ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding

Tovi Grossman & George Fitzmaurice
Autodesk Research
210 King St. East, Toronto, Ontario, Canada, M5A 1J7
{firstname.lastname}@autodesk.com

ABSTRACT
We investigate the use of on-line contextual video assistance to improve the learnability of software functionality. After discussing motivations and design goals for such forms of assistance, we present our new technique, ToolClips. ToolClips augment traditional tooltips to provide users with quick and contextual access to both textual and video assistance. In an initial study we found that users successfully integrated ToolClip usage into the flow of their primary tasks to overcome learnability difficulties. In a second study, we found that with ToolClips, users successfully completed 7 times as many unfamiliar tasks, in comparison to using a commercial professionally developed on-line help system. Users also retained the information obtained from ToolClips, performing tasks significantly faster one week later.

Author Keywords
Help, Learnability, Understanding, Video Tool Tips, Tooltips, ToolClips, Balloon Help.

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces - Graphical user interfaces (GUI).

General Terms
Design, Documentation, Experimentation, Human Factors.

INTRODUCTION
Despite all the advances the HCI field has made, today's user interfaces can still be hard to use [16] and frustrating for users [21]. The usability of a system depends on many factors, but it is agreed upon that learnability is one important component [8, 32], if not the most fundamental attribute [24]. In a recent paper, Grossman et al. provides a thorough description of 5 categories surrounding software learnability [14]. Here, we specifically target the *Understanding* learnability issue – the problem users encounter when they have located a tool or function, but are unable to understand how to use it [14].

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

A traditional approach to address this issue is to provide documentation. Unfortunately, text-based on-line help can be difficult to follow [20], causing users to be reluctant to use it [12, 29]. In recent years, video tutorials have become a prevalent source of information for users [28]. However, in general, these are accessed outside of the UI context, through external websites [28]. While contextual forms of assistance, such as tooltips [11], do exist, they rarely possess the level of detail which would be required to understand how to use a complex tool, and do not leverage the potential benefits of animated assistance [15, 33]. It would be desirable if the benefits of media-enriched resources were provided within the application context.

Driven by our goal of addressing the understanding learnability issue, we investigate the possibility of contextual video assistance. Through careful design considerations, we have developed ToolClips, which augment traditional tooltips with extended video and documentation content. In a first study we demonstrate a full implementation of ToolClips within an existing user interface, and show that they can be successfully used within the flow of high-level tasks. In a second study, we demonstrate that ToolClips can significantly improve a user's understanding of how to use UI elements in comparison to a traditional commercially developed on-line help system. Contrary to previous results, we also observe a positive impact on retention of learning. Following our discussion of these studies, we discuss the issues surrounding the implementation of contextual video assistance, such as content creation and localization.

RELATED WORK
Important early results in software assistance, such as those proposing minimalist and task-centered documentation [6, 7, 37] and user-centered on-line help systems [20, 30], have shaped the on-line help systems which are common today. Since the conception of on-line help, there have been explorations into contextual [1, 11, 19] and video-based [3, 18, 19] delivery, with less research exploring both [38].

Video and Animated Documentation
Shneiderman [31] argues that graphical demonstrations can be the most direct way for novices to learn procedural knowledge. However, when the ideas were first proposed in the early 90's, its implementation was expensive and time-consuming [18], and in general, on-line documentation was

"crippled" by hardware and software limitations [12]. The evolution of CPU power, memory capacities, and video hosting services, has now made video documentation a real possibility [22], which we further explore in this paper.

With this emerging potential, numerous researchers have investigated various forms of animated assistance. This ranges from animating icons [4], to providing full animated demonstrations [18, 25, 27, 33], which may involve superimposing moving animated characters of input devices [35] or providing audio narrations [15, 35]. Plaisant et al. propose an important set of starting guidelines for when and how to use recorded demonstrations [28]. Our own design goals are influenced by these guidelines.

Evaluations of Animated Assistance

Despite claims that animated assistance should be considered [31, 34], and certain studies showing some benefits [4, 15, 33], there is no decisive result in the literature which tells us when video assistance is most beneficial, or even if it is beneficial at all. In fact, a number of recent papers have argued against the use of animations [13, 17, 19]. For example, Grabler et al. state that "prior studies have shown that video-based instructions are far less effective than static tutorials because they force users to work at the pace of the video" [13].

The negative remarks towards video assistance are consistently based on the prominent studies done by Palmiter [25-27] and Harrison [15] in the early 90's. Our review indicates that the results produced by these early studies are not entirely negative.

To the contrary, Palmiter found that animated demonstrations allowed users to complete tasks significantly faster than text-only instructions [25]. It was only after a week that the users did not perform as well. However, the authors state that this was likely because the initial demonstrations matched the users' tasks exactly, allowing them to perform "superficial processing" by "mimicking the animated demonstrations".

Harrison's work [15] showed a lack of significant results when comparing animated assistance to still graphics, but never claimed that animations were worse. In fact, the authors reported a slight advantage towards the animations. Furthermore, the tasks involved navigations through a menu system and "did not involve the aspect of motion beyond the movement of the cursor". In tasks of a more graphical nature, the benefits of animation may be more prominent.

Contextual Assistance

Traditionally, there has been an explicit distinction made between the documentation of the software and its user interface [12, 19, 20, 34]. Despite psychology research which exposes the benefits of contextual assistance [2], help systems are generally offered through completely separate components. This can result in delayed, disruptive, inconsistent, and obtrusive help systems [1, 20].

One of the most successful forms of contextual assistance is tooltips [11], which provide short textual descriptions of UI elements when the cursor hovers over the item. Tooltips have been shown to improve a user's ability to locate functionality [10]. However, because they are so brief, tooltips are less useful for learning how to use tools [11].

Microsoft's "Enhanced Tooltips"[1] and Autodesk's "Progressive Tooltips"[2] provide longer descriptions and static images, but still do not provide full usage descriptions. Adobe's "Knowhow"[3] delivers detailed contextual help, but in a persistent window. Side Views [36] display dynamic previews in the pop-up window of a command, but do not teach users how to use the command.

Recent developments in tutorial content have included contextual associations, such as Stencils-based tutorials [19], Graphstracts [17] and "Photo Manipulation Tutorials" [13]. However, these tutorials are not *accessed* contextually, and are each geared towards a single, specific task. We seek a technique which allows users to quickly access relevant contextual assistance during application usage, and which is independent to the user's task.

Contextual Video Assistance

The two concepts of video and contextual assistance have rarely been combined. Previous research in animated assistance has not explicitly addressed content access [15, 26]. Existing video tutorials are generally accessed outside of software applications, through external websites, not taking advantage of convenient contextual associations.

One good example is Google Sketchup[4], which has an "instructor" providing contextual animated assistance. However, the animated content is used to demonstrate what the tool can do and not necessarily help the user learn how to use it. We seek a technique that can better address the *Understanding* learnability issue. The GestureBar [38] is an interesting recent design which displays short animations when their associated icons are selected, to show users how to perform gestures. With ToolClips, we extend this idea to the general problem of learning software functionality.

DESIGN GOALS

In the following, we will design and evaluate a new help resource that provides detailed tool-based information through contextual video assistance. Based on previous research, we seek to satisfy the following design goals.

1. On-line

Since users typically do not read manuals [7, 15, 23, 25, 30], our main interest is directed at users encountering interface challenges during actual usage. Thus, we focus on assistance that is conveniently accessed *on-line*: within the flow of the user's primary task.

1 http://msdn.microsoft.com/en-us/library/cc872782.aspx#enhancedTooltips, 01/04/2010
2 http://images.autodesk.com/adsk/files/whats_new_in_revit_structure_2010.pdf, 01/04/2010
3 http://labs.adobe.com/technologies/knowhow/, 01/04/2010
4 http://sketchup.google.com/, 01/04/2010

2. Contextual
We consider *contextual* assistance to be assistance which is linked and accessed through to the system's UI [4, 11]. This is especially important for the *Understanding* issue, since, by definition, users have already arrived at a tool of interest.

3. Detailed
The assistance should contain as much relevant information as possible. When users fail to find information through the help system, they may become reluctant to use it [12].

4. Segmented
Evidence has shown that when users are in the flow of their task they will only want to acquire the minimum information that they need [6, 7]. By providing segmented videos, users should be able to quickly identify and navigate to content they are interested in [15].

5. Low Transaction Cost
If we hope for the assistance to be used, there should be a low transaction cost associated with its access [5]. Users should not need to worry about codec's, browser plug-ins, and download speeds [17, 28]. The user should also be able to return to their own work quickly and without any awkward interactions or jarring transitions.

6. Unobtrusive
A contextual help system should not persistently occlude the user's working data, as users may want to visually compare provided examples to their own work [20].

SYSTEM IMPLEMENTATION

Our work is implemented within Paint.NET[5], an open source raster-based image editing application. This application was chosen because it is a generic graphical application. Video content is displayed using WPF, within the Paint.NET Visual Studio C# Solution.

TOOLCLIPS

ToolClips extend traditional tooltips by integrating narrated video clips and in-depth textual documentation. A new progressive interaction model is used to ensure that they will still be *unobtrusive*. This provides a simple, yet unexplored solution, which could easily be incorporated into most GUI programs, which already possess tooltips.

Displaying ToolClips and Their Initial Appearance

Just like traditional tooltips, ToolClips fade in when the cursor dwells over an associated visual user interface component for 400ms, providing a *low transaction cost*. ToolClips are positioned directly to the right of vertical palettes (Figure 1), and directly below horizontal palettes, so that other icons are not occluded. In their initial form, ToolClips are compact, similar to traditional tooltips, with the addition of a text and media icon that the cursor can click (Figure 1). This ensures that they will be *unobtrusive*.

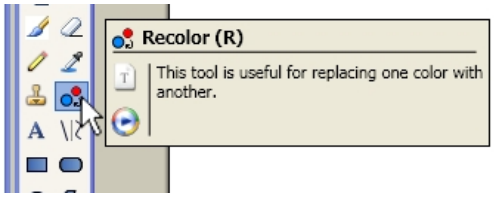

Figure 1. Upon initial display ToolClips appear similar to traditional tooltips, but also contain a text and media icon.

Dismissal

Traditional tooltips fade out as soon as the cursor leaves the associated tool icon. Because ToolClips contain interactive content, this behavior must be modified. Instead, ToolClips slowly fade out as the cursor moves away from it. The alpha level of the ToolClip (ranging from 0 to 1) is calculated as:

$$Alpha = \frac{100 - (D - D_{\min})}{100},$$

where D is the distance between the cursor and the rectangle bounding the text and media icons (or 0 if inside this rectangle), and D_{min} is the minimum value that D has taken on since the ToolClip was displayed. The ToolClip is dismissed if the alpha level reaches 0. Thus, if the distance to the ToolClip ever increases by 100 pixels, the ToolClip will fade out, further supporting their *low transaction cost*. This type of distance-mapped alpha level is similar to the behavior of the Office 2007 "Mini Toolbar"[6].

Accessing Extended Content

The unique dismissal mechanism of ToolClips allows them to contain interactive content. The extended content of ToolClips is accessed by clicking on either the text or the media icon. The text icon will bring up the documentation content of the ToolClip, and the media icon will access the video content. Because a dwell is needed to activate a new ToolClip, the cursor can pass over intermediate icons while travelling to access a displayed ToolClip.

When extended content is accessed, ToolClips enlarge to a movable and sizable window. The persistency of the window allows them to be placed outside of the application's boundaries, so that they can remain *unobtrusive*, allowing users to switch between viewing the ToolClip's content and working with their document. Once the ToolClip becomes persistent, it is closed through the close icon in its top right corner. This is similar to the interaction model of Side Views [36].

Extended Video Content

Clicking on the media icon transforms the ToolClip into an interactive media player, containing narrated video tutorials on the associated tool (Figure 2). Alternatively, the scroll-wheel can be rolled in either direction once the initial ToolClip has appeared to quickly display the video content.

5 http://www.getpaint.net/, 01/04/2010

6 http://office.microsoft.com/en-us/word/HA101736241033.aspx, 01/04/2010

Figure 2. Clicking on the media icon transforms the ToolClip into a media player with video content.

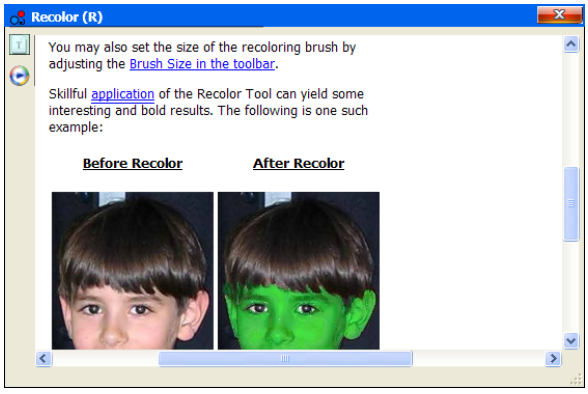

Figure 3. Clicking on the text icon enlarges the window to provide documentation on the tool.

Video Clips

Each ToolClip includes one initial video and up to three additional clips. To support our *segmented* design goal, the actual videos are made to be as short as possible, and are segmented based on topic or primary lesson. The first video, which begins playing as soon as the video content is accessed, gives the user a bare minimum demonstration of how to use the tool. Subsequent clips branch off into various other lessons relevant to the tool. Preliminary evaluations led us to generating clips 10-25 seconds in duration. The media player had all of the typical navigation functionality, including a "next button", so that users could quickly jump to the next video in the series (Figure 2).

Video Thumbnails

Our initial evaluations showed that it was important to display video thumbnails linked to other clips available in the series (Figure 2, right). This allows users to quickly locate and navigate to relevant content. Viewed as a group, they also provide the user with a pictorial overview the tool.

Switching between ToolClips

Users can switch between ToolClips for different UI components. If an existing ToolClip is in its initial smaller form, it will be dismissed as soon as the cursor dwells on a new tool and the new ToolClip is displayed. If an existing ToolClip is in its larger persistent form, it will persist until the user accesses a new tool's extended ToolClip content, at which point the existing ToolClip will be dismissed. If the user has modified either the position or size of the ToolClip being replaced, then the new ToolClip will take on these previous values. This allows users to set up preferred viewing locations and sizes.

Extended Documentation Content

While our focus was on exploring contextual video content, we wanted ToolClips to be as *detailed* as possible. For this reason, we also provided a text icon on the ToolClips, which once clicked, would provide further documentation about the tool. The content is provided in a scrollable window (Figure 3). This behavior is similar to some tooltips which allow users to press F1 to go directly to the on-line help system article relevant to the associated tool [11]. By providing the content within the ToolClip, we eliminate the need for any external browsing systems or software, which may have unpredictable behaviors. Thus, ToolClips maintain their *low transaction cost*.

STUDY 1: USAGE OBSERVATIONS

We conducted two studies comparing ToolClips to traditional help facilities. In this initial study, we obtain feedback and usage data on our design, and gain insights into potential design iterations. This study looks at high-level Paint.NET tasks that require multiple tool usages. The second study isolates and focuses on individual tool usage understanding, to determine how well ToolClips address the *Understanding* learnability problem.

Apparatus

The study was run on a Dual 2.4Ghz Windows XP PC with a standard keyboard and mouse, on a 30" display at 2560x1600 resolution. The application was run at 1024x768 centered in the display, allowing users to view help content off the main window if desired. Headphones were provided in the ToolClips condition.

Participants

We recruited 16 (10 male, 6 female) paid participants, ranging in age from 17 to 32. We targeted participants with little or no experience with Paint.NET, since our interest was in users who would need to learn how to use the tools.

Procedure

We designed a series of eight tasks that progressed in complexity, and taken together exhausted the 22 tools in the main Paint.NET tool palette. Each participant completed the eight tasks in the same order. The tasks generally involved the use of 2-5 tools. For each task, users were given a different template file to begin with, and a printout of the goal image which they needed to create.

An experimenter was present throughout the entire study to make observations and provide assistance. Assistance was only provided if the user had been stuck for 2 minutes,

timed by the experimenter. Assistance was made as brief as possible to maintain consistency across tasks and participants. The experimenter would first suggest that the users try to obtain help, and if that failed, would provide the necessary help. This allowed us to gather more usage data on ToolClips, which was the main goal for this study.

Help Condition

In the baseline condition, tooltips were available for each of the icons in the tool palette, providing the name and a short description of the tool. When a tooltip was displayed, users could hit F1 to go straight to the existing Paint.NET help article for that tool. Users could also hit F1 when a tooltip was not visible to go to the main Paint.NET help page[7].

ToolClip Condition

ToolClips were provided for all 22 tools in the main tool palette. The integrated video content provided the same information as the on-line help, but in video format. An animated cursor icon was overlaid on the videos to indicate the cursor position and button presses [35].

Design

A between-participant design was used. The independent variable was the condition (*Help*, *ToolClip*).We divided the 16 participants into 8 pairs, so that members within each pair had similar experience with graphics editing programs. For each pair, we randomly assigned the members to one of the two conditions. Participants performed the eight tasks in a single session lasting 45-75 minutes, depending on the user's pace. Each task lasted roughly 5 to 10 minutes. We did not use any warm-up tasks, but we did provide users with an overview of the program and description of the resource (*Help* or *ToolClips*) which was available.

Results

ToolClip Usage

The extended ToolClip content was accessed an average of 2.08 times per task, or a total of 118 times, 46 times through the text icon and 72 times through the media icon. The scroll wheel was used to access the video content 24 times. An interesting effect was that most users preferred either the documentation content or the video content, and few relied on both (Figure 4). These users indicated after the study that they chose the medium they were personally more comfortable learning from. This validates our decision to include both text and video content within the ToolClips.

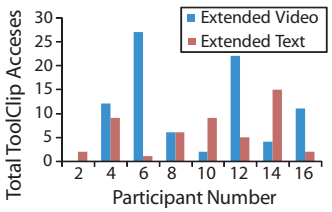

Figure 4. Video vs. text usage of ToolClips.

Once opened, users only switched between the text and video 6 times. However, users switched between the video segments within a ToolClip a total of 54 times. This demonstrates that the segmented videos were a popular feature. However, in some cases users were unsure which segment to view. Because each clip was so brief (10-25s), users rarely navigated the timeline (only 2 times).

Traditional Help Usage

Users in the *Help* condition strongly preferred the help content linked from tooltips (by hitting F1, once the tooltip was displayed). Such content was accessed an average of 1.03 times per task, whereas the global help system was used only 0.21 times per task. On a couple of occasions, users closed the global help system so that they could use the linked help instead, since it made it easy to locate the desired article. This demonstrates the importance of providing contextual assistance for static help as well.

Traditional Tooltip Usage

We recorded a tooltip use if it was visible for at least 0.5 seconds. Tooltips were extremely valuable to users. Across all tasks, tooltips were used an average 8.87 times per task. Consistent with previous work [10], our observations indicated that tooltips were predominately used for locating desired functionality, but rarely helped a user understand how to use the tool.

Learnability Issues Encountered and Overcome

The between-participant design allowed us to obtain an initial comparison with traditional help. Because the experimenter assisted users when they were stuck, we could not use completion time as an accurate measure for comparison. Instead, we recorded the number of times a user got stuck (for two minutes), and the number of times a participant successfully used a resource (*Help* or *ToolClip*) to independently solve a problem.

Repeated measures analysis of variance showed that the condition did have a significant effect ($F_{1,14} = 7.90$, $p < .05$) on the number of times users got stuck per task, with values of 1.19 for Help and 0.53 for ToolClip. Similarly, the condition had a significant effect ($F_{1,14} = 6.03$, $p < .05$) on the number of times a help resource was successfully used to overcome a problem, with 0.45 successful uses for Help, and 1.00 for ToolClip (Figure 5). Within the ToolClip condition, 30% of the successful uses were from viewing the integrated documentation, while 70% of the successes came from viewing the video content.

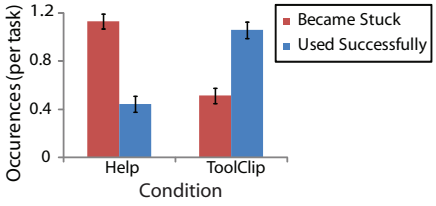

Figure 5. Rate at which users became stuck and used the resources successfully.

7 http://www.getpaint.net/doc/latest/en/index.html, 01/04/2010

Subjective Results

After the study, participants rated their assigned resource from 1 to 5 on how helpful it was (avg. 4.3/5), if it got in the way of completing the task (avg. 1.6/5), if it was easy to use (avg. 4.3/5), and overall enjoyment (avg. 4.2/5). The condition did not have a significant effect on the responses. We feel the most important result to be taken from this is the low score when asked if the technique "got in the way". This was a potential concern with ToolClips, but the low score, in addition to our own observations, indicated that this was not problematic.

DESIGN ITERATION

The results from the initial study were quite positive. The potential problem, that ToolClips would be intrusive, was not observed. Users managed the location and size of ToolClips to their liking. The ability to view documentation in addition to video content proved useful for the users that preferred getting help in a text format.

The only usability issue we noticed was that users sometimes found it difficult to identify a relevant clip from the thumbnail image. As such, the only design iteration we performed was to include a title and caption with the thumbnail images, to help users identify a clip of interest (Figure 6). This increases the width of the ToolClip, but, only after the user clicks on the media icon.

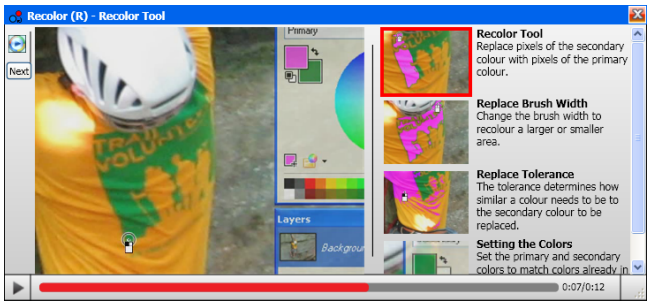

Figure 6. The revised design of ToolClips added captions and short text descriptions for each of the video segments.

STUDY 2: EMPIRICAL EVALUATION

To determine if ToolClips are effective at reducing *Understanding* learnability problems, we preformed a study where participants were required to complete lower level tool-based tasks. To ensure that the users would need to learn how to use the tools, and not just guess, we conducted this study with AutoCAD, a design and architecture application that is notoriously difficult to learn. To implement ToolClips within AutoCAD, we modified our Paint.NET application to run in the background, and display ToolClips when the cursor dwelled over AutoCAD icons.

Apparatus

The apparatus was the same as in the initial study. The only difference was that audio was provided through external speakers, so users would not have to wear headphones.

Participants

Ten (4 male, 6 female) paid volunteers, ranging in age from 18 to 40, participated in the study. We recruited competent computer users, but with no experience using AutoCAD, and only novice experience with other graphics manipulation software systems, such as Photoshop.

Procedure

We choose six basic level tools in AutoCAD to study (Mirror, Extend, Dimension, Fillet, Copy, and Hatch). One task was designed for each of these tools. Each task required 1-3 uses of the tool, which exposed some of the options available for that tool. The tasks would take an expert 1 to 2 minutes to complete. Users were given a template file to begin each task with, and a printout of the goal image which they needed to create. Users were also told which tool they would use to complete the task, and where it was located. Figure 7 shows two of the tasks.

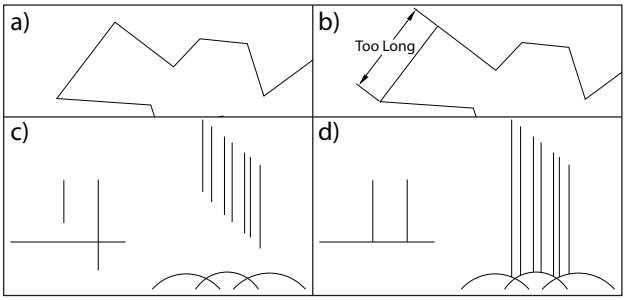

Figure 7. Two of the study tasks. a) Original image for the "Dimension Tool" task. b) Required final result. c) Original image for the "Extend Tool" task. d) Required final result.

An experimenter was present throughout the entire study to make observations, but no assistance was provided. Instead, a task ended if it had not been completed after 8 minutes. Before the study began, participants were instructed to use the provided help resources if unsure how to complete a task, to deter random trial and error.

Help Condition

In the baseline condition, a tooltip was available for the task's tool, providing the name and a short description of the tool. When the tooltip was displayed, users could hit F1 to go straight to the on-line help for that tool.

ToolClip Condition

The initial view of the ToolClips was the exact same as the tooltips, with the addition of the media icon. We did not include a text icon, since our goal in this study was to compare traditional help to contextual video assistance.

Content Generation

The articles used in the *Help* condition were a modified version of the existing AutoCAD "Command Reference" help articles[8]. Because of recent research showing the benefit of static images [13, 17], we manually inserted

8 http://docs.autodesk.com/ACD/2010/ENU/AutoCAD 2010 User Documentation/, 01/04/2010

cropped screenshot images at important steps of the documentation files. The *ToolClip* video content was generated to provide similar information as the on-line help.

To ensure that the information provided in both mediums was equal, we performed an equivalencing procedure [26]. Two raters viewed the help and video content for each of the six tools, and were asked to note any information inconsistencies, and give an overall rating of equivalence, from 1 (the exact same information is provided) to 5 (the information being provided is completely different). The average consistency rating, across both raters, was 1.25, with only minor differences being highlighted. We edited the help articles to compensate for these differences and did not carry out a second equivalencing, since the initial pass produced a suitably high rating [26].

Design

To control for participant effects, we used a within-participant design. The main independent variable was the condition (*Help*, *ToolClip*). The experiment was initially planned for a single session. However, because of the results, which will be discussed below, we performed a second follow-up session one week later. In each session, the participants performed all 6 tasks in the same order. The 6 tasks were divided into two blocks of 3 tasks, with the condition changing between blocks. Half the participants were assigned to *Help* in the first block, while the other half were initially assigned to *ToolClips*. The complexities of the tools encountered in each block had a similar range.

In the first session, the users were first given a 10 minute guided introduction to AutoCAD, so they could learn the basics of the user interface. Before each block began, the condition was described to the user. This session lasted about 60 minutes. In the second session, the introduction to AutoCAD was omitted, but users were still given 2 minutes to refamiliarize themselves. This session lasted 30-45 minutes.

Session 1 Results

Completion Rates

In the first session, participants had much more difficulty in the *Help* condition than we had anticipated. Across all 10 participants, only 3 of the 30 tasks were completed successfully. This was not the case with the *ToolClip* condition, and there was a strong effect of the condition on completion rate ($F_{1,9} = 25.1$, $p < .001$), with completion rates of 10% for *Help* and 70% for *ToolClip* (Figure 8). The results were uniform across each of the 6 tasks (Figure 9).

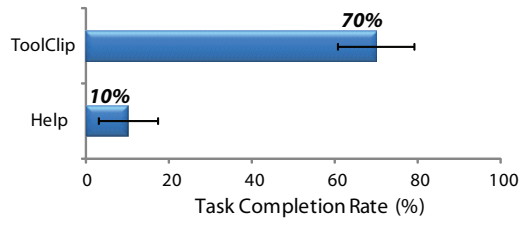

Figure 8. Task completion rates for *ToolClips* and *Help*.

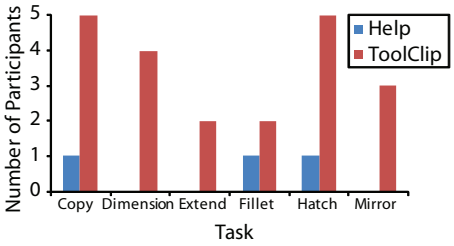

Figure 9. Number of participants that successfully completed each task, out of a maximum of 5.

Completion Times

To analyze completion time, we removed trials that were not completed within the allotted 8 minutes. A statistical comparison of completion times between conditions was not possible, since only 3 of the 30 tasks were completed in the *Help* condition. The average completion time for the *ToolClips* condition was just over 5 minutes (306 seconds).

Resource Usage

For both conditions, we tracked the number of times the help resource was activated. In every trial of the study, the *Help* or *ToolClip* was used at least once. *Help* was accessed an average of 1.6 times per trial. In general, users brought the article up once, and then placed it to the side of their working window, and switched between windows when necessary. The access rate for *ToolClips* was higher but not significantly at 2.03 per trial ($p = .36$). The general usage was similar to the help – users would bring up the ToolClip, watch a segment once or twice, and then place the ToolClip off to the side for future reference. This validates our design decision to make the ToolClip a persistent window, once the extended video content was accessed.

Subjective Results

Unsurprisingly, the subjective questionnaire we administered after the session indicated a strong preference towards ToolClips. When asked about the helpfulness of each condition, ratings were significantly higher for *ToolClips* (4.6/5) than for *Help* (2.9/5) ($F_{1,9} = 11.8$, $p < .01$). Participants were also asked to rank their condition preference, with 1 representing a preference towards *Help* and 5 representing a preference towards *ToolClips*. The average response was 4.5 (Figure 11). No participants indicated a preference towards the *Help* condition.

Iteration of Help Content for a Second Session

While we had expectations that ToolClips would improve completion rates, we were surprised to see such poor results for *Help*, given that each task only required the use of a single basic tool, which users were directed towards. This demonstrates how significant the *Understanding* learnability problem can be, especially for a complex software application. The increase in completion rates for ToolClips demonstrates that contextual video content can assist with the *Understanding* problem.

Clearly, these results are promising for ToolClips, as they drastically improved completion rates in comparison to an existing, professionally developed, on-line help system. While this comparison to an existing help system is a previously used methodology [17], we are still left wondering if the on-line help system could be improved without the addition of videos. In particular, the help pages in AutoCAD tend to be more *function-oriented* in nature [9], describing what the tool does, how to access its features, and what these features do. In contrast, ToolClips are inherently more *operative* in nature [9], providing the information on the tool through procedural demonstrations.

We decided to conduct a follow-up session to see if the format of the help content would have an impact on our results. We invited the same users back, so that we could also test retention from the ToolClip condition.

We created 6 new help articles for each of the tools, made to be static representations of the ToolClip content. Each new help page was divided into sections, matching the video segments from the ToolClip. Above each section, we displayed the same representative thumbnail used for the associated ToolClip segment. Within each section, the narrations from the associated video segment were transcribed and separated into numbered steps. Beside each step, a representative image from that section of the video was provided, except for simple steps such as "Hit Enter".

Session 2 Results

Task Completion Rate

In this session, the completion rates were closer. The completion rate for ToolClips was still higher (83.3% vs. 56.7%), but the difference was not significant at the .05 level ($F_{1,9} = 3.27$, $p = 0.10$). It is interesting to note that the completion rate for the new static help was also lower than the rate in the *first* session for ToolClips (70%) (Figure 10). This is somewhat surprising since users already had an entire session of experience before using the new static help. Although this difference was not significant (p=.34), it does indicate that ToolClips may be beneficial, even in comparison to *operative* static help.

Task Completion Time

The completion time was significantly affected by the condition ($F_{1,40} = 6.05$, $p < .05$), with average completion times of 242s for *Help* and 166s for *ToolClip*. Another important effect is that completion time was significantly reduced for the ToolClip condition between the first (306s) and second (166s) sessions ($F_{1,44} = 21.5$, $p < .001$) (Figure 10). This contrasts previous results [25-27], demonstrating that video assistance does not necessarily have a negative impact on retention, if the resource can be readily accessed.

Resource Usage

The usage of the resources in the second session was similar as the first. Users still accessed the *Help* or *ToolClip* at least once in every trial. However, we observed that users had a much better idea of where to navigate within the content to obtain the desired information.

Subjective Feedback

We asked participants to rate the helpfulness of the new operative help articles. Participant responses increased from 2.9/5 from the first session to 4.1/5 for this session. This value was no longer significantly different from the helpfulness rating (4.6/5) given for *ToolClips* ($p = 0.48$). However, when asked to rate their preference between the ToolClips and the new static help, users still preferred ToolClips. The rating only reduced slightly from 4.5/5 to 4/5, with 7 of the 10 participants still indicating a preference towards the ToolClips (Figure 11).

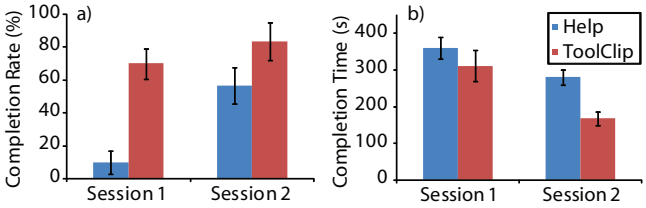

Figure 10. a) Completion rates across both sessions. b) Completion time across both sessions.

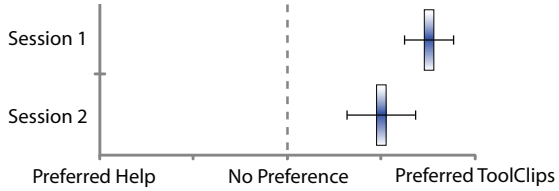

Figure 11. Participant preferences. In Session 2 the comparison was between ToolClips and the new static help.

DISCUSSION

Summary of Results

Our first study demonstrated a system wide implementation of ToolClips within Paint.NET. The UI design was well received, and users were able to utilize ToolClips successfully within the flow of their tasks. While some participants preferred the integrated documentation, most preferred to use the video content.

In our second study, we tested the effect of ToolClips on the *Understanding* learnability issue by designing tasks within AutoCAD that isolated a single tool at a time. ToolClips provided a significant advantage in comparison to AutoCAD's existing on-line help system, increasing task completion rate from 10% to 70%. In a second session, ToolClips also showed benefits in comparison to operative static help articles that were designed to mimic the ToolClip content, with 7 out of 10 participants still preferring the video assistance. Furthermore, the video content did not appear to be detrimental to retention.

We made a number of observations in the experiment that might explain the benefit of the animated nature of ToolClips, in comparison to the static help.

Most importantly, users could continuously monitor the procedure for completing a task. As a result, users were more likely to identify their errors or recognize when they were on the right track. For example, when changing the text in a linear dimension, the dimension line temporarily disappears. The static images in the initial and revised help article did not capture this subtle change, and this unexpected behavior often caused users to assume they had done something wrong. With the ToolClips, users expected and recognized this behavior, since they had observed it in the video. While some of these subtle visual changes could be described or illustrated in static help articles, it would be virtually impossible to convey the entire and continuous visual experience to the user without an actual video.

In addition, with the videos, users could quickly understand what object an instruction referred to. In the static help, users sometimes misinterpreted the reference object of an instruction, even when images were provided.

Lastly, users also seemed to have more "trust" in the videos than the static help. We often observed that when following a specific section of the static help, if the user made a slight error, they would give up on that section of the help. With the videos, users seemed to be equally likely to make initial minor errors, but seemed more likely to try to identify their mistake, rather than seeking out a different video segment.

One drawback of the ToolClips we observed was that users sometimes wanted to work as the video was playing, but could not match its pace. As a result, they had to go back and forth to pause and play the video, so that they could carry out the same steps as the video, one at a time. This type of strategy was easier to carry out with the static help, since the steps generally appeared in a single view.

Scope of Implications and Relation to Previous Work

The positive results for video assistance contrast previous studies [15, 25-27] and claims [13] regarding video assistance. We do not contest these previous findings, since our results only pertain to certain settings. Here, we outline the scenarios that our results can be generalized across.

First, the target applications that we studied are highly graphical in nature, involving continuous cursor and object movements. This is quite different from the original studies of animated assistance [15, 27], which were based on navigating traditional GUI menu and icon systems.

Second, in our study, the videos did not demonstrate the actual tasks that users were trying to perform. As a result, users could not just mimic observed behavior, as they could in many of the previous studies on tutorial help [13, 15, 17, 25-27]. We believe that this is the reason the videos did not have a negative impact on retention.

Third, in our second study, the tasks were low-level and only required the use of a single tool, since we were focusing on the *Understanding* problem. In this problem domain, the short and segmented videos that ToolClips provide may be particularly appropriate.

In addition, our studies targeted novice users. Our experiences and observations lead us to believe that ToolClips would also be useful for intermediate or even expert-level users that are exposed to a new tool for the first time. However, this does require further investigation.

The most positive result of our studies came from the significant improvement in comparison to AutoCAD's existing on-line help system. It is important to highlight that this result does not imply that there is such a dominant advantage of video help over *all* forms of static help. A benefit of ToolClips is they have the capability to deliver static help as well, through the extended text icon. Our initial study showed that users are able to choose the modality they are most comfortable with, and the second session of our second study showed that the static help, which is provided, should be *operative* in nature.

FUTURE WORK AND CONCLUSION

Our work has identified and validated some of the beneficial properties of contextual video assistance, and opens up some new areas, which we now outline.

A potential challenge for any form of video assistance is the time and effort required to develop video content. One solution would be to allow end users to develop and share their own tutorials. The abundance of existing independent video tutorials indicates that users would be willing to do so. For example, there are over 10 000 hits returned for the query "Photoshop Tutorial" on Google Video.

As an early exploration into this potential, we integrated a "developer" mode into our system, which allows simple creation of new clips within the actual application. In the developer mode, a record icon is displayed in the ToolClip (Figure 12). Clicking on this icon begins a video screen capture. In our prototype, a captured video is added to a local directory for the specified tool. In a commercial implementation, the new content could be uploaded to a central server, similar to Microsoft's "Community Clips"[9]. A unique aspect of this design is that the recording is initiated through a ToolClip, so the video can be associated with the tool, and subsequently accessed contextually by other users through that tool's ToolClip.

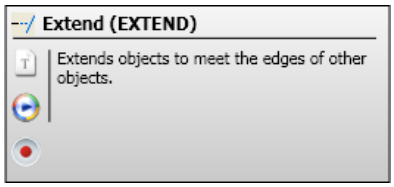

Figure 12. In the developer mode, a record icon is displayed in the ToolClip to create and upload a new video.

If such a system were implemented, there would need to be a mechanism to manage the extra content. Future work could explore different combinations of automatic filtering and administrative monitoring.

[9] http://communityclips.officelabs.com/, 01/04/2010

To conclude, we have proposed, implemented, and evaluated contextual video assistance for helping with the *Understanding* learnability problem. Through careful adherence of a set of design goals, we implemented a new technique, ToolClips, which, in a series of studies, were shown to have some important beneficial qualities. Based on our results and observations, we believe that ToolClips would be a useful addition to graphical applications, and have great potential for improving software learnability.

ACKNOWLEDGEMENTS

Ian Chan, Hyunyoung Song, and Julian Lepinski.

REFERENCES

1. Ames, A. L. (2001). Just what they need, just when they need it: an introduction to embedded assistance. *ACM SIGDOC*. 111-115.
2. Anderson, J. R., Boyle, C. F., Farrell, R. and Reisner, B. J. (1984). Cognitive principles in the design of computer tutors. *Cognitive Science Society*. 2-9.
3. Baecker, R. (2002). Showing instead of telling. *ACM SIGDOC*. 10-16.
4. Baecker, R., Small, I. and Mander, R. (1991). Bringing icons to life. *ACM CHI*. 1-6.
5. Buxton, W., Billinghurst, M., Guiard, Y., Sellen, A. and Zhai, S. (2009). *Human Input to Computer Systems: Theories, Techniques and Technology - A Work in Progress*. www.billbuxton.com/inputManuscript.html
6. Carroll, J. M. (1990). *The Nurnberg Funnel*. MIT Press.
7. Carroll, J. M. and Rosson, M. B. (1987). *Paradox of the active user*. *Interfacing thought: cognitive aspects of human-computer interaction*. MIT Press. 80-111.
8. Dix, A., Finlay, J. E., Abowd, G. D. and Beale, R., *Human Conputer Interaction*: Prentice-Hall, Inc.
9. Dutke, S. and Reimer, T. (2000). Evaluation of two types of online help information for application software: Operative and function-oriented help. *Journal of Computer-Assisted Learning*. 16:307-315.
10. Ehret, B. D. (2002). Learning where to look: location learning in graphical user interfaces. *CHI*. 211-218.
11. Farkas, D. K. (1993). The role of balloon help. *Journal of Computer Documentation*. 17(2):3-19.
12. Goodall, S. D. (1992). Online help: a part of documentation. *Systems Documentation*. 169-174.
13. Grabler, F., Agrawala, M., Li, W., Dontcheva, M. and Igarashi, T. (2009). Generating photo manipulation tutorials by demonstration. *ACM SIGGRAPH*. 66.
14. Grossman, T., Fitzmaurice, G. and Attar, R. (2009). A Survey of Software Learnability: Metrics, Methodologies and Guidelines. *ACM CHI*. 649-658.
15. Harrison, S. M. (1995). A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. *ACM CHI*. 82-89.
16. Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *Int. J. of Human-Computer Studies*. 64(2):79-102.
17. Huang, J. and Twidale, M. B. (2007). Graphstract: minimal graphical help for computers. *UIST*. 203- 212.
18. Jackson, S. (2001). Editing computer hardware procedures for multimedia presentation. *ACM SIGDOC*. 68-72.
19. Kelleher, C. and Pausch, R. (2005). Stencils-based tutorials: design and evaluation. *ACM CHI*. 541-550.
20. Knabe, K. (1995). Apple guide: a case study in user-aided design of online help. *ACM CHI*. 286-287.
21. Lazar, J., Jones, A. and Shneiderman, B. (2006). Workplace user frustration with computers: An exploratory investigation of the causes and severity. *Behaviour and Information Technology*. 25(3):239-251.
22. Liu, T.-Y., Huang, Y.-C. and Chen, W.-C. (2001). A novel algorithm for real-time full screen capture system. *Multimedia Signal Processing*. 395-400.
23. Mack, R. L., Lewis, C. H. and Carroll, J. M. (1983). Learning to use word processors: problems and prospects. *ACM Trans. on Inf. Systems*. 1(3):254-271.
24. Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.
25. Palmiter, S. and Elkerton, J. (1991). An evaluation of animated demonstrations of learning computer-based tasks. *ACM CHI*. 257-263.
26. Palmiter, S. and Elkerton, J. (1993). Animated demonstrations for learning procedural computer-based tasks. *Human.-Computer Interaction*. 8(3):193-216.
27. Palmiter, S., Elkerton, J. and Baggett, P. (1991). Animated demonstrations vs. written instructions for learning procedural tasks: a preliminary investigation. *Int. J. of Man-Machine Studies*. 34(5):687-701.
28. Plaisant, C. and Shneiderman, B. (2005). Show Me! Guidelines for Producing Recorded Demonstrations. *Visual Languages and Human-Centric Comp*. 171-178.
29. Rettig, M. (1991). Nobody reads documentation. *Communications of the ACM*. 34(7):19-24.
30. Sellen, A. and Nicol, A. (1995). Building user-centered on-line help. *Human-computer interaction: toward the year 2000*. Morgan Kaufmann Publishers. 718-723.
31. Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *Computer*. 16(8):57-69.
32. Shneiderman, B. (1997). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Longman Publishing Co., Inc.
33. Spannagel, C., Girwidz, R., Lothe, H, Zendler, A. and Schroeder, U. (2008). Animated demonstrations and training wheels interfaces in a complex learning environment. *Interacting with Computers*. 20(1):97-111.
34. Sukaviriya, P. (1988). Dynamic construction of animated help from application context. *ACM UIST*. 190-202.
35. Sukaviriya, P., Isaacs, E. and Bharat, K. (1992). Multimedia help: a prototype and an experiment. *ACM CHI*. 433-434.
36. Terry, M. and Mynatt, E. D. (2002). Side views: persistent, on-demand previews for open-ended tasks. *ACM UIST*. 71-80.
37. Tuck, R. and Olsen, D. R. (1990). Help by guided tasks: utilizing UIMS knowledge. *ACM CHI*. 71-78.
38. Zeleznik, R. C., Bragdon, A., Liu, C.-C. and Forsberg, A. (2008). Lineogrammer: creating diagrams by drawing. *ACM UIST*. 161-170.