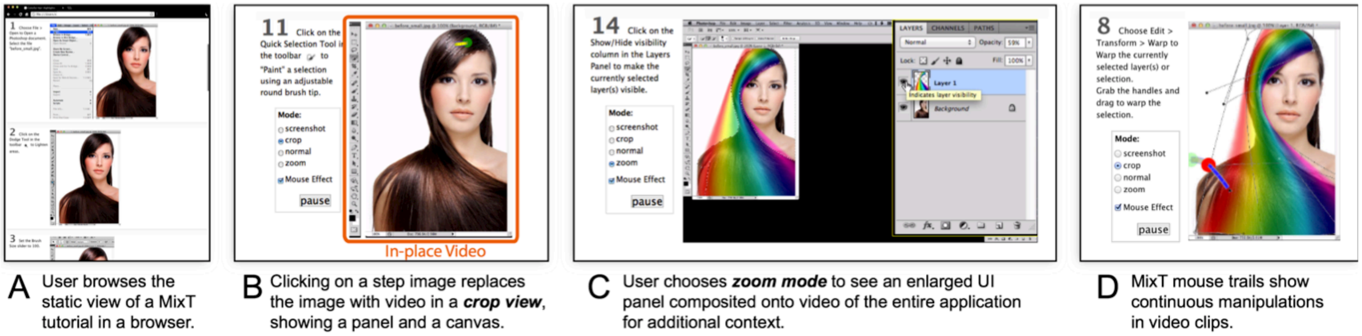
MixT: Automatic Generation of Step-by-Step Mixed Media Tutorials

Pei-Yu (Peggy) Chi[1], **Sally Ahn**[1], **Amanda Ren**[1], **Mira Dontcheva**[2], **Wilmot Li**[2], **Björn Hartmann**[1]

[1]University of California, Berkeley — Computer Science Division
{peggychi, sallyahn, aren, bjoern}@berkeley.edu

[2]Advanced Technology Labs, Adobe
{mirad, wilmotli}@adobe.com



1 Choose File > Open to Open a Photoshop document. Select the file "before_small.jpg".

2 Click on the Dodge Tool in the toolbar to Lighten areas.

3 Set the Brush Size slider to 100.

11 Click on the Quick Selection Tool in the toolbar to "Paint" a selection using an adjustable round brush tip.

14 Click on the Show/Hide visibility column in the Layers Panel to make the currently selected layer(s) visible.

8 Choose Edit > Transform > Warp to Warp the currently selected layer(s) or selection. Grab the handles and drag to warp the selection.

Mode: ○ screenshot ● crop ○ normal ○ zoom ☑ Mouse Effect (panels B and D)

Mode: ○ screenshot ○ crop ○ normal ● zoom ☑ Mouse Effect (panel C)

A User browses the static view of a MixT tutorial in a browser.

B Clicking on a step image replaces the image with video in a ***crop view***, showing a panel and a canvas.

C User chooses ***zoom mode*** to see an enlarged UI panel composited onto video of the entire application for additional context.

D MixT mouse trails show continuous manipulations in video clips.

Figure 1: MixT generates tutorials that contain static and video information from task demonstrations. Videos are automatically edited and offer different views to highlight the most relevant screen areas for a step. Visualizing mouse movement helps user understand a complex action.

ABSTRACT

Users of complex software applications often learn concepts and skills through step-by-step tutorials. Today, these tutorials are published in two dominant forms: *static tutorials* composed of images and text that are easy to scan, but cannot effectively describe dynamic interactions; and *video tutorials* that show all manipulations in detail, but are hard to navigate. We hypothesize that a mixed tutorial with static instructions and per-step videos can combine the benefits of both formats. We describe a comparative study of static, video, and mixed image manipulation tutorials with 12 participants and distill design guidelines for mixed tutorials. We present MixT, a system that automatically generates step-by-step mixed media tutorials from user demonstrations. MixT segments screencapture video into steps using logs of application commands and input events, applies video compositing techniques to focus on salient information, and highlights interactions through mouse trails. Informal evaluation suggests that automatically generated mixed media tutorials were as effective in helping users complete tasks as tutorials that were created manually.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Author Keywords: Software tutorials; instructions; video

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST '12, October 7–10, 2012, Cambridge, Massachusetts, USA.
Copyright 2012 ACM 978-1-4503-1580-7/12/10...$15.00.

INTRODUCTION

Learning how to use software applications often happens opportunistically as users need to accomplish specific tasks. When it is unclear how to achieve the desired results, many users turn to step-by-step tutorials, which describe the set of operations required to complete a task. Visual editing applications, such as applications for drawing, photo editing, and 3D modelling, require visual tutorials that show not only how to navigate the user interface but also how to manipulate the canvas, image, or 3D model.

There are two main forms of visual step-by-step tutorials. *Static tutorials* use text and images to describe the set of operations required to accomplish a task. *Video tutorials* are screen recordings of the tutorial author performing the the task. Both forms of instructional content have strengths and weaknesses. Static tutorials are easy to scan forward and backward because they show all instructions. Offering both text and images, they are well suited for people who prefer to learn by looking at images and those who prefer to learn by reading text [11]. However, it can be difficult for users to understand continuous, complex manipulations such as painting a region, adjusting control points, or rotating a 3D object in static tutorials. In contrast, videos are effective at showing exactly how an application responds to user interaction, but it is hard to navigate back to previous steps or to look ahead in a video timeline [18].

We hypothesize that a combination of static and video instructions can improve users' success in following tutorials. We focus on image-editing software in particular, because it is widely used and has a large collection of tutorials accessible in bookstores (books and magazines) and on the web (e.g., user forums and video sharing sites), but we suspect that our findings are generally applicable to visual editing software. With mixed static and video tutorials, users may effectively learn complicated actions (e.g., applying brush strokes) from tutorial video clips, and quickly access basic actions (e.g., copying a layer) from static text and images.

To test our hypothesis, we carried out a within-subjects study comparing static, video, and mixed media tutorials. 12 participants completed three workflows, one for each format. We found that videos are especially valuable for actions that involve brushing, control point manipulation, and adjustment of continuous parameters. We also found that the availability of video reduces the number of repeated attempts users make to execute a step. The study results led to four design guidelines for mixed media tutorials: 1) offer a scannable overview of steps; 2) include small but legible videos; 3) add visualizations of canvas interactions such as brushing to the videos, and 4) enable users to choose the most appropriate visual representation for each step.

To enable instructors to create mixed media tutorials, we introduce MixT, a system that takes a user demonstration and automatically generates mixed tutorials that show static step-by-step content and also include in-place video clips for each operation (Figure 1). MixT generates these materials from screencapture video and recorded traces of application commands and input device events. MixT segments video into steps, applies video compositing techniques to focus on salient screen regions, and highlights canvas interactions through mouse trails. The web-based tutorials give users interactive control over when to see images or videos, and how to render videos. A quantitative analysis of nine automatically generated MixT tutorials indicates that our algorithms for segmenting videos into steps and detecting salient regions within frames are effective (<8% error rates). In addition, informal user feedback suggests that MixT tutorials were as effective as manually created tutorials in helping users complete tasks.

In summary, the main contributions of this paper include:

- a categorization of the types of user operations for which video is useful.
- a set of design principles for how to embed video in step-by-step tutorials, derived from a formative study.
- a general approach for automatically generating mixed media tutorials from demonstrations, and algorithms for implementing this approach for Adobe Photoshop.
- an evaluation of automatically generated mixed media tutorials.

RELATED WORK

Previous HCI research on instructional content falls into four main categories: 1) new tutorial formats and interfaces [1,6,9,12,14,15,18]; 2) automated methods for generating tutorials [3,8,10,18]; 3) studies evaluating the effectiveness of different instructional formats [8,9,11,16,17]; and 4) techniques for searching and analyzing collections of tutorials [5,13]. Our work on generating and evaluating mixed-media tutorials addresses the first three of these topics. In this section, we describe related work on new tutorials formats and automatic generation of instructional content. The following section discusses previous studies on tutorial effectiveness in the context of our own formative study.

New forms of instructional content: While static step-by-step and video tutorials are the most prevalent forms of instructional content, researchers have been exploring new formats and interfaces for learning materials. Many efforts propose instructional aids that work in conjunction with the target application, including in-application step-by-step wizards [1,12,6] and Q&A forums [15], video-based tooltips [9], interactive video tutorials [18], command recommenders [14], and interface facades for mapping commands between applications [19]. Some systems also include features that facilitate navigation within tutorials, such as annotated video timelines [10,18] and reactive "current step" indicators [6]. Our work introduces an interactive mixed-media tutorial format that combines aspects of both step-by-step and video-based learning aids.

Automatic generation of learning materials: One of our key contributions is a method for automatically generating mixed-media tutorials from user demonstrations. Most existing tutorial generation approaches analyze traces of user interactions through the application or screencast video of the demonstration. Methods that analyze user action traces [8,3,10] often capture application-level semantics about specific tools and their purpose, while techniques that analyze screen recordings [18,20,4] use computer vision to identify GUI interactions such as clicking on an icon or button. In MixT, we combine and extend these approaches to create step-by-step tutorials that incorporate text, images, and several formats of video.

FORMATIVE USER STUDY

Researchers provide different findings on the effectiveness of media formats of software tutorials. Evaluating the instructional potential of videos began in the 1990s. Palmiter, Elkerton [16,17], and Harrison [11] studied the effect of animated demonstrations on learning and instruction recall. More recently, Grabler *et al.* compared how users followed book tutorials, videos, and automatically generated static tutorials [8]. Their results showed that automatically generated text and image tutorials outperformed video or book instructions on time and errors. Grossman *et al.* studied the effectiveness of embedding short (10-25 second) video clips in applications [9]. They found that participants who had access to video-based tooltips were significantly faster in completing tasks than those who viewed static ones.

While these studies suggest that there is still some debate over the tradeoffs between step-by-step static and video tutorials, they provide strong support for two key claims: step-by-step tutorials help users make fewer errors by allowing them to work at their own pace, while videos can help provide subtle details of complex interactions that are difficult to represent statically. Based on these findings, we

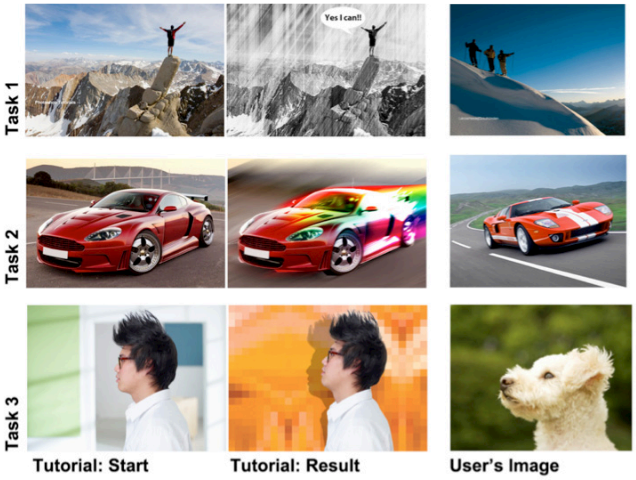
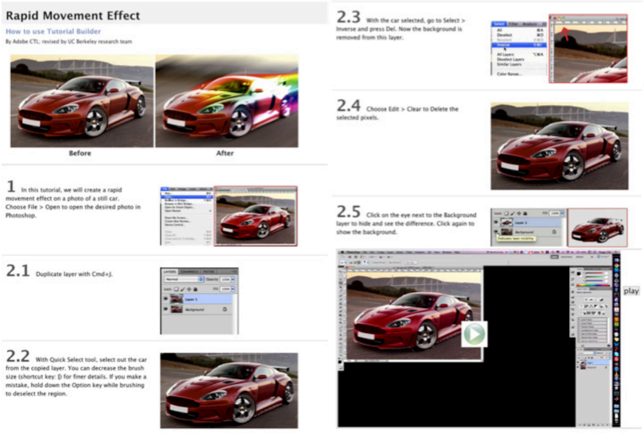


Figure 2: In our formative study, participants completed three tutorials with images similar but not identical to the originals.



Rapid Movement Effect

How to use Tutorial Builder

By Adobe CTL; revised by UC Berkeley research team

1 In this tutorial, we will create a rapid movement effect on a photo of a still car. Choose File > Open to open the desired photo in Photoshop.

2.1 Duplicate layer with Cmd+J.

2.2 With Quick Select tool, select out the car from the copied layer. You can decrease the brush size (shortcut key: [) for finer details. If you make a mistake, hold down the Option key while brushing to deselect the region.

2.3 With the car selected, go to Select > Inverse and press Del. Now the background is removed from this layer.

2.4 Choose Edit > Clear to Delete the selected pixels.

2.5 Click on the eye next to the Background layer to hide and see the difference. Click again to show the background.

Figure 3: In the mixed condition, participants saw an HTML page with static images and text; they could expand each step to view a video of that step (here: step 2.5).

designed a formative user study that investigates whether video clips can be incorporated into a step-by-step framework to help users follow certain types of image-editing tasks within a tutorial.

Study Design

Hypotheses. Our formative study aims to test the following two hypotheses:

H1 Image manipulation tutorials that mix static images and video clips are more effective than all-static or all-video tutorials.

H2 Users benefit more from seeing video clips instead of static text and images for certain types of commands.

Participants. We recruited 12 participants (5 males and 7 females, aged 20-52), 4 from a campus student design group and 8 from a computer software company, and compensated each with a $15 gift card for participating. Our tutorials focused on achieving specific tasks in Adobe Photoshop. We recruited participants who had prior expertise with Adobe Photoshop, but who were not expert users. To demonstrate expertise, potential participants first completed an online screening test that asked them to follow a short image manipulation tutorial and submit the resulting file. The selected participants had between 1 and 20 years of experience using Photoshop.

Tasks and Material. The study was based on a within-subject design. We looked through Photoshop books and selected 3 different image manipulation tasks with similar levels of difficulty and complexity (see Figure 2). Each tutorial comprised 15-20 steps. We focused on tutorials that included new, less common features such as the *liquify* tool, *gradient warp* tool, and *puppet warp* tool to increase the chance that participants would encounter unfamiliar tools. For each tutorial, we created three types of presentations: 1) *static* (in HTML format displayed on the screen), 2) *video* (on YouTube with audio narration), and 3) *mixed* (web interface shown in Figure 3 without audio). To ensure that different formats presented equivalent information where possible, we first recorded and narrated our video tutorials, then manually generated the static version by writing text instructions based on the narration and annotating and cropping frames of the video. To create mixed tutorials we started with the static tutorials and added the corresponding screencapture video segment for each step. To view the video segment for a step in the mixed tutorial, the user had to click on the image for that step. We scaled these videos to a fixed resolution of 800x500 pixels so that at least 2-3 steps would fit on screen when the videos were expanded. Many online tutorials do not offer full-screen resolution videos; even when high-resolution videos are available, they are hard to use as they force users to continually switch between the video and application windows. We disabled the soundtrack in the mixed tutorial to avoid situations when users only relied on auditory instructions instead of learning from static or video formats.

For each task, participants were given a source image that was distinct, but thematically similar to the image manipulated in the tutorial itself. This study design choice was motivated by the fact that users typically want to transfer the techniques found in tutorials to their own images.

Procedure and Environment. Each session consisted of 1 warm-up task and 3 experimental tasks. The warm-up task was a short 5-step static tutorial. In the 3 experimental tasks the format and task order were randomized. Each 60-minute session was conducted in a lab environment, using computers running Mac OS X, Adobe Photoshop CS5.1 and a web browser (Google Chrome) for viewing tutorials. Each participant was provided with a keyboard and a mouse and was allowed to adjust the equipment setting such as the monitor position and mouse tracking speed during the warm-up task. Photoshop and the web browser were arranged side-by-side on a 30-inch monitor with a resolution of 2048x1280 pixels. During the study, we used screen capture software to record user performance.

Measurement

To evaluate **H1**, we report the number of *errors* and *repeated attempts* that the participants made for each task. While our ultimate goal is skill acquisition and retention, we focus on the pragmatic goal of improving users' success in following tutorials and performing the instructions. We record an *error* if the participant performed a command incorrectly or skipped a step in the tutorial. While errors give a sense for the effectiveness of the tutorials, they do not measure the extraneous work users might have to perform when they have trouble understanding the correct outcome of a step. For example, if a user makes an error and then correctly executes several steps before recognizing the problem, we count this as a single error, even though the user must go back to fix the problem and then redo the subsequent steps. In addition, users may select the right command, but be dissatisfied with the result of their image and try again (e.g., redrawing a gradient). In such cases, we record all executions of the same step following the first attempt as a *repeated attempt*. Note that we do not count adjustments of continuous parameters or refinements of selection regions as repeated attempts because in these cases, the user is focusing on a single action rather than repeating a previously executed step. We do count a repeated attempt if the user entirely undoes a step to then retry it.

To evaluate **H2**, we count the number of different users who click on the video for each step in the mixed tutorials. To determine whether some types of commands benefit from videos more than others, we bin each step into one of the following five command categories based on the types of user interaction and UI elements it involves: brushing/drawing, manipulating control points (e.g., mesh-based warping, spline editing), parameter adjustment (e.g. using a slider to change opacity), UI navigation (e.g., switching tools, finding menu items), and layer operations.

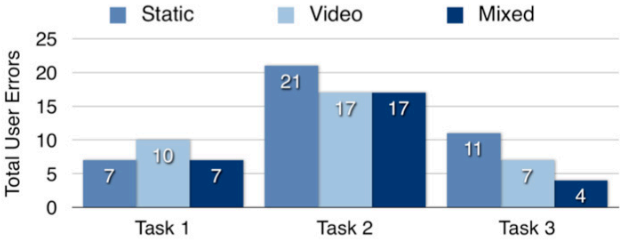


Figure 4: Users tied for fewer errors with mixed tutorials.

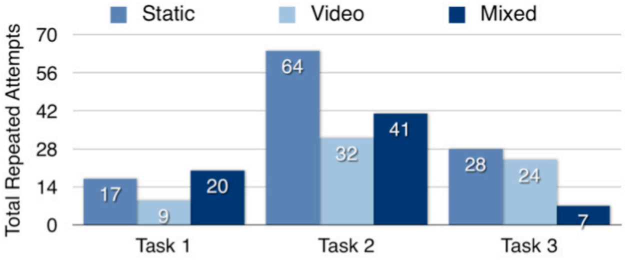


Figure 5: In two of three tasks, participants made more repeated attempts at executing steps with static tutorials than with mixed tutorials. Video tutorials had the fewest attempts.

We also collect qualitative data by observing how users follow the presented information and obtain additional feedback via 5-point Likert-scale questions (e.g., "The <condition> tutorial was easy to follow.") and open-ended questions (e.g., "Compared with static tutorials, what were the pros and cons of the mixed media tutorial?").

RESULTS OF FORMATIVE STUDY

Based on the quantitative data and observations from our study, we gained several insights about how users interact with static and video content.

User Performance on Image Editing Tasks

Our analysis of user performance supports **H1**. As Figure 4 shows, mixed tutorials resulted in the fewest total number of errors (28 for mixed, 34 for video, 39 for static) across all three tasks and produced an equivalent or fewer number of errors compared to static and video for any given task. In terms of extraneous work, the mixed condition resulted in many fewer repeated attempts than static tutorials and slightly more than video tutorials (65 for video, 68 for mixed, 109 for static, see Figure 5). Although the differences in errors and repeated attempts are not statistically significant – likely due to the small study size and differences between the tasks – the overall trends suggest that mixed tutorials help users make fewer errors and do less extraneous work compared to static and video tutorials.

In addition to these quantitative results, we observed a few specific behaviors that had an impact on user performance:

The scannable nature of the static and mixed tutorial formats helped users follow along and avoid missing steps that might result in errors. In the video condition, users were more likely to accidentally skip steps because they were working at a different pace than the video. They also had trouble finding previous steps when trying to identify the source of an error.

In the static condition, users had trouble understanding how to perform steps that involve complex or unfamiliar UI elements and interactions. As we discuss in the following subsection, these were often the same steps where users decided to play the videos in the mixed condition. With only static text and images, users often made errors or had to repeat such steps multiple times.

Participants used the video clips in the mixed condition in a few different ways. Some users played the video *before* attempting the step to familiarize themselves with the relevant UI elements and interactions. In some cases, users also played the video at the same time as they performed the action, which corresponds to what Palmiter and Elkerton described as "mimicking actions" [10]. We suspect both of these behaviors helped reduce errors, especially for complex or unfamiliar steps. In addition, several users also played the video *after* completing a step, as a way to confirm that they had performed the step correctly and "debug" what went wrong if they made an error. This confirmation behavior helped reduce repeated attempts by making it easier to recognize and fix errors sooner.

In some cases, users had trouble seeing all of the relevant details in the mixed videos because the videos were scaled down to 800x500 pixels. For example, when using the *puppet warp* tool, users missed that dragging in the vicinity of a control point (instead of on top of the control point) initiated a control wheel for a rotation rather than a translation maneuver. Although participants neither complained about not being able to resize the video in the MixT condition nor chose full-screen mode in the video condition, they explained that they would hope to clearly see the key part of the video.

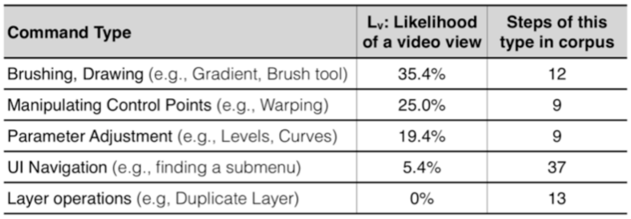
Which Commands Led to Video Views?

Our analysis of which mixed steps prompted users to click on the corresponding videos suggests that users did indeed find the video clips more beneficial for some types of steps over others (**H2**). To compute the *likelihood of a video view* (L_V) for the five command categories described earlier, we first determine L_V for each individual step (across all three mixed tutorials) by computing the fraction of users who clicked to view the video for that step, and then we average likelihoods across all the steps in each command category.

As Table 1 shows, L_V is highest for steps that involve brushing/drawing, manipulating control points and parameter adjustments. Based on our observations, videos help users perform the first two types of commands (brushing/drawing and manipulating control points) by explicitly demonstrating the necessary mouse movements rather than requiring the user to infer what to do from text and images alone. In addition, we noticed that some participants used the video clips to determine how precise they needed to be for certain brushing or selection tasks, which is hard to convey with a static representation. Text descriptions such as “rough” and “detailed” are relative and can be interpreted differently by individuals, but seeing how much time the demonstrator devotes to the task provides a much clearer estimate of the required precision for that task. As for parameter adjustments, users may be relying on the videos to provide some context about the range of visual effects the relevant parameters cover in order to determine what parameter values to use for their own image. Unlike static images, which only show the final parameter values and resulting effect, video demonstrations often show how the canvas changes as continuous parameters are updated (often via sliders), which may give users a better sense for the desired outcome of the step.

User Preferences for Tutorial Types

The results of our questionnaire show that while participants had varying opinions on the static and video tutorials, all users strongly agreed that the mixed tutorial was easy to follow. Participants had difficulty finding the tools that the static tutorials referenced and remarked that there were not enough visuals. For full-length videos, participants disliked having to pause the video to complete each step. For the mixed tutorial, half the participants found that video was the most useful among different media components for understanding a step instruction. One participant acknowledged that because the mixed tutorial allowed him to “break down the process into simple steps,” he was able to easily find the point where he had made a mistake. Another user explained that videos would be most helpful if the tasks were more advanced and in-depth. On the other hand, users had different preferences within a task. Expertise could be tool-specific: users might find static instructions sufficient for one set of operations (e.g., duplicating layers), and needed to watch videos for another set they were less familiar with (e.g., the *puppet warp* tool). Overall, these responses suggest that users appreciated being able to choose from static and video features in mixed tutorials.



Command Type	L_V: Likelihood of a video view	Steps of this type in corpus
Brushing, Drawing (e.g., Gradient, Brush tool)	35.4%	12
Manipulating Control Points (e.g., Warping)	25.0%	9
Parameter Adjustment (e.g., Levels, Curves)	19.4%	9
UI Navigation (e.g., finding a submenu)	5.4%	37
Layer operations (e.g, Duplicate Layer)	0%	13

Table 1: Participants watched videos most often for brushing, control point manipulation, and parameter adjustments.

DESIGN GUIDELINES

Based on the findings from our formative study, we propose four design guidelines for creating effective mixed media tutorials that combine text, images and videos.

Scannable steps: Scannable steps provide valuable context and facilitate navigation within tutorials. To leverage these benefits, videos in mixed tutorials should be presented in a format that supports scanning.

Small but legible videos: To make mixed media tutorials scannable and enable users to work with the tutorial and their application side-by-side, the videos for individual steps should use the minimum amount of screen real estate while still being legible. Ideally, videos should clearly depict the most important portions of the UI for each step (e.g., dialog boxes, panels, or canvas) while hiding or deemphasizing less relevant regions.

Visualize mouse movement: Our study indicates that videos are most useful for steps that involve brushing, drawing and manipulating control points, but even in videos, it can be difficult to see the exact motion or path of the mouse during such interactions. Visualizing mouse movement and events helps viewers understand the relevant spatio-temporal characteristics of the demonstration.

Give control to the user: Our observations of user behavior suggest that expertise and familiarity with the specific tools or interactions in a tutorial is likely to have an impact on which instructional format (static or video) is best for a given user. Videos help users understand, confirm and debug steps with unfamiliar tools, while static images and text are quicker and easier to skim. Thus, mixed tutorials should let users choose the most appropriate format at the granularity of individual steps.

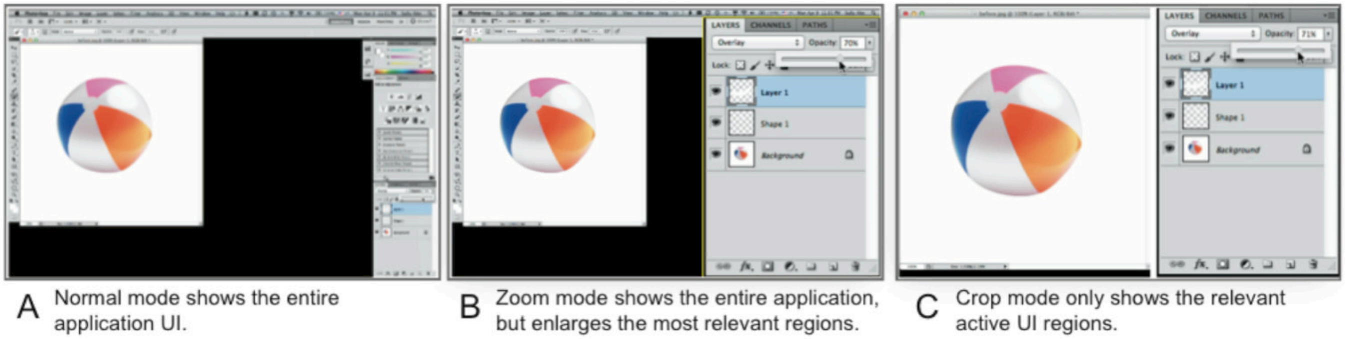

A Normal mode shows the entire application UI.
B Zoom mode shows the entire application, but enlarges the most relevant regions.
C Crop mode only shows the relevant active UI regions.

Figure 6: MixT offers three video playback options: Normal mode (A), zoom mode (B) and crop mode (C).

GENERATING MIXED MEDIA TUTORIALS WITH MIXT

The benefits of mixed media tutorials are unlikely to be realized if creating such materials is too tedious, time-consuming, or if it requires more expertise than creating other tutorial formats. To lower the authoring barrier, we designed MixT, a system that automatically generates mixed media tutorials from user demonstrations. While the MixT architecture can apply to different media creation applications, our current implementation is specific to creating tutorials for Adobe Photoshop.

Tutorial Format

MixT generates HTML tutorials with embedded videos that follow the design guidelines identified in our study. By default, our interface presents a textual description and screenshot for each step, just like a standard static tutorial (see Figure 1A). Clicking on the screenshot replaces the static image with a video player that plays the segment of the original demonstration that corresponds to the written step instructions. For example, a screenshot of a layer panel enhances the instruction "*Select Soft Light from the drop-down menu for Blend Mode*," and the corresponding video clip shows continuous mouse action to the menu, expanding the drop-down menu, moving down to click on the feature, and shows the canvas change. By presenting steps as text and images with video clips that are accessible on demand, MixT tutorials retain the scannability of static tutorials while giving users the option of static- or video-based instruction at each step. To ensure that steps remain scannable and that the tutorial can still be viewed alongside the image editing application (without window switching), we scale each in-place video to at most 700 pixels wide and display text instructions on the left.

Video Playback Options

The MixT video player gives users additional control over the format of playback. Three different modes (normal, zoom, and crop) each emphasize different types of information (Figure 6). In addition, users can display a pointer trace visualization to clarify the path of mouse interactions.

Normal mode shows the entire application window (Figure 6A). This mode preserves all context, but because MixT scales videos down to at most 700x440 pixels positioned next to the text instructions, it may be hard to see precise manipulation or small widgets or handles in the UI.

Zoom mode also shows the entire application window, but performs a non-uniform enlargement of specific UI regions. In particular, the application area being manipulated (e.g., a menu, dialog, or the canvas) is enlarged to fill the full height of the frame and composited on top of the original video in another video layer (Figure 6B). If a dialog also modifies pixels on the canvas, both areas are enlarged and positioned such that they do not overlap. This video composition effectively creates a focus-plus-context view that makes important regions easier to see at a given video resolution [7]. Commercial screencasting software commonly includes a pan-and-zoom technique to make interactions legible in small videos. However, testing early prototypes of MixT suggested that such a technique is not appropriate for brief, single-step videos, as it is hard to establish application context in such short video segments.

Crop mode does not show the entire application — it only shows the currently active area (a tool bar, dialog, main menu, or a panel) and the canvas if being changed (Figure 6C). This offers the unique benefit of showing both a user interface manipulation (e.g., moving a layer opacity slider), and the effect on the image (e.g., parts of the image becoming transparent), while minimizing all other visual distractions. Like the zoom mode, cropped videos are very compact since only the relevant portions of the UI are shown.

Mouse visualization: To help video viewers understand interactions with the canvas, MixT can render a trace visualization of the mouse (Figure 7). These traces show a fading path of the most recent positions of the cursor and encode mouse state using color: *click* events are shown in green (mouse down) and red (mouse up), while mouse *movement* events are shown in purple, and mouse *dragging* events are shown in yellow. Commercial screencasting software also includes mouse visualization techniques. However, they are usually limited to clicking, and the visualizations are typi-

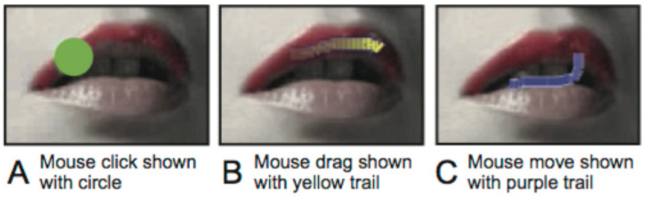

A Mouse click shown with circle
B Mouse drag shown with yellow trail
C Mouse move shown with purple trail

Figure 7: Mouse visualization distinguishes moving and dragging.

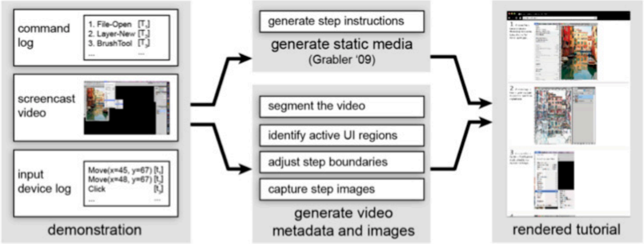


Figure 8: MixT generates tutorials from video and log files.

cally rendered into the final video. In contrast, MixT emphasizes dragging because such interactions are especially relevant for image manipulation. Furthermore, our trace visualizations can be toggled on and off interactively in real-time. By default, the visualizations are enabled for all steps. However, users can change this behavior through an option in the video player.

AUTOMATIC GENERATION PIPELINE

To generate mixed media tutorials, tools can either help authors create new tutorials (e.g., [8]), or they can reformat existing video tutorials into step-by-step videos (e.g., [18]). Our system adopts the former approach and extends Grabler *et al.*'s system for generating static tutorials from demonstration to include videos [8]. Producing mixed media tutorials requires three steps (Figure 8). First, MixT captures an application command log, screencast video and an input device event log and synchronizes them. Second, MixT generates appropriate media files and descriptions: it transforms the log into text instructions, segments the video into steps, and identifies active UI regions, which inform the different video playback modes. Finally, MixT composes the text, images, and video into one document and adds mouse interactions as visualizations on top of the videos. For each step in the tutorial, MixT produces the three video formats and one representative step image.

Recording the Demonstration

MixT records three different time-synchronized data streams during a demonstration: a history of executed application commands; a trace of mouse events; and screencapture video of the entire application interface. To capture application commands, such as opening a file, selecting a region, or hiding a layer, we use Tutorial Builder[1], a freely available Photoshop plug-in that records commands and transforms them into text instructions through text templates. To synchronize Tutorial Builder output with the screencast video and mouse streams, we timestamp the command log during the user demonstration. We obtain mouse event traces on Apple's OS X operating system through the Event Taps[2] API, which observes system-wide input events. We record full-screen video with the commercial Camtasia application[3].

Generating Video Metadata and Step Images

After acquiring the time-synchronized data, there are three technical challenges:

1) Segmenting the video into steps: MixT segments the screencapture video into individual steps based on the timestamps for individual commands in the command log. We map each command Cmd_i with timestamp T_i to a video segment that starts at T_i and ends at T_{i+1}.

2) Identifying active UI regions for zooming and cropping: MixT uses zooming and cropped side-by-side views to preserve legibility for videos at small video frame sizes. To create these specialized views, MixT needs metadata describing the relevant pixel coordinates for each step. Each command in the command log contains information about the logical UI regions that are involved (*e.g.*, the toolbar, a dialog box, or the canvas). However, many commands can be invoked in multiple ways. For example, the *New-Layer* command can be accessed through the application menu, or an icon on the Layer Palette. Therefore, MixT must find and select the correct UI regions from a set of candidates. MixT first uses pixel-based template matching [18] to locate these areas on the screen. MixT then identifies the active UI region by inspecting the mouse event log to see which candidate region received a mouse click at the recorded timestamp T_i of Cmd_i. If there are no mouse clicks detected (e.g., a command invoked via keyboard shortcut), MixT treats the entire frame as the active UI region to ensure the application response is visible in the video.

3) Adjusting segmentation boundaries: While segmenting the video based on the command log timestamps produces a reasonable rough alignment between steps and video segments, there are some cases where a command is recorded after important UI events have taken place. One typical case is menu navigation. For example, to use the *Replace Color* operation, the user moves from the *Image* menu through the *Adjustment* sub menu to the *Replace Color* option. The entire traversal sequence is relevant information as it explains how to reach the menu item. However, the command log only records *Replace Color* when the operation is invoked, which means that a video segment generated based solely on command timestamps would not include the menu traversal.

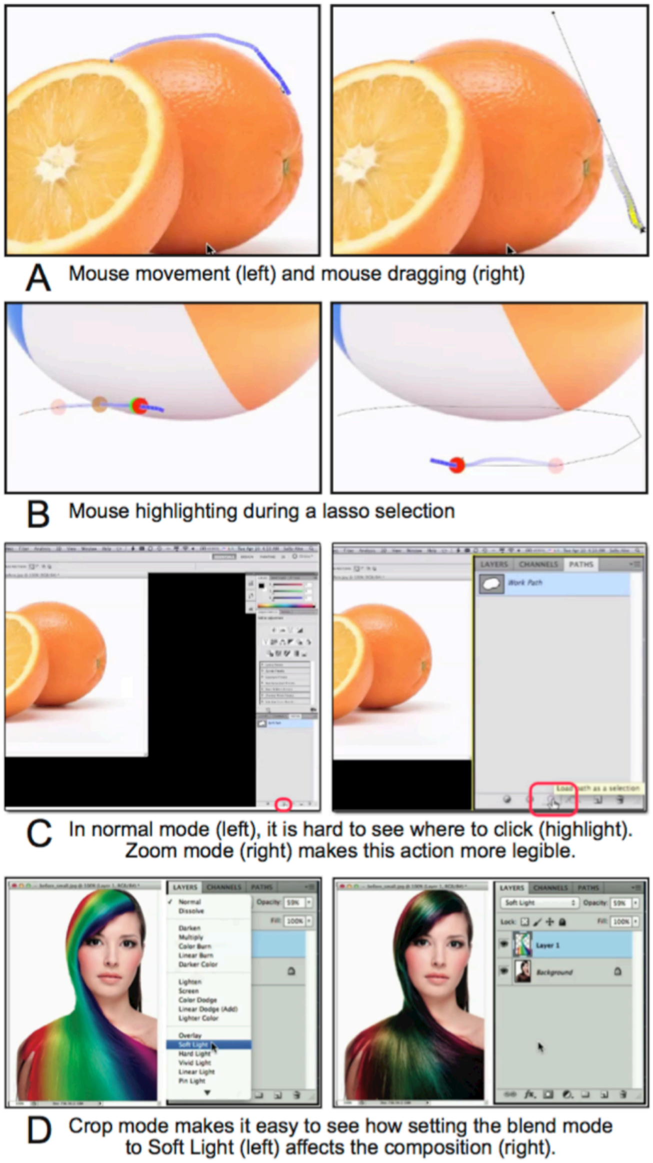
MixT adjusts step boundaries by leveraging template matching and the mouse event log. To compute the start time of the tutorial step for Cmd_i, our algorithm starts at the command timestamp T_i, looks backward for all mouse clicks that occur within any visible candidate UI region for the command (e.g., menus, panels, toolbar) between T_i and the recorded T_{i-1} of the previous command, and sets the adjusted start time of the step T_i' to the time of the earliest mouse click. Adjusting step boundaries in this manner ensures that the step video clip shows all the relevant actions associated with the command.

4) Capturing screenshots and after images: Finally, in order to generate a representative static image for each step, MixT selects the most informative frame of UI interaction

[1] http://labs.adobe.com/technologies/tutorialbuilder/

[2] http://developer.apple.com/library/mac/#documentation/Carbon/Reference/QuartzEventServicesRef/Reference/reference.html

[3] http://www.techsmith.com/camtasia.html


A Mouse movement (left) and mouse dragging (right)

B Mouse highlighting during a lasso selection

C In normal mode (left), it is hard to see where to click (highlight). Zoom mode (right) makes this action more legible.

D Crop mode makes it easy to see how setting the blend mode to Soft Light (left) affects the composition (right).

Figure 9: Automatically-generated MixT results.

from the video. For example, if the step changes the layer blend mode, we show an image of the expanded blend mode menu where the appropriate option is highlighted. If the command involves a dialog box, we show the final frame when the dialog is visible. To do so, MixT selects the last frame that includes any candidate UI region within the duration of the step [T_i', T_{i+1}'] and crops the frame to the UI region to produce the representative static image. Furthermore, MixT also captures the state of the canvas from the last frame of the step as an "after" image that helps viewers understand how the canvas was affected.

The MixT video analysis is implemented in MATLAB. In our dataset of nine test tutorials (see Table 2), the average duration of a step video is 12.4 seconds – our analysis takes an average of 6.3 seconds to process each step.

Composing the Mixed Tutorial

The step-by-step mixed-media tutorial with text instructions, structured screencast video, and step images is composed and presented in a web interface. The MixT tutorial viewer is implemented with standard Web technologies (HTML5, CSS3, and JavaScript) so it can be easily deployed online. In particular, all video compositing and matting is done in real-time using HTML5 video. Just-in-time compositing enables users to change viewing options *on the fly* without pre-rendering multiple video presentations. For example, during the video playback users can choose to show the enhanced mouse movement inside its frame for additional detail of a brush stroke, or disable the effect and focus on the incremental change on a canvas in real-time.

RESULTS

We gathered nine different tutorials and recorded working through each tutorial in Photoshop. We then used MixT to automatically generate mixed media tutorials from these demonstrations. This section describes this corpus. The following section then evaluates the generated tutorials quantitatively and qualitatively.

Our tutorials came from both online and book sources: two from "Adobe Photoshop CS5 Classroom in a Book," five from `photoshopstar.com`, one from `makeuseof.com`, and one from `icanbecreative.com`. All are popular resources for Photoshop learners. Two of the tutorials were also used in the formative user study. The selected tutorials had a total of 165 steps and covered all five command types: they contained 15 brushing/drawing operations, 14 control point manipulations, 30 parameter adjustments, 67 UI navigations, and 39 layer operations. The demonstrations were recorded on three different laptops running Photoshop in full screen with native resolutions of 1680x1050, 1440x900, and 1280x800 pixels.

Overall, the MixT tutorials that we generated exhibit the desired characteristics that we identified in our formative study: scannable steps, small but legible videos, visualized mouse operations, and user control over presentation format. We highlight some interesting results generated by MixT and also refer readers to the provided video figure, which has additional examples.

Scannability: The step-by-step layout of our tutorials makes them easy to scan. For example, at a glance, we see that the tutorial "Turning an Image into an Old Photo" involves several adjustment layer operations, while the tutorial "Creating Artistic Effects" involves more parameter adjustments and brushing commands.

Mouse visualization: Our mouse visualizations help clarify several interactions. They clearly communicate the difference between clicking and dragging, a distinction that is fundamental to operations such as path manipulation but hard to glean from screen capture video. For example, Figure 9A shows the difference between moving around the contour of an object without drawing a path (left), and dragging a Bézier handle to adjust a path segment (right).

Mouse trails and click markers were also useful for showing the trajectory of lasso selections (Figure 9B).

Zoom and crop modes: For many steps, the zoom and crop videos offer clear legibility benefits over the normal video mode. In our corpus, zoom mode was especially valuable for highlighting actions on small buttons that occurred near the frame boundaries, e.g., in the layers palette (Figure 9C right). Such operations are easy to miss in a normal, scaled video (Figure 9C left). Crop mode was useful in showing the effect that parameter selection has on the canvas. Figure 9D shows two successive frames that illustrate how changing a layer's blending mode affects the image. Enlarging the canvas in these modes also helps users see the details of effects, such as applying the *eraser* tool on the canvas to enhance the underlying layer (Figure 7).

EVALUATION

To evaluate MixT, we measure the performance of our automatic tutorial generation pipeline and gather user feedback on the effectiveness of the resulting MixT tutorials.

Expert Inspection of Generated Results

We examined the segmented and cropped videos for each step of our nine converted tutorials and recorded the following errors. If a clip does not include all actions of the current step, we record a *segmentation error*. If the screenshots or zoom/crop videos do not show the appropriate application regions, we record a *region finding error*; if the system fails to identify the active region and shows the overall UI instead, we record a *region finding miss*; if they show some relevant regions but omit others, we record an *incomplete region*.

Table 2 shows the results of our inspection. On average, MixT correctly segmented steps around 92% and found relevant regions of complete views 92% of the time. These error rates suggest that our automatic generation pipeline performs reasonably well for a variety of real-world tutorials, but there is room to improve our segmentation and region-finding accuracy.

Table 2. Error rates for automatically generated tutorials.

Task (time)	Steps	Segmentation Error	Region Error	Region Miss	Incomplete Region
T1 (2'30)	19	15.8%	0.0%	0.0%	10.5%
T2 (3'19)	21	19.0%	0.0%	14.3%	0.0%
T3 (6'02)	30	3.3%	3.3%	0.0%	10.0%
T4 (2'44)	16	12.5%	0.0%	0.0%	0.0%
T5 (4'37)	9	0.0%	0.0%	0.0%	0.0%
T6 (6'10)	21	4.8%	0.0%	0.0%	0.0%
T7 (2'58)	21	4.8%	0.0%	0.0%	4.8%
T8 (3'59)	16	5.6%	11.1%	5.6%	5.6%
T9 (1'41)	12	0.0%	0.0%	0.0%	8.3%
AVG (3'46)	**18**	**7.3%**	**1.6%**	**2.2%**	**4.4%**

User Experiences: Working with MixT Tutorials

We conducted a small user evaluation with four participants (2 males and 2 females, aged 25-29, with 5-12 years of Photoshop experience) to gather feedback on the usability of our MixT tutorials. We selected four of our nine evaluation tutorials with features such as the *brush* tool, *pen* tool, *puppet warp* tool, and *gradient warp* tool – commands that led to many video views in our formative study (Table 1). These test tutorials were generated with an earlier version of MixT that did not use the mouse event log to refine the step segmentation and active UI region finding as described earlier. This previous implementation relied solely on the command log and computer vision, which resulted in lower segmentation accuracy (84%) and region-finding accuracy (90%).

Participants were introduced to the MixT system and then asked to work through the set of tutorials, analogously to our formative study. We asked participants to comment on their process using the think-aloud method, and afterwards asked open-ended questions to elicit additional detail.

Successes. The participants found the same benefits in automatically-generated MixT tutorials as earlier participants found in manually-created tutorials. Participants commented that videos helped them understand steps that were complicated or text that was ambiguous or did not contain explanations why certain steps were taken: *"Videos were convenient when trying to get a sense of a complex operation." / "I tended to watch the videos when the text wasn't clear."* Examples included making a selection from a path (1.75 views per user) and the *puppet warp* tool (2.75 views). Note that a clip might be viewed more than once by a single user. Participants also watched videos to confirm their results because *"The photo/screenshot (...) wasn't as actively helpful in guiding what I needed to do or confirming that I was doing the right thing."*

Multiple participants commented positively on the utility of automatically segmented videos that focus on short step clips about the task at hand: *"I didn't have to sit through 5 mins of intro to get a video description of the task I was interested in" / "What I liked the most about the mixed tutorial was the ability to only watch short segments of video that was relevant. Often with video tutorials I find myself sitting and waiting for the content that I need. Because of this, I tend to avoid video tutorials in favor of text. The mixed tutorial was a nice way to achieving the best of both worlds."*

Shortcomings. Our participants identified useful suggestions for improving our tutorial design. Currently, static images do not provide sufficient *information scent* about the contents of the video – it was hard to judge how long each video was and whether there were remaining important actions in a clip. Therefore, participants sometimes skipped important information that resulted in editing errors (e.g., not adjusting the pose after placing pins using the *puppet warp* tool). In addition, the minimal play/pause interface was deemed insufficient: *"Navigating the videos was difficult [...] It was also hard to go back in the video to observe missed steps. Adding standard playback controls might help."* One approach to remedy this would be to analyze the video and provide thumbnail frames of the video clip that highlight the clip's content and length.

As mentioned earlier, our automatic tutorial generation pipeline computes the correct video segments and finds the right spatial regions to highlight for most steps. However, the few segmentation and region finding errors that users encountered sometimes caused important information to be hidden in crop or zoom mode. As a result, participants referred back to the normal video mode more often than we expected, even though the crop and zoom videos were typically more legible. For the 41 video segments that were watched, the average view counts per step were 0.66 for crop mode, 0.24 for zoom mode, and 1.8 for normal mode. Participants commented on the impact of segmentation errors: *"Sometimes the video doesn't line up – and you have to go to the step before to see what's going on."* We consider this as an opportunity to include the tutorial authors in the loop to modify computer-generated tutorials.

LIMITATIONS AND FUTURE WORK

The current MixT implementation has some important limitations that should be addressed in future work. One missing yet interesting component is the audio content, such as a tutorial author's narration in the video demonstrations. Spoken explanations of the demonstrated actions can help viewers understand the rationale behind a sequence of steps. However, narration and interactions may not always occur in synchrony and it is an open problem to segment combined audio and video tracks appropriately into steps. MixT also does not provide opportunities for the tutorial creator to edit a demonstration. To maximize the benefits of mixed media tutorials, we are interested in exploring ways to provide an editing interface for tutorial authors to easily examine and modify automatic results, and to add annotations that can provide rationale in a lightweight way before sharing their demonstrations.

CONCLUSION

This paper introduced MixT, a system that automatically generates step-by-step mixed media tutorials from user demonstrations. We motivated the design of MixT through a formative study that suggested that step videos help users understand complex direct manipulation operations. MixT's architecture uses a command log, an input device log, and screencapture video to generate tutorials. It applies video compositing techniques to focus on salient information, and highlights interactions through mouse trails. Our informal evaluation suggests that automatically generated MixT tutorials were as effective in helping users complete tasks as tutorials that were created manually.

ACKNOWLEDGMENTS

Work at Berkeley was supported by Adobe, Google, a Berkeley Fellowship for Graduate Studies, and a Siebel Foundation Fellowship.

REFERENCES

1. Bergman, L., Castelli, V., Lau, T., and Oblinger, D. DocWizards: a System for Authoring Follow-me Documentation Wizards. In *Proc. UIST '05*, ACM Press (2005).
2. Brunelli, R. Template Matching Techniques in Computer Vision: Theory and Practice. *Wiley* (2009)
3. Denning, J. D., Kerr, W. B., and Pellacini, F. MeshFlow: Interactive Visualization of Mesh Construction Sequences. In *Proc. SIGGRAPH '11*, ACM Press (2011).
4. Dixon, M. and Fogarty, J. Prefab: Implementing Advanced Behaviors Using Pixel-Based Reverse Engineering of Interface Structure. In *Proc. CHI '10*, ACM Press (2010).
5. Ekstrand, M., Li, W., Grossman, T., Matejka, J., and Fitzmaurice, G. Searching for Software Learning Resources using Application Context. In *Proc. UIST '11*, ACM Press (2011).
6. Fernquist, J., Grossman, T., and Fitzmaurice, G. Sketch-Sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proc. UIST '11*, ACM Press (2011).
7. Furnas, G. W. Generalized Fisheye Views. In *Proc. CHI '86*, ACM Press (1986).
8. Grabler, F., Agrawala, M., Li, W., Dontcheva, M., and Igarashi, T. Generating Photo Manipulation Tutorials by Demonstration. In *Proc. SIGGRAPH '09*, ACM Press (2009).
9. Grossman, T., and Fitzmaurice, G. ToolClips: an Investigation of Contextual Video Assistance for Functionality Understanding. In *Proc. CHI '10*, ACM Press (2010).
10. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. In *Proc. UIST '10*, ACM Press (2010).
11. Harrison, S. A Comparison of Still, Animated, or Nonillustrated On-line Help with Written or Spoken Instructions in a Graphical User Interface. In *Proc. CHI '95*, ACM Press (1995).
12. Kelleher, C., and Pausch, R. Stencils-based Tutorials. In *Proc. CHI '05*, ACM Press (2005).
13. Kong, N., Grossman, T., Hartmann, B., Fitzmaurice, G. and Agrawala, M. Delta: A Tool for Representing and Comparing Workflows. In *Proc. CHI '12*, ACM Press (2012).
14. Matejka, J., Li, W., Grossman, T., & Fitzmaurice, G. CommunityCommands: Command Recommendations for Software Applications. In *Proc. UIST '09*, ACM Press (2009).
15. Matejka, J., Grossman, T., and Fitzmaurice, G. IP-QAT: In-Product Questions, Answers & Tips. In *Proc. UIST '11*, ACM Press (2011).
16. Palmiter, S. and Elkerton, J. Animated Demonstrations vs Written Instructions for Learning Procedural Tasks: a Preliminary Investigation. In *International Journal of Man-Machine Studies* (1991), 34, pp. 687-701.
17. Palmiter, S. and Elkerton, J. Animated Demonstrations for Learning Procedural Computer-Based Tasks. *Human-Computer Interaction*. 8, 3 (1993), pp. 193–216.
18. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. F. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. In *Proc. UIST '11*, ACM Press (2011).
19. Ramesh, V., Hsu, C., Agrawala, M., and Hartmann, B. ShowMeHow: Translating User Interface Instructions Between Similar Applications. In *Proc. UIST '11*, ACM Press (2011), 1–8.
20. Yeh, T., Chang, T.-H., and Miller, R.C. Sikuli: Using GUI Screenshots for Search and Automation. In *Proc. UIST '09*, ACM Press (2009), 183–192.