Ambient Help

Justin Matejka, Tovi Grossman, George Fitzmaurice
Autodesk Research
210 King St. E., Toronto, Ontario, Canada
{firstname.lastname}@autodesk.com

ABSTRACT

In this paper we present Ambient Help, a system that supports opportunistic learning by providing automatic, context-sensitive learning resources while a user works. Multiple videos and textual help resources are presented ambiently on a secondary display. We define and examine a collection of design consideration for this type of interface. After describing our implementation details, we report on an experiment which shows that Ambient Help supports finding more helpful information, while not having a negative impact on the user's productivity, as compared to a traditional help condition.

Author Keywords
Help, Learnability, Understanding, Ambient System

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI)

General Terms
Design, Human Factors

INTRODUCTION

Despite advancements in research making complex applications easier to operate, learning how to use these programs can still be a problem. There are often thousands of commands or functions to choose from, and many more ways to combine them into workflows. When using an unfamiliar area of the program, even expert users may require assistance [21].

For most large applications, many hours are spent creating an exhaustive set of documentation for the program's features. These references can sometimes be helpful for finding information about a particular command, but material on how the commands work together or how to perform higher level tasks are often lacking. Even though these standard help systems have improved in recent years, many people are reluctant to use them [37]. Some have found past help systems ineffective and refuse to give the new version a chance, while others would rather continue working and attempt to figure out what they need on their own [13].

Seeing other people work can sometimes be more valuable than traditional help resources [45]. Videos can serve as a reasonable approximation of watching another person's screen, and at the time of writing there are over 350,000 videos on YouTube for the query "Adobe Tutorial" alone. However, all of these videos run the risk of being underutilized since they are not available within the application so users are unaware of their existence or have difficulty finding them. ToolClips [20] show videos in application tooltips, but they are only available for help at the tool-level and don't always show higher level taskflows which utilize the tool. Also, users are still required to explicitly access the videos by calling up the tooltip.

Opposite to the traditional "pull" model where a user explicitly asks for help resources, an implicit "push" model for help has been explored [26]. These systems potentially provide help in situations where the user didn't even know they needed it, such as awareness of unknown functionality or transitioning from a novice workflow to a more expert one. Such learning could be referred to as "unintentional" learning [42] since the user has learnt something without even knowing there was something they needed to know. Previous attempts such as Microsoft's Clippy [26] and "Tip of the Day" have not been ambient, requiring user input on the system's terms, not the user's, and as such have been considered frustrating and ineffective.

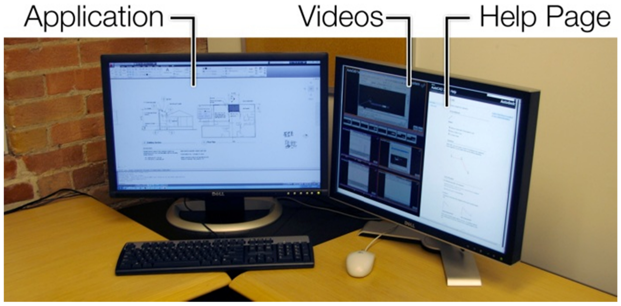


Figure 1. Structure of Ambient Help system. Main application monitor on the left, dynamically updating ambient help display on the right containing 5 videos and a web help page.

In the paper we present Ambient Help, a system that supports opportunistic learning by providing automatic, context-sensitive learning resources while a user works. Multiple videos and textual help resources are presented ambiently on a secondary display [Figure 1]. Opportunistic learning is supported during natural breaks in the user's workflow since the information is preloaded for convenient viewing. This design reduces the transaction cost traditionally associated with obtaining help information.

After reviewing related work, we discuss design considerations for an ambient help system, present our implementation, and report on a study that shows our system supports finding more useful information, while not having a negative impact on the user's productivity, compared to a traditional help system.

RELATED WORK

Contextual and Video Based Help

Since the outset of on-line help, there have been explorations into contextual [2,12,29] and video-based [5, 27,29] delivery, with less research exploring both [20,1]. Shneiderman [44] argues that graphical demonstrations can be the most direct way for novices to learn procedural knowledge. However, its implementation was expensive and time consuming when the ideas were first proposed in the early 90's [27], and in general, on-line documentation was "crippled" by hardware and software limitations [15]. The evolution of CPU power and video hosting services has made video documentation a viable possibility [31].

In terms of publishing, it is often easier to record a 5 minute tutorial video than it is to take the necessary screenshots and write up an easy to follow and useful tutorial [39]. However, with text based help a quick glance might provide the needed assistance, where a video may require a longer viewing period to get the same information [16]. It has also been found that some users naturally prefer videos while other prefer text-based help [20].

Despite psychology research which exposes the benefits of contextual assistance [3], help systems are generally offered through completely separate components. This discontinuity can result in delayed, disruptive, inconsistent, and obtrusive help systems [2,30].

Previous research in animated assistance [23,38,1] has explored contextual demonstrations. One good example is Google Sketchup which has an "instructor" panel providing tool-based contextual animated assistance whose content changes as the user switches tools. While their assistance consists of very basic animations which only illustrate how the tools function, we use full motion videos which show how the tools can be used in a workflow.

Ambient Information Systems and Displays

Ambient and peripheral displays [8,35,40,46] support the monitoring of noncritical information with the intent of not distracting or burdening the user. One typical example of an ambient display is Portholes [10] which periodically takes a video snapshot of members of a community and presents them as a grid of images for each user to view and have an awareness of who is in and available for communication.

Pousman and Stasko [41] describe four dimensions of consideration when designing an ambient display: information capacity, notification level, representational fidelity, and aesthetic emphasis. In our approach, we wish to present contextual learning resources in an ambient display where the information capacity and representational fidelity is high and always available to be absorbed by the user during breaks of work activity. Also, we do not interrupt the user, so the notification level is low.

Evaluation of the usefulness of ambient displays is often challenging [24,33,35,43]. The evaluation criteria typically centers around noticeability, comprehension, relevance, division of attention, and engagement. While these are all important, we are particularly interested in increasing the temptation to learn something new, perhaps serendipitously. This is similar to over the shoulder learning [45] where users can see how other users work.

One concern with an information rich ambient display is that it may induce information overload [28]. While this could be true, the very nature of our ambient display is that it contains optional information that can be ignored during intense work or during high levels of information processing in the main display.

Buxton describes a framework for modeling systems which factor in not only human-human and human-computer interaction but a second dimension of foreground and background activity [7]. Here, foreground activities are in the "fore of human consciousness – intentional activities" such as typing in a computer. Background activities take place "in the periphery 'behind' those in the foreground" such as being aware of someone in the next office typing. Our intent is that Ambient Help easily transitions between background and foreground activity.

Interruptability and Notification

A great deal of research has studied technology-mediated interruptions and notification of system or application events [1,17,22,36]. The concern is that interruptions can be an annoyance, adversely affect a user's task completion time, and make them vulnerable to mistakes. Thus, researchers have explored ways to mitigate the problems by deferring interruption until users are more open to the information during their work. Attentive user interfaces attempt to recognize some basic user states (presence, head position, mouse activity) and act differently according to user behavior [14]. Our goal is not to interrupt the user, particularly during their intense work.

Transaction cost

A key goal for our system is to reduce the transaction cost of a user accessing help and learning resources while working. Research has explored the dynamics of switching between work and help [4] and shown that small interaction improvements can cause categorical behavior changes that far exceed the benefits of decreased task times [6,18].

The Cocoa Buzz system [11] runs on a user's extra display, or a large shared display, and automatically cycles through different information sources (such as news sites) customized by the user. About every minute it updates and generates a new information collage. Our system differs in that the learning resources are contextually retrieved.

Instead of a user explicitly entering in search terms, implicit and activity based web searching is being researched. Maekawa et al. [32] have explored *query free* web searching based on activity detected by sensors and internet-connected domestic appliances. For example, while washing a coffee maker, a web page is automatically retrieved on how to clean it using vinegar. Henzinger et al. [25] automatically select web pages that a user might want to see which are relevant to a TV program they are currently watching. This implicit style of searching system matches our approach but differs in that we focus on software learnability [21].

DESIGN CONSIDERATIONS

In the following we look at a number of design goals we considered while creating the Ambient Help system to be as useful as possible, while not being disturbing or distracting.

Ambient Nature

A primary consideration is for the help content to be displayed ambiently; that is, in an always-on display which does not require user input to function. In this way, the time-cost to access the help content is reduced since user interaction is not required to call up the information. Since the display is persistent, the user can look over for help information at any time, and it should be designed in a way so there is no cost associated with *not* using the display; i.e., it can be easily ignored. Similarly, to minimize distraction the system should never interrupt the user.

Relevance of Content

For an ambient system to be useful, it must be able to find and display content which is relevant to the user. This is different from Tip of the Day where you have small pieces of help information, or standard help systems which give access to more content, neither of which use any knowledge of the user's context. Besides attempting to show help content which is a good match for what the user is looking for, it is also useful to present generally useful or interesting content to enhance the variety of the system and provide the opportunity for exploration and serendipitous learning [34].

Transition Between Ambient and Interactive

Ideally there should be no explicit switch required to move between ambient and interactive modes of operation. Users should be able to move between the main and help windows freely without concern for the state of the help system.

Multiple Forms of Information

Traditionally, product support materials have been delivered in mostly text, with some diagrams mixed in. A limited number of help videos are now being published by the software manufacturers, but with the recent advancements in the ease of posting videos to the internet (YouTube, etc.) many individuals and small third-party companies have started posting their own tutorial videos. Videos have the benefit of being able to demonstrate the concept being described. Textual resources on the other hand have the benefit of being more easily skimmed or consumed quickly. Besides official product documentation, there are also significant web-based textual resources such as discussion boards and tutorial web pages which many users turn to for information.

With their being benefits to both video and textual help information, we recommend supporting both.

Channels of Information

Ambient display systems can range from displaying a single stream of information up to a great number simultaneously. By displaying only one channel, that single item takes up the entire information bandwidth of the display, and as such, when the user looks over they are assured to see it. With many channels on the same display, each individual channel must be of smaller size and possibly lower fidelity. With more channels the chances are greater that useful content will be displayed. However, if too many things are going on, that useful content might be missed. In the end, designers should ensure a balance between the extremes of too little and too much information.

Location

An ambient help system could be implemented on a secondary monitor or within the primary monitor. In our experience we have found that users are quicker to dismiss items which take space away from their primary application. Instead of using a secondary display on the same computer, the system could also run on a secondary system, such as an Apple iPad or similar device.

Content Variance

If the content of the ambient display never changes, users will stop looking over because they have already seen the information. On the other hand, if the content changes too often it could be a distraction, or relevant information could be swapped out while the user is trying to view it. As such, there should be a balance between updating too slowly leaving the information stagnant, and updating so frequently that the content doesn't have time to be used.

SYSTEM DESCRIPTION

Overview

Given these design considerations we now present our exploration into offering ambient help. The Ambient Help system is displayed on a secondary monitor placed beside the main monitor [Figure 1]. The user sits directly in front of the main application monitor with the secondary monitor angled off to the side, keeping the application in the center of focus while the ambient display is visible in the periphery. The system automatically presents help content relevant to the commands the user is working with.

Ambient and Interactive Modes

On the left side of the screen are relevant help videos, and on the right side is a web page [Figure 2]. We display closed captioning information as an overlay for videos with the information available making it possible to follow along without the audio playing. We choose to use the command reference pages as the source for the web pages since there is a page for each command with consistent formatting, making it easier for users to find information, and making for a smoother transition between pages.

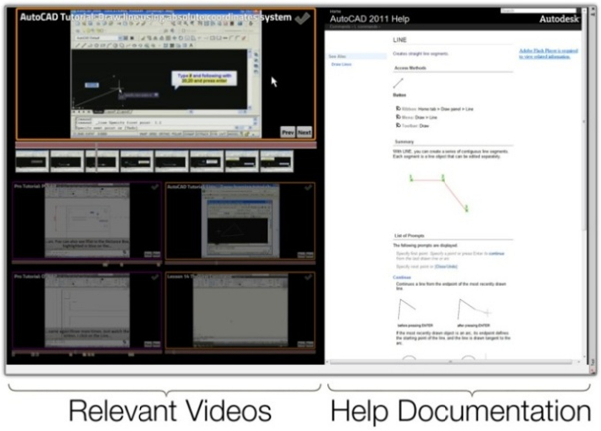


Figure 2. Ambient Help interface.

Our system reduces potential distractions by muting the audio and dimming the videos while the cursor is over the main application. We also reduce the frame rate at which videos are shown (only changing frames once every 5 seconds, with a 1 second smooth fade between frames). Being connected to the main application allows the system to use the application context to show relevant help content without specific input from the user.

When the cursor is within the help window, the video with cursor focus is shown at full brightness while the others remained dimmed. By default, when the cursor returns to the application window, the videos return to being dimmed and muted at the reduced frame rate. However, there is the option to "check" a video [Figure 3] which will keep the video playing at a reduced volume when the focus returns to the main application. This allows the user to keep listening to the video while working, and also ensures that the video will play to completion without being replaced.

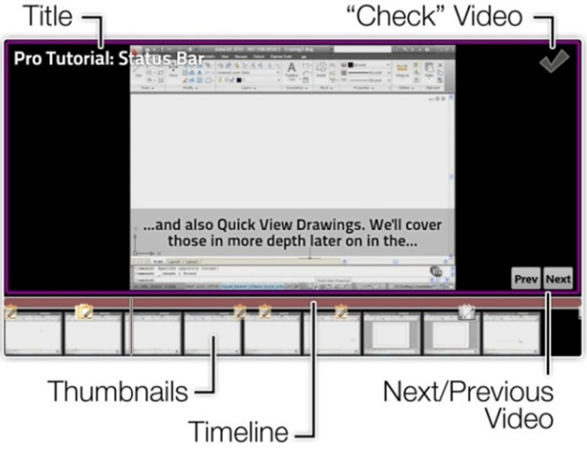


Figure 3. Elements of the Ambient Help video player.

In our system, the display may only change at the beginning of a new command, and even then, it may remain stable if the same command has been issued multiple times in succession, or the currently playing videos are sufficiently relevant. Between command issuances the content remains constant so users are able to read the help page or watch the videos without fear of them being replaced.

Based on the channels of information design consideration, we chose to show 5 videos. One video resides in the top "primary" position (at 970 by 536 resolution) and 4 smaller videos (at 490 by 278) below [Figure 2]. At this size we feel even the small videos are large enough for the user to get a sense of what is going on in the video while still allowing us to display a number of different clips to increase the chances of presenting a useful one. While the user is working and the display is updating, the video determined to be the most relevant is automatically transitioned up to the primary position, in theory placing the most useful video in the top, largest, position. The user can also manually promote a smaller video to the top by clicking on it, making it larger and easier to see the details.

If a video the user did want to watch happens to be replaced, it can be retrieved by clicking the "Prev" button in the video player. The interface also includes a timeline and thumbnails which allow navigation to different points in the video, and a button to move to the next video selected by the relevance algorithm described later.

AutoCAD Plug-in

The Ambient Help system was implemented as an AutoCAD plug-in using the ObjectARX framework. The system hooked into the `CommandStarted` messages from AutoCAD to get a notification whenever a user clicked on, or otherwise started to use a command. With these notifications the plug-in tracked which commands the user had recently used and was currently using. The videos came from two sources: a local cache of videos obtained from a YouTube search for "AutoCAD Tutorial" (1,379 videos), and an AutoCAD training DVD (107 videos) [Figure 2].

Automatic Video Analysis

Ideally, all tutorial videos ever created would be marked up with Chronicle-like [19] metadata indicating where in the video commands occurred, settings were changed, layers were modified, etc. Hopefully applications will someday allow users to record videos with this type of metadata, but for now, nearly all video help content is untagged.

Knowing which commands happened where allows us to find more germane videos. Additionally, we can seek directly to positions in the video to display relevant content, thus overcoming the need to manually scrub the video looking for command invocations. In the video timeline we display markers for the commands [Figure 4] that make it easier for users to navigate to parts of the video where commands of interest were used.

As a proof-of-concept to show one way in which this command scraping process could be automated, we created a program to analyze our collection of 107 professionally produced AutoCAD tutorial videos and generate the command stream tags for each. Limiting our source material for this process allowed us to make the simplifying assumptions that all videos would have the UI at the same scale, and the elements would be in the same positions. A system such as Sikuli [47] or Prefab [9] could be used to improve the algorithm to locate and recognize UI elements.

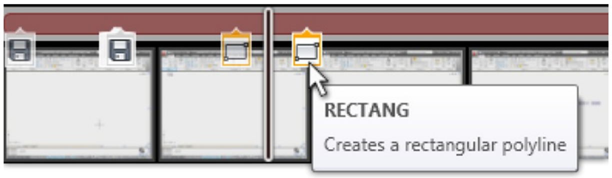
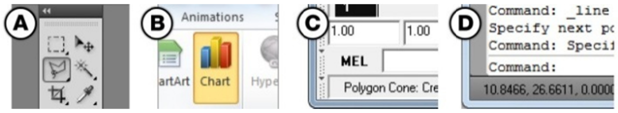


Figure 4. Video timeline showing command markers.

Possible Approaches for Detecting Commands

Different applications support unique ways of visually determining which tool is active or which command has just been issued. For example, in Adobe Photoshop you can tell the active tool by looking at which tool button is rendered darker than the others [Figure 5a]. This information is persistent and available for the entire time the tool is active. The buttons of many products become highlighted to one colour when the mouse if over, and another darker colour when they are clicked [Figure 5b]. Some products such as Autodesk Maya have a status bar indicating which tool or menu item the cursor is over [Figure 5c]. By combining this information with when the mouse button was clicked (perhaps from the audio track), one could determine which tools and menu items were used.



```
Command: _line
Specify next pc
Command: Specif
Command:
10.8466, 26.6611, 0.000(
```

Figure 5. Example avenues for determining tool and command usage in: (A) Adobe Photoshop, (B) Microsoft Office, (C) Autodesk Maya, (d) Autodesk AutoCAD.

Commands in AutoCAD can be entered in multiple ways: clicking on a button, typing into the command line, issuing a shortcut, entering a command alias, or issued from an automated script. However, regardless of how the command is issued, all command history shows up in the command line window [Figure 5d] creating a good option for tracking command usage. It is possible to close the command line window in AutoCAD, but most users leave it open, and it is open in all of the videos in our sample collection.

Optical Character Recognition (OCR)

In the AutoCAD command line history, a line of the format `Command:name` indicates that the command "name" has been issued. We started by capturing individual image frames with the OpenCV toolkit. We cut out areas of the image for each of the 4 visible text lines and attempted to run some standard OCR packages (ABBYY FineReader, SimpleOCR, ocropus, and others) to extract the text. Since most OCR packages are tuned to recognize printed and then scanned text, they performed quite poorly on the text in our video frames—generally recognizing less than 50% of the characters correctly. This is consistent with previous work [47] where it appears that most of the text was not recognized correctly. To improve on this, we rendered the alphabet of characters using the same font and anti-aliasing settings as a default installation of AutoCAD, scaled to the same amount as the text in the video. Then, we were able to step along the line comparing each character in the image to our dictionary of rendered characters. Enlarging the source image and moving it around slightly to account for sub-pixel rounding, then using a squared difference comparison of the pixel values let us find the matching character with greater than 99% accuracy. This, combined with a dictionary of the commands in AutoCAD let us achieve 100% accuracy in command recognition.

Video Relevance

At the beginning of each command, a relevance score, R, was calculated for all available videos weighting the current command most highly and weighting the previous five commands with a decreasing power function:

$$R(v) = \sum_{i=1}^{5} S(v, c_i) \times (6 - i)^2$$

Each video v's relevance score for a given command c, $S(v,c)$ was calculated using the following formula:

$$\begin{aligned} S(v,c) = {} & v.title(c) \times 10 \\ & + v.keywords(c) \times 8 \\ & + v.description(c) \times 7 \\ & + \sum(v.comment(c) \times 6 + 0.02) \\ & + v.commands(c) \times 12 \end{aligned}$$

where $v.title(c)$ is 1 if the title of the video contained the command name c, otherwise it is 0. Similarly, $v.keywords(c)$, $v.description(c)$, $v.comment(c)$ and $v.commands(c)$ operate on the keywords, description, comments and command stream (if available) respectively. For each user comment the video had, 0.02 points were added to account for generally popular videos. In this model, having the command used in the command stream is worth more than having the command name in the title, which is in turn more valuable than having the command mentioned in the keywords, description or comments. If a video is already playing in the display, its score is multiplied by a factor starting at 1.2 which decreases linearly down to 0.0 after 5 minutes. This gives videos that have just started playing priority, and then makes it easier for videos to be replaced after they have been playing for some time. Videos among the most recent 50 to be viewed are excluded from the calculations to support variation in the selections. The videos with the highest relevance scores are selected to be played when a new command is started.

This is just one rating scheme, which we have found works well with our mixed collection of YouTube and professionally authored tutorial videos.

EXPERIMENT

We wanted to evaluate Ambient Help to see if users, in work-like situations, could derive some instructional value from the system. In addition, the study allowed us to look at distraction and productivity compared to traditional help.

Participants

Twelve paid volunteer participants (8 male, 4 female) between the ages of 23 and 30 were recruited through an email list of AutoCAD users. To be eligible the users were required to be proficient with AutoCAD and were asked how long they had been using the program and how often they typically use it, with results ranging from 4 to 10 years, and 3 to 7 days/week respectively. All participants had used AutoCAD in a professional setting as well as receiving formal classroom training with the software.

Apparatus

The experiment was conducted in an enclosed office using a 3.16GHz quad-core desktop computer running Windows 7 64-bit Edition. The graphics card was a NVIDIA Quadro FX 5600 and was running two 24” Dell monitors each at a resolution of 1920 by 1200. Participants were seated with their chair and keyboard directly in front of the main monitor with the second monitor angled and positioned to the right [Figure 1].

Design

A repeated measures within-participant design was used with the independent variables being *task* (*intense*, *casual*) and *help system* (*ambient*, *manual*). The ordering of *task* and *help system* were counterbalanced. Each of the four trial conditions lasted 15 minutes. During each trial the statistics the experimenter judged and counted were: number of videos watched, number of help pages viewed, the number of useful/interesting items found, and in which data stream they were found in.

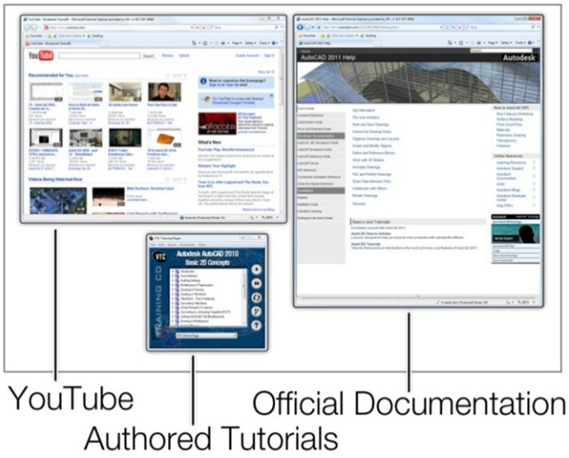


Figure 6. Setup of manual help condition.

Help Systems

The system described in this paper was used as the *ambient* condition. As a baseline comparison, the *manual* help system consisted of three windows: YouTube and the AutoCAD 2011 online documentation webpages, and the interface to an AutoCAD professional training DVD [Figure 6]. In this way the manual condition had all of the same material as the ambient help system, but was not integrated into AutoCAD and the content would need to be manually found by the participants.

Tasks

The *intense* and *casual* tasks were created to simulate two types of working conditions: those where the user is very focused on the task and cannot afford to be distracted (for instance, they are working towards a very tight deadline), and those where the user is working, but not in any particular rush (for example, they might spend some extra time browsing the internet or talking to colleagues).

For the *intense* task, users were given an AutoCAD document with a graphic underlay of 14 kitchen cabinet designs [Figure 7]. They were instructed to recreate these cabinets as quickly and accurately as possible. As motivation, the participant who completed the most cabinets was to be awarded an extra $100 gift card.

The task was designed to be simple enough that most experienced AutoCAD users would be able to draw the cabinets without much trouble. The cabinets themselves were also designed to each be of similar complexity so we could count and compare the user’s cabinet drawing performance in both help system conditions.

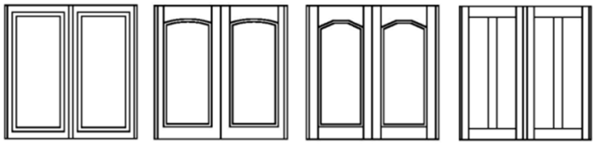


Figure 7. Portion of the intense task document.

For the *casual* task, participants were given a set of blueprints for a house and told to modify or extend the drawings in any manner they chose. Some examples given were to add a patio or some landscaping elements. The only requirement was that they use 3 commands from each of two command lists: *basic* commands, and *rare* commands. The *basic* list contained 20 of the most frequently used AutoCAD commands such as LINE, ERASE, and RECTANG, which are commands that all AutoCAD users would know. The *rare* list contained a collection of 10 commands which are among the 100 most frequently used commands, but are used by a small (<5%) percentage of the user population. The idea being that these would be useful commands that the participants were not familiar with. To ensure the commands selected from this list were new to the user, they were instructed to only choose commands from this list they had not used before. This allowed us to test the help systems when users may explicitly need help and also when they are doing exploratory learning.

Procedure

Upon arrival the participants were told that we were testing to see if *“a secondary display of information could be helpful to people while using AutoCAD”*. It was also explained that we would be presenting the help content in two different ways and we would be looking to see if they found any information that was useful or interesting to them. They were encouraged to *"glance at, or interact with, the content on the secondary monitor"* as much as they wanted to, and if they found anything useful or interesting, to let us know so we could keep a record. If during the study the examiner saw the participant looking at the secondary monitor for 10 seconds, he would prompt the user to explain what they were looking at and if they found anything useful.

Results

Because the two tasks were so different, separate analysis was performed for each of the *task* conditions.

Useful Items Found

To assess the effectiveness of the Ambient Help system for providing useful information to the user, we looked at the number of useful or interesting items found in each condition [Figure 8]. For the *casual* task, repeated measure analysis of variance showed a main effect for *help system* ($F_{1,11}$ = 12.6, p < .005) with means of 3.25 items for *ambient* and 1.25 items for *manual*, a 2.6 times increase. In the *intense* task the means were 0.92 items for *ambient* and 0.17 for *manual*. This difference however did not reach significance ($F_{1,11}$ = 2.83, p = .121).

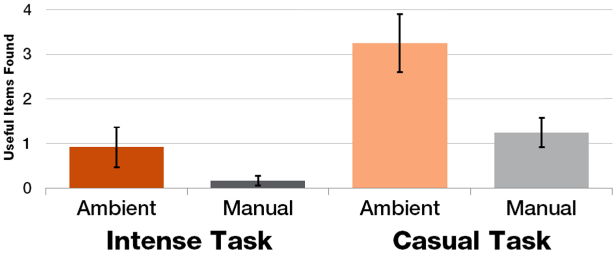


Figure 8. Average number of useful items found in each of the four task/help system conditions. (Note: Error bars report standard error).

With the cursor tracking data we can also look at how much time the users spent on average in AutoCAD and on the second monitor for each condition [Figure 9]. It is interesting to note that users found 2.6 times as many useful items while using the Ambient Help system in the *casual* task although they spent about the same amount of time on the second display for both help system conditions (17.7% and 18.4%). This result is consistent with the goal of the Ambient Help system to have users find more useful items, and they are able to do that without spending any more time than they would in a traditional manual help situation.

Impact on Performance

As a measure of performance in the *intense* task, we looked at the number of cabinets completed with each help system [Figure 10]. The mean results were 7.88 for *ambient* and 8.63 for *manual* with standard deviations of 4.54 and 3.73 respectively. The analysis of variance did not show the difference to be significant ($F_{1,12}$ = 0.974, ns) supporting our belief that Ambient Help would not have a negative impact when the user is immersed in an intensive task.

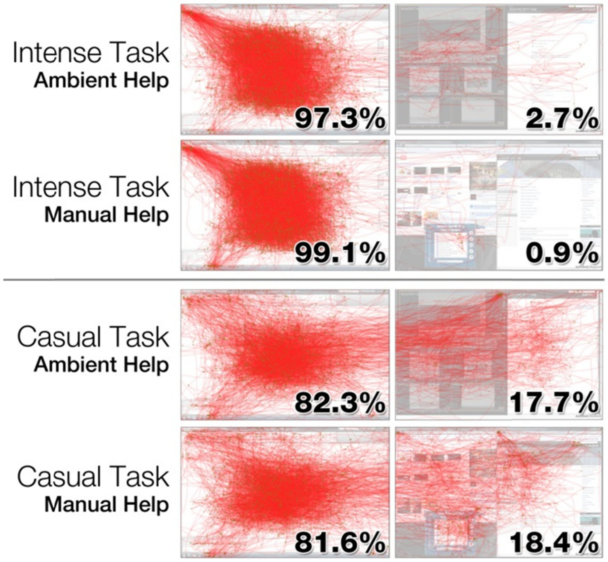


Figure 9. Cursor movement for each condition of all participants. Percentages show fraction of time spent on each monitor.

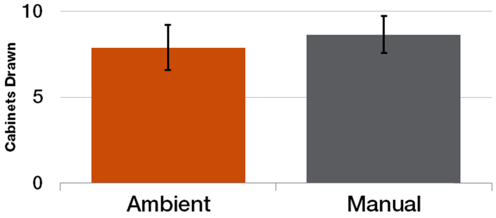


Figure 10. Number of cabinets drawn in the intense task. (Note: error bars report standard error)

After each condition the participants were asked to rate some statements on a Likert scale (between 1-Strongly Disagree, and 7-Strongly Agree). One of the statements was *"The content of the extra monitor was helpful"* [Figure 11]. The results have been aggregated into three categories: Disagree (1, 2), Neither (3, 4, 5) and Agree (6, 7). The majority of users (8 of 12) said they found the *ambient* system helpful in the *casual* task compared to only 1 user who found the *manual* system helpful.

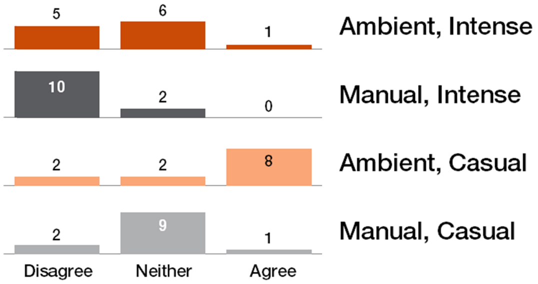


Figure 11. Results for *"The content of the extra monitor was helpful"*.

Participants were also asked if they *"found the content of the extra monitor distracting"*. In both the *intense* and *casual* tasks, 3 of 12 users agreed this was the case for the ambient help system [Figure 12].

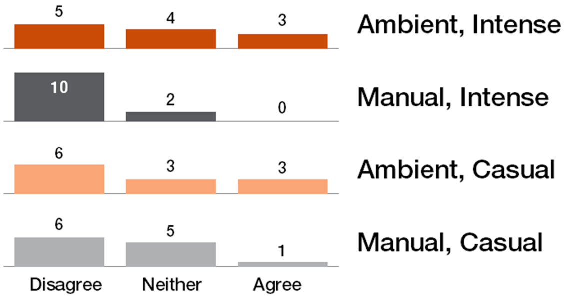


Figure 12. Results for *"The content of the extra monitor was distracting"*.

While not a longitudinal study, if we look at each user's rating of the distraction during the first task they used the ambient help interface compared with their second, 5 users said the interface was distracting during their first task, compared to only 1 during the second task [Figure 13].

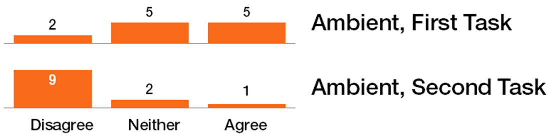


Figure 13. Results for *"The content of the extra monitor was distracting"* during the ambient help condition in first vs. second task.

With these results, we believe that many users would not find the interface distracting given time to get used to it.

Besides distraction, participants were also asked if the *"content of the extra monitor hindered their productivity"* [Figure 14]. Few users felt that either help system hindered their productivity in either task. This is of particular importance in the *intense* task where the participants were trying to be as productive as possible. It is promising that for this task the *ambient* and *manual* systems had essentially the same subjective rating, suggesting that even when not in use, the ambient display is no more of a hindrance to a user than a monitor showing static content.

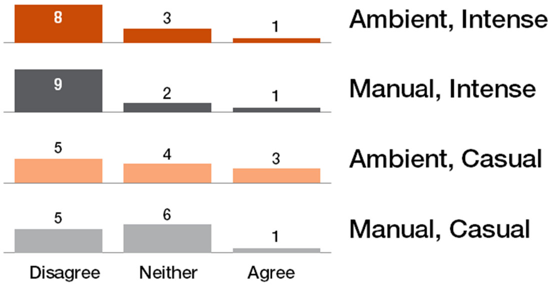


Figure 14. Results for *"The content of the extra monitor hindered my productivity"*.

Content Sources

For each of the two content sources (*videos*, and *help pages*), the number of items viewed was significantly higher in the *ambient* than the *manual* help condition [Figure 15]. For *videos* the means were 4.08 and 1.67 for *ambient* and *manual* respectively ($F_{1,11}$ = 5.04, $p < .05$) and for *help pages* the means were 4.83 and 2.83 ($F_{1,11}$ = 8.25, $p < .05$). It is important to note that here "viewed" means that the participant spent some amount of time actually looking at the content, not that it was simply shown on the screen.

The number of useful items found was also higher using the ambient help system for both *videos* (1.0 vs. 0.5) and *help pages* (3.17 vs. 0.92), however only the difference with *help pages* reached significance ($F_{1,11}$ = 7.44, $p < .05$). Overall the ratio of useful items found to items viewed is significantly higher for *help pages* than for *videos* (0.448 vs. 0.184: $F_{1,11}$ = 7.56, $p < .05$) as well as for *ambient* help compared to *manual* help (0.425 vs. 0.203: $F_{1,11}$ = 8.17, $p < .05$). We believe this ratio is higher for help pages than videos since the static pages are easier to consume quickly. Several times we saw participants look at a video for 10 to 20 seconds and then move on to the help page because the "video was taking too long". The time it takes to get useful information from a video can hopefully be reduced by more complete tagging of the video content allowing the system to play the video at the point of the relevant information.

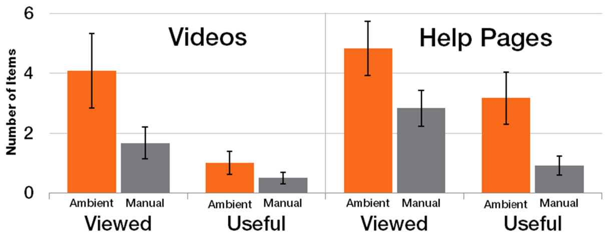


Figure 15. Number of videos and help pages viewed and found useful in each of the help system conditions. (Note: error bars report standard error)

Overall when users were asked if they felt either system could be useful, 7 of 12 agreed the Ambient Help system could be useful compared to 0 for *manual.* [Figure 16]. In fact, one user during the manual help condition remarked *"Oh no, my videos are gone now. I wish I had them back, they would be really useful right now"*.

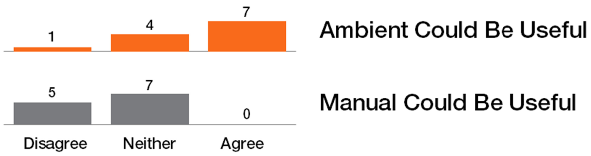


Figure 16. Results for *"A system like <Ambient/Manual> could be useful"*.

DISCUSSION & FUTURE WORK

Our study has shown that the Ambient Help system significantly improved the ability of the user to find useful information during casual work, while generally not objectively, or subjectively, hindering performance during intense tasks. A longer term study, or one performed on the users' own machine in their standard environment would be useful to see how these results hold over a longer period of time and explore the impact of the system on higher-level performance and learning. Given the often negative reactions users have to new help systems and interface designs, it would be interesting to see in what scenarios users would choose to keep the ambient display open.

We measured the number of useful or interesting items that the participants found with the system. It would be valuable to explore the relationship between information that is *interesting* to the user now and the potential it has to become *useful* in the future.

A potential concern with such an active help system is that it may be distracting to users. We found that after their second 15-minute exposure to our system most users did not find it distracting. However, we believe the system could be even less distracting if we choose not to animate the position changes of the videos from the bottom to the primary position while in the ambient mode. There may be other ideas such as pausing the videos completely while over the main application, or updating less frequently in general which would be interesting to explore.

We chose to show five videos and one web-based help page but a study to examine how many channels of information supports the best mix of providing information without the user becoming overloaded would be valuable. We suspect this might come to the finding that different users would like to see different volumes of information and would lead to the question of how to design the interface to support varying amounts of channels.

While we only used two sources of videos and one source of text-based help for this prototype, future work could explore using more sources and how to best integrate them into a single system. Along with more sources a more sophisticated algorithm could be developed for finding relevant content. Additional pieces of context which could be used to improve context matching include: which Ribbon tab is active, which dialogs are open, and the actual content of the document. Although not accessible through the public API, YouTube has the ability to automatically generate closed captioning data for their videos. This text might be a valuable resource for trying to automatically determine what is going on in each of the clips.

By virtue of being an always-on display, the transaction cost to access the help information is already very low. We could decrease the cost of interacting with the content by introducing keyboard shortcuts to perform some of the more common actions such as listening to a video. It would also be nice to reduce the amount of time required for a user to scan a video for useful content. We have thought of introducing some graphic overlays to the videos to more obviously show which commands are being used, and a longer selection of the closed captioning text to serve as a "glanceable summary" so they can be more quickly scanned for useful information.

CONCLUSION

Design considerations for an ambient system that provides help content using a push rather than pull delivery model were discussed. The resulting Ambient Help system offers the potential to increase the user's learning in multiple ways: both "intentional" learning, where the user knew they needed help, and perhaps as importantly, "unintentional" learning, where the push model presents information the user wouldn't even know to look for. In the casual task, which simulates what we envision to be the primary environment for using this type of system, Ambient Help led to 2.6 times as many useful insights as the control. Besides being helpful during casual tasks, users generally agreed that the system did not negatively impact their productivity during intense work and this result was supported by the quantitative data. Although implemented in AutoCAD, we believe Ambient Help is a general solution which could be made to work with almost any software application where learning is an important issue.

ACKNOWLEDGEMENTS

We thank Ramtin Attar for designing the AutoCAD tasks.

REFERENCES

1. Adamczyk, P. D., Iqbal, S. T., and Bailey, B. P. (2005). A method, system, and tools for intelligent interruption management. In *Proceedings of the 4th International Workshop on Task Models and Diagrams.* 123-126.
2. Ames, A. L. (2001). Just what they need, just when they need it: an introduction to embedded assistance. *ACM SIGDOC*. 111-115.
3. 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.
4. Andrade, O., Bean, N., and Novick, D. (2009). The macro-structure of help. *ACM SIGDOC 2009.* 143-150.
5. Baecker, R. (2002). Showing instead of telling. *ACM SIGDOC*. 10-16.
6. Brandt, J., Dontcheva, M., Weskamp, M., and Klemmer, S. R. (2010). Example-centric programming: integrating web search into the development environment. *CHI*, 513-522.
7. Buxton, W. (1995). Integrating the periphery and context: A new taxonomy of telematics. *GI*, 239-246.
8. Cadiz, J. J., Venolia, G., Jancke, G., and Gupta, A. (2002). Designing and deploying an information awareness interface. *ACM CSCW '02*, 314-323.
9. Dixon, M. and Fogarty, J. (2010). Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. *ACM CHI,*1525-1534.
10. Dourish, P. and Bly, S. (1992). Portholes: supporting awareness in a distributed work group. *CHI*, 541-547.
11. Eagan, J. R. and Stasko, J. T. (2008). The buzz: supporting user tailorability in awareness applications. *ACM CHI*. 1729-1738.
12. Farkas, D. K. (1993). The role of balloon help. *Journal of Computer Documentation*. 17(2):3-19.
13. Fitzmaurice, G., Matejka, J., Mordatch, I., Khan, A., and Kurtenbach, G. (2008). Safe 3D navigation. *I3D*, 7-15.

14. Fosso, F. and Porta, M. (2009). A vision-based attentive user interface with (semi-)automatic parameter calibration. *Proceedings CompSysTech '09*, vol. 433. 1-6

15. Goodall, S. D. (1992). Online help: a part of documentation. *Systems Documentation*. 169-174.

16. Grabler, F., Agrawala, M., Li, W., Dontcheva, M. and Igarashi, T. (2009). Generating photo manipulation tutorials by demonstration. *ACM SIGGRAPH*. 66.

17. Grandhi, S. and Jones, Q. (2010). Technology-mediated interruption management. *Int. J. Hum.-Comput. Stud*. 68, 5 (May. 2010), 288-306.

18. Gray, W. D. and Boehm-Davis, D. A. (2000). Milliseconds Matter: An Introduction to Microstrategies and to Their Use in Describing and Predicting Interactive Behavior. *Journal of Experimental Psychology: Applied*, 6 (4), 322-335.

19. Grossman, T., Matejka, J., and Fitzmaurice, G. (2010). Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. *ACM UIST'10*.

20. Grossman, T. and Fitzmaurice, G. (2010). ToolClips: an investigation of contextual video assistance for functionality understanding. *ACM CHI '10*. 1515-1524.

21. Grossman, T., Fitzmaurice, G., and Attar, R. (2009). A survey of software learnability: metrics, methodologies and guidelines. *ACM CHI*, 649-658.

22. Hameed, S., Ferris, T., Jayaraman, S. and Sarter, N. (2009). Using Informative Peripheral Visual and Tactile Cues to Support Task and Interruption Management. *Human Factors*, 51 (2). 126-135.

23. Harrison, S. M. (1995). A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. *CHI*. 82-89.

24. Hazlewood, W. R., Connelly, K., Makice, K., and Lim, Y. (2008). Exploring evaluation methods for ambient information systems.*CHI Ext. Abstracts*, 2973-2978.

25. Henzinger, M., Chang, B., Milch, B., and Brin, S. (2005). Query-Free News Search. *ACM WWW*, 101-126.

26. Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *ACM SIGCHI*. 159-166.

27. Jackson, S. (2001). Editing computer hardware procedures for multimedia presentation. *SIGDOC*. 68-72.

28. Karr-Wisniewski, P. and Lu, Y. (2010). When more is too much: Operationalizing technology overload and exploring its impact on knowledge worker productivity. *Comput. Hum. Behav*. 26, 5 (Sep. 2010), 1061-1072.

29. Kelleher, C. and Pausch, R. (2005). Stencils-based tutorials: design and evaluation. *ACM CHI*. 541-550.

30. Knabe, K. (1995). Apple guide: a case study in user-aided design of online help. *ACM CHI*. 286-287.

31. 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.

32. Maekawa, T., Yanagisawa, Y., Sakurai, Y., Kishino, Y., Kamei, K., and Okadome, T. (2009). Web searching for daily living. *ACM SIGIR'09*, 27-34.

33. Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays. In *ACM CHI '03*, 169-176.

34. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. (2009). CommunityCommands: command recommendations for software applications. *UIST* 193-202.

35. Matthews, T. (2006). Designing and evaluating glanceable peripheral displays. *DIS '06*. ACM, 343-345.

36. McFarlane, D. C. and Latorella, K. A. (2002). The scope and importance of human interruption in human-computer interaction design. *Hum.-Comput. Interact*. 17, 1 (Mar. 2002), 1-61.

37. Novick, D. G. and Ward, K. (2006). Why don't people read the manual? *ACM international Conference on Design of Communication*. 11-18

38. Palmiter, S. and Elkerton, J. (1993). Animated demonstrations for learning procedural computer-based tasks. *Human.-Computer Interaction*. 8(3):193-216.

39. Plaisant, C. and Shneiderman, B. (2005). Show Me! Guidelines for Producing Recorded Demonstrations. *Visual Languages and Human-Centric Comp.* 171-178.

40. Plaue, C. and Stasko, J. (2007). Animation in a peripheral display: distraction, appeal, and information conveyance in varying display configurations. *GI*, 135-142.

41. Pousman, Z. and Stasko, J. (2006). A taxonomy of ambient information systems: four patterns of design. In *ACM AVI '06*, 67-74.

42. Sawyer, R.K. (Ed.), The Cambridge Handbook of the Learning Sciences. Cambridge University Press, 2006.

43. Shen, X., Moere, A. V., Eades, P., and Hong, S. (2008). The long-term evaluation of Fisherman in a partial-attention environment. In *ACM Beyond Time and Errors: Novel Evaluation Methods For information Visualization, BELIV '08*, 1-6.

44. Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *Computer*. 16(8):57-69.

45. Twidale, M. B. (2005). Over the Shoulder Learning: Supporting Brief Informal Learning. *CSCW*, 505-547.

46. Vogel, D. and Balakrishnan, R. (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. *ACM UIST 2004*, 137-146.

47. Yeh, T., Chang, T., and Miller, R. C. (2009). Sikuli: using GUI screenshots for search and automation. *ACM UIST*. 183-192.

48. Zeleznik, R. C., Bragdon, A., Liu, C.-C. and Forsberg, A. (2008). Lineogrammer: creating diagrams by drawing. *ACM UIST*. 161-170.