools (4.7), tool filtering (4.6), and the number of steps (3.9) were lowest rated features among those available only in ToolScape. The result is not surprising because our database displayed only 10 videos at once, and the benefit of using top tools or filter results was not substantial. Top tools based simply on frequency is problematic because in many cases top-ranked tools include generic ones frequently used in other tasks as well, such as *New Layer* or *Duplicate Layer*. In the future with a larger corpus of annotated videos, we plan to apply an algorithm like TF-IDF to emphasize tools unique to the current task.

Two participants mentioned that having two view modes is confusing. Although after image (6.7) was the highest rated feature, it was not accessible if a Storyboard required wider space than the screen width. The results suggest that the final product should be always visible.

9.2. The watching interface

Features providing higher interactivity and non-sequential access were both highly rated and frequently used. All participants actively used the ability to navigate directly to specific images and commands. They clicked interactive image and command links 8.9 times on average ($\sigma = 6.7$) in a single task.

The number of non-sequential access events in the video player provides a measure of how linear the watching pattern was in each interface condition. These events are triggered 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, while in Baseline there were 96 in the twelve study sessions. Out of the 150 events in ToolScape, 107 were interactive link clicks and 43 were pause button clicks or direct scrubbing on the player. This indicates that interactive links largely replaced the needs for pause or scrubbing and encouraged the stepwise navigation of the workflow.

There was no notable difference in the number of videos open between the interfaces. Users in the ToolScape condition opened 2.3 ($\sigma = 1.4$) videos and those with Baseline opened 2.1 ($\sigma = 1.4$).

Several users were concerned with information overload in the interactive timeline. The timeline generator places a thumbnail slightly below or above the preceding one when occlusion is detected, but this is not enough when there are multiple adjacent short steps. Future iterations will strengthen favored features and address user concerns.

10. DISCUSSION

The findings from Study 2 suggest that the learners performed better, believed they performed better, and gained more confidence in their skills when they used ToolScape in image manipulation tasks. It is worth noting that the learners in both conditions used the same resources. It was the way they browsed, navigated, watched, and interacted with video tutorials that affected the outcome. This section attempts to provide plausible explanations for how ToolScape helped learners in their design tasks. We also discuss the limitations of our studies and outline design lessons from the design experience of ToolScape.

10.1. How ToolScape Helps Learners

While there were individual differences in how the learners locate, learn, and apply the skills from video tutorials, we noticed certain patterns in the learning process. The patterns differed evidently depending on where they were in the learning process.

10.1.1. Choosing the first video. For the most part, the learning cycle started with reviewing multiple candidate tutorials from the results page in ToolScape. When deciding which video to watch, learners relied on workflow information such as steps involved and intermediate images as well as other metadata. While ToolScape's filtering option enables narrowing down the search space by including or excluding specific tools, the learners engaged with the task for the first time were reluctant to use the filters. This is probably because it is uncertain which tools to filter as they do not have the right mental model yet to initiate fine-grained search. After scanning a few Storyboard summaries from the list, the learners selected their first video to watch.

When watching the first tutorial on a topic, many learners chose to watch linearly from the beginning. For videos with a long introduction, some skipped to the first step by clicking on the interactive link on the timeline. Consistent with observations from previous research [Grabler et al. 2009; Pongnumkul et al. 2011], the learners often missed or did not understand some steps in their first attempt because pacing was not personalized. This is when they turned to interactive links again: many clicked on the closest previous tool or image link to replay the step until they understood it.

While some learners watched the tutorial to the end to study all the steps involved in accomplishing the desired effect, others started applying the steps to their image while watching the tutorial. This watch-and-apply loop repeated for each step for those who chose to follow along the tutorial. These learners frequently clicked on interactive links as well, because they processed the workflow as discrete steps.

10.1.2. Watching more videos until satisfied. Six of the twelve learners completed their task using ToolScape with only a single video. The rest watched three to six videos. The number of videos watched did not correlate with time to completion. The learners decided to watch more videos for various reasons: the learner wanted to inspect more examples, improve the image further by combining techniques from multiple videos, or see alternative approaches to a step they found difficult to follow, etc. These learners came back to the ToolScape browsing interface and restarted scanning and reviewing. This time, however, the filtering feature in ToolScape was more helpful because they had a clearer search goal.

When watching a video, many learners modified their design *while* watching a video tutorial. The changes had become more fine-grained as they watched more videos. The demand for precise positioning of the video playhead was higher, because these learners wanted to pick and apply a specific skill from a clip. Some learners mixed techniques from multiple videos to improve the output quality.

10.2. Design Implications for Video Tutorial Interfaces

We revisit the major design decisions in ToolScape to extract design implications. There has been little research investigating how interaction techniques based on content-specific annotations affect learning. This section intends to encourage designers of video browsing interfaces to consider the design suggestions supported by experimental results from our studies.

10.2.1. Provide interactive, non-sequential ways to navigate videos. The participants in our study used multiple methods to watch videos in a non-sequential manner. The specific design choices to visualize non-essential parts, include before/after images for *each step*, and label steps with full text, not just markers, were all highly favored and frequently used. Allowing users to control the flow and pace seems to yield higher user satisfaction and self-efficacy gain.

The design insight of providing more learner control and interactivity proved to be effective. Higher interactivity with the material can encourage learners to reinspect [Tversky et al. 2002], which might have led learners to understand the workflow better. More learner control increases feelings of competence, self-determination, intrinsic interest [Lepper and Chabay 1985], which in turn might have led to improved performance as well as increased self-efficacy in ToolScape.

10.2.2. Step-by-step access to the workflow should be available. A distinctive attribute of ToolScape is the use of step-by-step, content-based annotations, specifically a visualization of tools used and intermediate results. Our study demonstrates that learners frequently navigated how-to videos non-sequentially, actively skipping, replaying, and jumping between steps. Many existing tutorial enhancement systems leverage the internal application context and attempt to fully recover the workflow. While the annotations collected by ToolScape cannot completely replicate the workflow because they omit some details such as fine-grained parameter tuning, we argue that step-by-step information can yield learning gains without the cost of requiring programmable in-app access.

10.2.3. Visualize both global and local changes in the workflow. Visual comparison is important both when scanning multiple workflows and when reviewing adjacent steps within a workflow. Being able to contrast before and after images and track visual progress is highly favored by users. Because workflows often involve multiple complex steps, local changes in addition to the initial and final artifact should be visualized and accessible. Also, displaying an end product all the time helps users keep track of their goal, which essentially makes ToolScape's interactive timeline a usable worked example.

10.3. Limitations of the Studies

We limited the scope of our study to control various factors that might affect the participants' learning experience. Here we discuss three of the limitations of the study configuration, and suggest what needs to be done to generalize the findings.

10.3.1. Novice Users. One limitation of the study is that all the participants were novices with limited experience in Photoshop. We suspect that experts will exhibit different learning patterns, although we believe that they will still benefit from better video summaries and interactive timelines. This work focused on novices because its original goal was to help novices master new skills. Future work will look at how experts use the system and compare how their usage pattern differs from that of novice learners.

10.3.2. Limited Database. Another limitation is that the user study used highly accurate annotations manually generated by researchers. Because collecting annotations with scalability, efficiency, and accuracy can significantly improve the impact of this work, our current work focuses on exploring and designing better annotation methods. We will collect and compare annotations from the automatic, crowdsourcing, and learnersourcing methods. Our initial results suggest that mixing multiple methods to complement each other can be a viable solution, because no one method yields a perfect result.

Also, our database was too small to simulate real-life search sessions where learners often encounter hundreds of results. The features of displaying most frequently used tools and filtering have not been heavily used. With a small database, users are not likely to benefit from aggregate information. We are currently building a database of thousands of annotated videos, which will raise interesting challenges for visualizing, sorting, and filtering multiple workflows.

10.3.3. Retention and Transfer. Our user study measured what Eiriksdottir and Catrambone [2011] categorize as initial performance, which measures how a learner performs a task highly similar to the training tasks (tutorials) while using the instructions. While further studies need to be conducted to see if the skill acquired is retained or is transferrable to a new situation, we believe that providing high interactivity and easily scannable summaries for how-to videos will lead to better retention and transfer.

11. CONCLUSIONS AND FUTURE WORK

This paper explores summarization methods and interaction techniques video learning interfaces can use to support learners in acquiring new skills from how-to videos. We introduce ToolScape, a video browsing and watching interface that includes the Storyboard summarization to support finding relevant videos and the interactive timeline to provide step-by-step navigation for learners. We conducted a two-part laboratory study to address the effects of ToolScape on browsing and watching how-to videos.In a comparative study with two other summarization techniques, we found that Storyboard shortens time to scan a workflow, without negatively affecting the relevance estimations. In a laboratory study that asked participants to apply new Photoshop design skills, participants using ToolScape rated their own work higher, gained more self-efficacy, and produced higher-rated designs than when using the baseline interface with no enhancements. We attribute the success to providing more learner control and interactivity with the procedural workflow.

Our immediate future work is to increase the size of the annotation database. To address the scalability problem, we plan to investigate automatic labeling methods as well as crowdsourcing, labeling game interfaces, and collecting annotations as a by-product of learner activities.

Also, the idea of using step-by-step annotations in a video interface can expand to other procedural task domains. We will explore domains such as makeup, cooking, or knitting to understand what knowledge can be transferred from the current work.

APPENDIX

In this section we present the questionnaires used in the user study.

A.1. Post-questionnaire for Study 1

We asked the participants the following questions after completing their tasks in Study 1. In the scale of 1 (strongly disagree) to 7 (strongly agree), these questions are designed to compare the usability and satisfaction of the three summarization methods.

— Q1. It was easy to use.
— Q2. It was easy to understand.
— Q3. It was enjoyable.
— Q4. It was informative.
— Q5. It represented the video well.
— Q6. It aided in deciding whether the video is relevant.
— Q7. It aided in deciding whether the video contains new skills to learn.
— Q8. It aided in deciding whether the video is advanced.
— Q9. It replaced watching the full video.
— Q10. It would be useful in browsing video tutorials.

A.2. Post-questionnaire for Study 2

We asked the participants the following questions after completing their design tasks in Study 2.

In the scale of 1 (strongly disagree) to 7 (strongly agree), the following two questions are self-assessment of the task and the output quality. The results of these questions are used to verify the hypotheses in Study 2.

— Q1. The task was difficult to perform. (task difficulty)
— Q2. The quality of my final image is high. (self-rating)

In the scale of 1 (strongly disagree) to 7 (strongly agree), these questions are designed to measure the user satisfaction of the video interface.

— Q1. It was easy to use.
— Q2. It was easy to understand.
— Q3. It was enjoyable.
— Q4. It was informative.

ACKNOWLEDGMENTS

The authors would like to thank Joel Brandt and Mira Dontcheva for insightful discussions. This work is supported in part by Adobe, and by Quanta Computer as part of the T-Party project. Juho Kim is supported by the Samsung Fellowship. Opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

Amal Arguel and Eric Jamet. 2009. Using video and static pictures to improve learning of procedural contents. *Computers in Human Behavior* 25, 2 (2009), 354 – 359. DOI:http://dx.doi.org/10.1016/j.chb.2008.12.014

Robert K. Atkinson, Sharon J. Derry, Alexander Renkl, and Donald Wortham. 2000. Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research* 70, 2 (2000), pp. 181–214. http://www.jstor.org/stable/1170661

Panos Balatsoukas, Anne Morris, and Ann O'Brien. 2009. An evaluation framework of user interaction with metadata surrogates. *J. Inf. Sci.* 35, 3 (June 2009), 321–339. DOI:http://dx.doi.org/10.1177/0165551508099090

A. Bandura. 1977. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review* 84, 2 (1977), 191.

Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. ACM, New York, NY, USA. DOI:http://dx.doi.org/10.1145/2208516.2208549

Peter A. Bibby and Stephen J. Payne. 1993. Internalization and the use specificity of device knowledge. *Hum.-Comput. Interact.* 8, 1 (March 1993), 25–56. DOI:http://dx.doi.org/10.1207/s15327051hci0801_2

Richard Catrambone. 1996. Generalizing solution procedures learned from examples. *Journal of Experimental Psychology-learning Memory and Cognition* 22 (1996), 1020–1031. Issue 4. DOI:http://dx.doi.org/10.1037//0278-7393.22.4.1020

Kai-Yin Cheng, Sheng-Jie Luo, Bing-Yu Chen, and Hao-Hua Chu. 2009. SmartPlayer: user-centric video fast-forwarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 789–798. DOI:http://dx.doi.org/10.1145/1518701.1518823

Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Bjorn Hartmann. 2012. MixT: Automatic Generation of Step-by-Step Mixed Media Tutorial. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. ACM, New York, NY, USA. DOI:http://dx.doi.org/10.1145/2208516.2208549

Michael G. Christel, Michael A. Smith, C. Roy Taylor, and David B. Winkler. 1998. Evolving video skims into useful multimedia abstractions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 171–178. DOI:http://dx.doi.org/10.1145/274644.274670

R.C. Clark and R.E. Mayer. 2007. *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning.* John Wiley & Sons. http://books.google.com/books?id=MOutGGET2VwC

Ruth C Clark, Frank Nguyen, and John Sweller. 2005. *Efficiency in learning: Evidence-based guidelines to manage cognitive load*. Pfeiffer.

Morgan Dixon and James Fogarty. 2010. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of the 28th international conference on Human factors in computing systems (CHI '10)*. ACM, New York, NY, USA, 1525–1534. DOI:http://dx.doi.org/10.1145/1753326.1753554

Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 18 (Dec. 2010), 24 pages. DOI:http://dx.doi.org/10.1145/1879831.1879836

E. Eiriksdottir and R. Catrambone. 2011. 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.

Erika L Ferguson and Mary Hegarty. 1995. Learning with real machines or diagrams: application of knowledge to real-world problems. *Cognition and Instruction* 13, 1 (1995), 129–160.

Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-sketch revolution: an engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM, New York, NY, USA, 373–382. DOI:http://dx.doi.org/10.1145/2047196.2047245

A. Girgensohn, J. Boreczky, and L. Wilcox. 2001. Keyframe-based user interfaces for digital video. *Computer* 34, 9 (2001), 61–67. DOI:http://dx.doi.org/10.1109/2.947093

Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. In *SIGGRAPH '09*. ACM, New York, NY, USA, 1–9. DOI:http://dx.doi.org/10.1145/1576246.1531372

Tovi Grossman and George Fitzmaurice. 2010. ToolClips: an investigation of contextual video assistance for functionality understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1515–1524. DOI:http://dx.doi.org/10.1145/1753326.1753552

Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: capture, exploration, and playback of document workflow histories. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology (UIST '10)*. ACM, New York, NY, USA, 143–152. DOI:http://dx.doi.org/10.1145/1866029.1866054

Rassule Hadidi and Chung-Hsien Sung. 1998. Students' acceptance of Web-based course offerings: an empirical assessment. (1998).

Susan M Harrison. 1995. A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 82–89.

Tim N Höffler and Detlev Leutner. 2007. Instructional animation versus static pictures: A meta-analysis. *Learning and instruction* 17, 6 (2007), 722–738.

Jeffrey D Karpicke and Janell R Blunt. 2011. Retrieval practice produces more learning than elaborative studying with concept mapping. *Science* 331, 6018 (2011), 772–775.

Caitlin Kelleher and Randy Pausch. 2005. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 541–550.

Juho Kim, Robert C. Miller, and Krzysztof Z. Gajos. 2013. Learnersourcing Subgoal Labeling to Support Learning from How-to Videos. In *CHI 2013 Extended Abstracts, to appear*. ACM, New York, NY, USA.

Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. 2012. Delta: a tool for representing and comparing workflows. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1027–1036. DOI:http://dx.doi.org/10.1145/2208516.2208549

Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems (CHI '13)*. ACM, New York, NY, USA, 1779–1788. DOI:http://dx.doi.org/10.1145/2466110.2466235

Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. 2011. ShadowDraw: real-time user guidance for freehand drawing. *ACM Trans. Graph.* 30, 4, Article 27 (July 2011), 10 pages. DOI:http://dx.doi.org/10.1145/2010324.1964922

Jo-Anne LeFevre. 1987. Processing instructional texts and examples. *Canadian Journal of Psychology/Revue canadienne de psychologie* 41, 3 (1987), 351.

Mark R Lepper and Ruth W Chabay. 1985. Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. *Educational Psychologist* 20, 4 (1985), 217–230.

Gary Marchionini, Yaxiao Song, and Robert Farrell. 2009. Multimedia surrogates for video gisting: Toward combining spoken words and imagery. *Inf. Process. Manage.* 45, 6 (Nov. 2009), 615–630. DOI:http://dx.doi.org/10.1016/j.ipm.2009.05.007

Phu Nguyen, Juho Kim, and Robert C. Miller. 2013. Generating Annotations for How-to Videos Using Crowdsourcing. In *CHI 2013 Extended Abstracts, to appear*. ACM, New York, NY, USA.

Jian-Quan Ouyang, Jin-Tao Li, and Yong-Dong Zhang. 2003. Replay boundary detection in MPEG compressed video. In *Machine Learning and Cybernetics, 2003 International Conference on*, Vol. 5. 2800 – 2804 Vol.5. DOI:http://dx.doi.org/10.1109/ICMLC.2003.1260028

Susan Palmiter and Jay Elkerton. 1991. An evaluation of animated demonstrations of learning computer-based tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. ACM, 257–263.

Susan Palmiter, Jay Elkerton, and Patricia Baggett. 1991. Animated demonstrations¡ i¿ vs¡/i¿ written instructions for learning procedural tasks: a preliminary investigation. *International Journal of Man-Machine Studies* 34, 5 (1991), 687–701.

Sunghyun Park, Gelareh Mohammadi, Ron Artstein, and Louis-Philippe Morency. 2012. Crowdsourcing Micro-Level Multimedia Annotations: The Challenges of Evaluation and Interface. In *Proceedings of International ACM Workshop on Crowdsourcing for Multimedia*.

Peter Pirolli. 1991. Effects of examples and their explanations in a lesson n recursion: A production system analysis. *Cognition and Instruction* 8, 3 (1991), 207–259.

Peter Pirolli and Stuart K. Card. 1999. Information Foraging. *Psychological Review* 106 (1999), 643–675.

Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael Cohen. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. In *UIST 2011*. ACM, New York, NY, USA.

Alexander Renkl and Robert K Atkinson. 2003. Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational psychologist* 38, 1 (2003), 15–22.

D.H. Schunk. 1990. Goal setting and self-efficacy during self-regulated learning. *Educational psychologist* 25, 1 (1990), 71–86.

Yaxiao Song, Gary Marchionini, and Chi Young Oh. 2010. What are the most eye-catching and ear-catching features in the video?: implications for video summarization. In *Proceedings of the 19th international conference on World wide web (WWW '10)*. ACM, New York, NY, USA, 911–920. DOI:http://dx.doi.org/10.1145/1772690.1772783

C.M. Taskiran, Z. Pizlo, A. Amir, D. Ponceleon, and E.J. Delp. 2006. Automated video program summarization using speech transcripts. *Multimedia, IEEE Transactions on* 8, 4 (aug. 2006), 775 –791. DOI:http://dx.doi.org/10.1109/TMM.2006.876282

Jaime Teevan, Edward Cutrell, Danyel Fisher, Steven M. Drucker, Gonzalo Ramos, Paul André, and Chang Hu. 2009. Visual snippets: summarizing web pages for search and revisitation. In *CHI '09*. ACM, New York, NY, USA, 2023–2032. DOI:http://dx.doi.org/10.1145/1518701.1519008

Ba Tu Truong and Svetha Venkatesh. 2007. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 3, 1, Article 3 (Feb. 2007). DOI:http://dx.doi.org/10.1145/1198302.1198305

Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. 2002. Animation: can it facilitate? *International journal of human-computer studies* 57, 4 (2002), 247–262.

Hans van der Meij, Peter Blijleven, and Leanne Jansen. 2003. *What makes up a procedure?* Lawrence Erlbaum Mahwah, NJ.

Stina Westman. 2010. Evaluation constructs for visual video summaries. In *Proceedings of the 14th European conference on Research and advanced technology for digital libraries (ECDL'10)*. Springer-Verlag, Berlin, Heidelberg, 67–79. http://dl.acm.org/citation.cfm?id=1887759.1887772

Susan Wiedenbeck. 1989. Learning iteration and recursion from examples. *International Journal of Man-Machine Studies* 30, 1 (1989), 1 – 22. DOI:http://dx.doi.org/10.1016/S0020-7373(89)80018-5

Barbara M. Wildemuth, Gary Marchionini, Meng Yang, Gary Geisler, Todd Wilkens, Anthony Hughes, and Richard Gruss. 2003. How fast is too fast?: evaluating fast forward surrogates for digital video. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (JCDL '03)*. IEEE Computer Society, Washington, DC, USA, 221–230. http://dl.acm.org/citation.cfm?id=827140.827176

Tom Yeh, Tsung-Hsiang Chang, and Robert C. Miller. 2009. Sikuli: using GUI screenshots for search and automation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09)*. ACM, New York, NY, USA, 183–192. DOI:http://dx.doi.org/10.1145/1622176.1622213

Dongsong Zhang, Lina Zhou, Robert O. Briggs, and Jay F. Nunamaker Jr. 2006. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management* 43, 1 (2006), 15 – 27. DOI:http://dx.doi.org/10.1016/j.im.2005.01.004

Barry J. Zimmerman, Albert Bandura, and Manuel Martinez-Pons. 1992. Self-Motivation for Academic Attainment: The Role of Self-Efficacy Beliefs and Personal Goal Setting. *American Educational Research Journal* 29, 3 (1992), 663–676. DOI:http://dx.doi.org/10.3102/00028312029003663

Received May 2013; revised ; accepted