ReMap: Lowering the Barrier to Help-Seeking with Multimodal Search

C. Ailie Fraser[1, 2], **Julia M. Markel**[2], **N. James Basa**[2], **Mira Dontcheva**[1], **Scott Klemmer**[2]
Adobe Research[1]; Design Lab, UC San Diego[2]
{fraser, mirad}@adobe.com; {jmarkel, nbasa, srk}@ucsd.edu

ABSTRACT

People often seek help online while using complex software. Currently, information search takes users' attention away from the task at hand by creating a separate search task. This paper investigates how multimodal interaction can make in-task help-seeking easier and faster. We introduce ReMap, a multimodal search interface that helps users find video assistance while using desktop and web applications. Users can speak search queries, add application-specific terms deictically (*e.g.*, *"how to erase this"*), and navigate search results via speech, all without taking their hands (or mouse) off their current task. Thirteen participants who used ReMap in the lab found that it helped them stay focused on their task while simultaneously searching for and using learning videos. Users' experiences with ReMap also raised a number of important challenges with implementing system-wide context-aware multimodal assistance.

Author Keywords

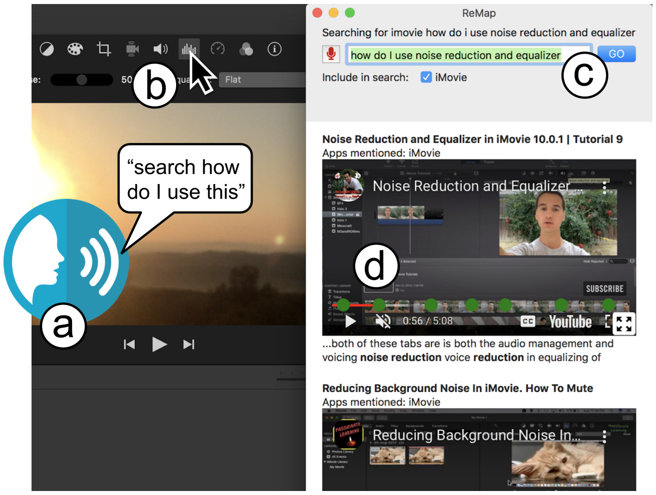
multimodal search; speech; deixis; contextual search

CCS Concepts

•Human-centered computing → Graphical user interfaces; Sound-based input / output;

INTRODUCTION: MAKING SEARCH MORE NATURAL

Help-seeking is often tedious and difficult. Searching for help requires switching mental context, visual attention, and input focus away from the task at hand. The user must first articulate a search query that will match the resources they seek. This can be prohibitively difficult for novices, who often don't have enough domain knowledge to know *what* to ask, let alone how to ask it [25, 32]. Then, once the user finds a help resource, they must switch their attention back and forth between the resource and their task to follow along with instructions [8]. Prior work has shown that integrating search with the user's context can make it easier to find and use help resources [4, 7, 9, 17, 37], but even still, people do not always search for help when they need it [9]. Coming up with a search query and filtering through results often feels like it will take longer than trial-and-error [9]. How might we lower the barrier to searching and make it easier for people to find the resources they need in the moment?


"search how do I use this"
a
b
Flat
50
ReMap
GO
c
d
Noise Reduction and Equalizer ...
SUBSCRIBE
0:56 / 5:08
YouTube
Reducing Background Noise In...

Searching for imovie how do i use noise reduction and equalizer

how do I use noise reduction and equalizer

Include in search: ☑ iMovie

Noise Reduction and Equalizer in iMovie 10.0.1 | Tutorial 9
Apps mentioned: iMovie

...both of these tabs are is both the audio management and voicing **noise reduction** voice **reduction** in equalizing of

Reducing Background Noise In iMovie. How To Mute
Apps mentioned: iMovie

Figure 1. ReMap is a multimodal search interface for finding learning videos. a) The user speaks their query. b) The user clicks on the noise reduction tool in iMovie while saying *"this."* c) ReMap automatically changes the word *"this"* to *"noise reduction and equalizer."* d) ReMap highlights relevant moments on the timeline of each video result.

This paper explores how multimodal interaction can make searching easier. When people help each other, they use language, gestures, and shared context to communicate. In contrast, finding help through search engines is still primarily text-based. What if users could communicate their needs as naturally as they would when asking a question in person? Leveraging the strengths of multiple modalities and integrating them smoothly can improve communication [28]; this work explores whether integrating multiple modalities into information search similarly helps people communicate their questions.

We introduce ReMap (Figure 1), a multimodal interface that allows users to search for learning videos using speech and pointing, without taking their hands (or mouse) off their current task. Users can initiate a search at any time by saying *"search,"* followed by their query (*e.g.*, *"how to make a text shadow"*). ReMap augments search queries with information about the user's context (*e.g.*, that they are using the graphic design software Canva) and highlights relevant moments on the

UIST '20, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7514-6/20/10 ...$15.00.
http://dx.doi.org/10.1145/3379337.3415592

timelines of video results based on the user's query and context. Users can play and navigate video results using speech commands while following along in their software, to avoid switching back and forth between windows. Users can also point at interface elements to include their names in the search query (*e.g.*, *"how do I use this"* while clicking on the noise reduction tool in iMovie), removing the need to learn or recall application-specific terminology. ReMap supports a wide variety of desktop and web applications, including graphic design software, movie editors, prototyping tools, and productivity software. A study with thirteen participants found that ReMap allows people to stay focused on their task while help-seeking. This paper contributes:

1. ReMap, a multimodal system that allows users to search for and navigate videos in context using speech and pointing,
2. an approach for enabling real-time deictic resolution of interface elements across software applications that leverages accessibility APIs,
3. findings from a qualitative lab study illustrating the benefits and challenges of multimodal help search, and
4. recommendations for improving the usability and robustness of a multimodal search interface.

ReMap introduces the first system-wide means for multimodal contextual help search. This paper provides direction and guidance for future work, as the increasing adoption of speech and touch brings new opportunities for natural interaction.

RELATED WORK

Presenting Help Resources in Context Eases Their Use

Embedding learning resources in software reduces the need for context switching and supports active learning [11, 17, 21, 22, 37]. Prior work has demonstrated the benefits of contextually presenting software videos [9, 11, 21], tutorials [17, 37], and discussion fora [22]. This work directly extends RePlay [9], a search interface for finding learning videos in context. RePlay uses relevant context from the user's activities across software applications to improve search results and highlight relevant moments in videos. RePlay uses OS accessibility APIs to obtain the names of interface elements the user clicks in any accessibility-labeled software. While a lab study found that RePlay helped people find results faster, the attentional cost of switching to RePlay discouraged participants from using it more frequently [9]. Novices often struggle to articulate queries with the right terminology [9, 32] and people are sometimes reluctant to take their hands off their current task to search for help when they need it [9, 27]. ReMap lowers the switching cost and cognitive load of help-seeking by introducing multimodal interaction for search and navigation.

The Power of Speech

Speech interaction is rapidly becoming ubiquitous; in 2019, 43% of global internet users reported using a voice assistant [1]. Speech input is especially appealing on mobile devices as people often use them on the go [12]; but even on desktop computers, many people use voice assistants for web search [24]. Speaking a search query is often easier and faster than typing, and it allows the user to ask a question like they would ask a friend; mobile voice queries tend to be closer to natural language than text queries [12]. People are especially likely to use speech when seeking audio or video results [12], supporting ReMap's approach of using speech to search for videos. Finally, speech may be more useful for people with specific goals: Laput *et al.* [18] found that when editing photos, speech was most useful when people knew exactly what they wanted to do. Similarly, ReMap is intended for situations where users have targeted questions in the middle of a task, as RePlay (the system ReMap extends) was most helpful in such cases [9].

Combining Modalities can Maximize Cognitive Abilities

Combining input from multiple modalities (*e.g.*, speech, gesture, touch) can reduce cognitive load for complex tasks [31], make tedious tasks more efficient [23, 28], reduce errors [28], increase precision [23], and even make tasks more enjoyable [18, 28]. Multimodal systems can maximize users' working memory by using different modalities for different types of information [15, 31]. For example, using speech to navigate tutorial videos while one's hands are busy with a physical task allows users to process the video and task simultaneously [5]. ReMap similarly partitions modalities between the user's task and ReMap's interface to maximize users' cognitive abilities (Figure 2). Users primarily use speech input and auditory output to search for and listen to videos while keeping their visual and motor attention on their main task.

Pointing at objects and spatial locations is often easier and more natural than describing them in words [2, 31]. When combined with speech, pointing allows people to communicate more precisely by referring to objects and locations with deictic terms (*e.g., "this"*, *"here"*) [2, 18]. ReMap allows users to deictically reference interface elements and canvas objects while speaking a search query. Speech and pointing can be especially beneficial for creative tasks, where maintaining a flow state is important. For example, using speech to access tools in graphic design [16] or drawing [33] software helps artists stay focused on their work. Not surprisingly, much prior work on multimodal systems has centered around such visual creative tasks [16, 18, 30, 33, 34, 35]. However, most of this prior work used speech to execute commands rather than issue search queries. This paper combines insights from multimodal creative systems and voice search systems to explore how multimodal search might be useful in creative software, though we also expect ReMap to be useful for other types of software tasks.

		Output Modality	Input Modality
Used for:	Creative task*	**Visual**	**Motor**
	Help-seeking	**Auditory**	**Verbal**

Figure 2. ReMap partitions the above modalities between the user's creative task and their help-seeking task to maximize cognitive abilities. The * indicates that the visual and motor modalities are not *exclusively* used for the creative task; users can also transfer their visual and motor attention to the help resources when needed to watch a video in detail.

REMAP SYSTEM DESIGN AND IMPLEMENTATION

ReMap's interface (Figure 1) can be positioned next to whatever other software is being used. Users can type a query in its search field and view results from YouTube with relevant moments overlaid on the timeline (Figure 1d). ReMap introduces three multimodal features to make searching easier: using speech to search, making deictic references in a search query, and navigating video results using speech commands.

Searching for Help Using Speech

At any time, the user can search for help by saying *"search"* followed by their query (Figure 1a). Much like other speech interfaces (*e.g.,* Siri), ReMap's search field displays the query as it is being transcribed. Once the user is finished speaking, ReMap executes a search with the transcribed query. This allows the user to keep their visual and motor attention on the current task while using the verbal input modality to search.

Making Deictic References in a Search Query

Especially with new software, people are often unfamiliar with an application's vocabulary but can point at application elements that are relevant to their goal. To alleviate the challenge of remembering application-specific terms and to keep the user's visual and motor attention on their current task, ReMap allows users to deictically reference interface elements and objects while speaking their query. If the user says *"this"* or *"that"* while clicking on a detectable element, ReMap replaces the pronoun with the element's name (Figure 1b-c). Detectable elements include buttons, checkboxes, sliders, images, graphics, text fields, and menu items. Similar to RePlay [9], ReMap uses the MacOS Accessibility API to obtain the names of clicked elements as long as they have accessibility labels. Many modern applications and websites label menus, buttons, and other interface elements. Some also label canvas elements (such as text boxes, images, and graphics) though many do not. This paper's study used Canva (`canva.com`) as the primary software; Canva labels most canvas elements and interface buttons.

While ReMap is detecting a speech query, it stores a list of every detectable element clicked. Once the user is finished speaking, ReMap replaces all occurrences of *"this"* and *"that"* with the element names in the order they were clicked before issuing the search query. This means that users need not speak the deictic reference at the exact same time as they click (which people rarely do [28]), they only need to click before they are finished speaking. While clicking an element could cause accidental input to the software itself, in practice we found most references tend to be either modal selections (*e.g.*, entering "text mode") or objects on the canvas. Future systems could consider disabling input to the software while the user is speaking so no accidental input is possible.

Navigating Video Results Using Speech Commands

ReMap allows users to navigate video results using speech commands. This was inspired by Chang *et al.*'s [5] findings that people often pause and skip forward or backward when following along with video tutorials so they can keep pace with the video, skip irrelevant content, or replay sections. Chang *et al.* [5] propose using speech to navigate tutorials for physical tasks so users do not have to take their hands off the task to navigate videos. Although ReMap is currently intended for digital tasks, not physical, we expect speech navigation to be similarly helpful, as it allows users to keep their motor attention on their task while using the verbal modality to control the video. Users can also keep their visual attention on their task while listening to the video's auditory output, switching their visual attention to the video only when necessary.

ReMap currently supports the following speech commands:

- *"play"* to play the first or most-recently played video
- *"play X video"* to play a specific video, where *X* is either the video's place in the results (*e.g.*, *"second"*) or *"next"* / *"previous"* relative to the current video
- *"next/previous/repeat marker"* to skip to a timeline marker
- *"pause"* and *"stop"* to pause the currently-playing video

While there are many other video-related actions that may also benefit from speech commands (*e.g.*, opening a larger window or adjusting video speed), ReMap initially focused only on the basic video controls (play, pause and navigation) to avoid overwhelming users with speech commands to learn.

Implementation

To enable efficient and flexible prototyping, we built ReMap as a combination of a desktop application and a custom web server with webpages for displaying videos and detecting speech. The desktop application is implemented as a MacOS Swift application (Figure 3a) so it can have access to the MacOS Accessibility API. The web server (Figure 3b) is implemented in Node.js. The speech webpage (Figure 3c) uses the Web Speech JavaScript API to detect and transcribe speech, and the video player webpage (Figure 3d) uses the YouTube Player JavaScript API to load and control videos. ReMap uses socket.io to communicate between the web server and the three client interfaces (the desktop application, speech webpage, and video player webpages).

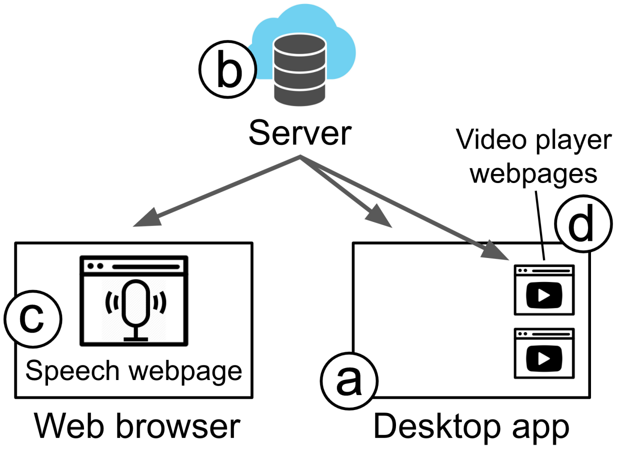


Figure 3. The ReMap system architecture. a) The user interacts with a MacOS desktop application that uses the Accessibility API to detect user context. b) A custom web server hosts c) a webpage for speech recognition and d) a video player webpage for displaying video results.

When launched, ReMap opens the speech webpage in the user's web browser. It can be minimized or hidden by the user. The speech webpage listens continuously so that the user can interact using speech at any time. The Web Speech API automatically determines when the user starts and finishes speaking, returning each phrase separately. If a phrase begins with the word *"search"*, the webpage sends the rest of the phrase to the server which sends it to the desktop application as a query. As the user continues to speak their query, the speech webpage sends the server the updated phrase every time a new word is detected, so the desktop application can display it in real time (Figure 1c). The desktop application displays video results in a list. Each video appears inside a separate Swift `WKWebView` object loaded to the video player webpage (with the video's YouTube ID as a query parameter).

If a phrase does not start with *"search"* and it matches a video navigation command (*e.g.*, *"next marker"*), the server sends this command to the desktop application, which keeps track of the most-recently played video and whether it is currently playing or not. Then, when the desktop application receives a navigation command, it can determine whether it needs to play a different video or pause/play the current one.

STUDY: USING REMAP FOR A GRAPHIC DESIGN TASK

To gain an initial understanding of how people use multimodal search for help, we conducted a think-aloud lab study with thirteen participants. Participants re-created a graphic design in Canva and used ReMap to search for help when they needed it. Overall we found that despite some usability and implementation challenges, multimodal video search was helpful, allowing participants to stay focused on their task while simultaneously searching for and navigating video resources.

Participants

Thirteen participants were recruited from mailing lists and flyers at a university. 10/13 participants had at least some experience using voice assistants (*e.g.*, Siri, Amazon Echo) (*mean* = 2.3/5). 6/13 participants had never used Canva before, and only one was very familiar with it. (*mean* = 1.8/5).

Procedure

Participants were asked to choose one of two infographic designs (Figure 4) and re-create it as accurately as possible in Canva. Both infographics were designed to require several operations that are not straightforward in Canva, to increase the likelihood that participants would have to search for help. Participants were asked to use only ReMap (no web search) when they needed to search for help. Participants were given a brief tutorial on how to use ReMap, and were asked to try three example speech commands to ensure they understood how it worked. They were encouraged to use speech as much as possible, but could also type their search queries into ReMap's search field and navigate videos using the mouse.

Participants were asked to think out loud while they worked, both to help the researchers understand their thought processes and to help offset the novelty challenge of interacting with a computer using speech. Since talking to a computer might feel unusual, especially while being observed in a study, the think-aloud protocol helped participants feel more comfortable by encouraging them to talk throughout the study. Participants were scheduled for 2-hour slots and were told to take as much time as they needed. Once they decided they were done (or if 1 hour and 45 minutes had passed), they were asked a series of interview and Likert-scale questions about their experience. Participants received a $30 USD gift card for their time.

Results: Multimodal Search Enables Multitasking

As Table 1 shows, participants found ReMap's multimodal features moderately helpful. Based on our observations, most participants used the multimodal features to work and search or watch videos simultaneously. They confirmed this during the post-task interviews, with several explicitly mentioning that ReMap's multimodal features allowed them to multitask, splitting their attention between the design task in Canva and the tasks of searching for and watching videos.

Searching with Speech: Often Useful, Sometimes Hard

Participants issued a total of 118 intentional search queries. 111 of these (94%) used speech, suggesting that participants were mostly able to communicate their queries with speech. An additional 3 speech queries were issued by mistake (not realizing they had spoken the *"search"* command) and an additional 7 were issued before the participant was done speaking (because they paused and the Web Speech API detected this as the end of the phrase). One participant did not search at all (the same participant that rated their familiarity with Canva as 5/5); the rest issued between 3 and 17 queries each.

Speech queries were 4.50 words long on average ($n = 111$, $SD = 2.36$), considerably longer than queries made with RePlay (2.53 words) [9]. This corroborates prior work showing that voice queries tend to be longer than typed queries on both desktop [24] and mobile [12]. Indeed, typed queries issued with ReMap were only 3.14 words long on average, but this was a small sample size ($n = 7$, $SD = 1.46$). 55/118 queries (47%) began with either *"how to"* or *"how do I"*.

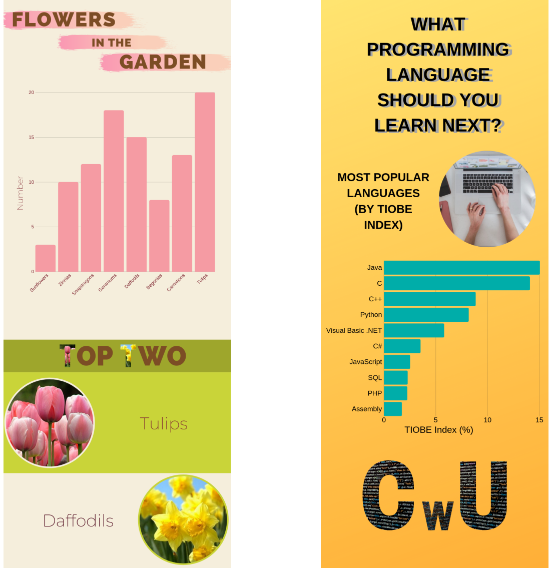


Figure 4. Study participants chose one of the above two infographic designs to re-create in Canva, using ReMap to search for help.

Feature	Timeline Markers	Play/Pause	Search	Deixis
Helpfulness	4.0	3.0	3.5	3.0

Table 1. Median ratings of ReMap's multimodal features. 1 = not helpful at all, 5 = very helpful.

63/118 queries (53%) were new and 39/118 (33%) were reformulations of previous queries (*i.e.,* rephrasing a query to find better results, a common action when help-seeking). The rest were attempts to fix either a failed deictic resolution (10/118, 8%) or a transcription error (6/118, 5%). Of those attempts, 5/10 deictic fixes and 6/6 error fixes were successful. 8/111 speech queries (7%) included a transcription error. 6/7 typed queries were manual revisions of a previous speech query (either error fixes or reformulations).

Several participants said that speaking their query allowed them to stay focused on the task in Canva while searching. Some also said it was easier or faster than typing. However, some other participants found it more difficult to speak their query, as they didn't always know what to say when they started, and they were not used to speaking out loud to search: *"I had a lot of trouble getting my queries totally straight or thought out in my head before starting to speak them"* (P13).

Deictic Resolution: Mostly Used for Canvas Elements

7/13 participants used deictic references at least once, and 22% (24/111) of spoken queries included a deictic reference. Participants said deixis helped them include words they didn't know in their queries and was often easier and faster than saying or typing the words explicitly. However, some felt they did not need to use deixis because they already knew the words they wanted to include: *"It might be more helpful for things that you don't know the exact terminology for, but I think I knew some of the terminology so just saying it felt faster"* (P1). Canva has a relatively simple vocabulary as its features are mainly limited to shapes, charts, images, and text.

23/24 deictic references referred to objects on the canvas (*e.g.*, text boxes, images, and charts); the other referred to a button on the toolbar. This reinforces prior work showing that people tend to make action-oriented queries rather than queries about tools [3, 9]. Although ReMap was intended primarily for novice users, the ability to deictically reference software tools may be more useful for experienced users who know what tool they need to accomplish a goal, while novices find it more useful to reference the objects they want to operate on.

Unfortunately, only 25% (6/24) of deictic references were successfully resolved to a name, mainly due to missing accessibility labels. For example, 10/18 unsuccessful deictic references were for charts, but charts in Canva do not have accessibility labels. We chose Canva for this study because it has more accessibility labels for canvas objects than most other creative software; many do not label anything on the canvas at all. However, even Canva does not label all elements.

Two participants also pointed out that their eyes naturally went to the search field where their query was appearing while they spoke it, which made it difficult to look at Canva to reference elements: *"even though I was trying to click here I was focusing on [the search field]"* (P9).

Video Navigation: Preference Depended on Participant

Most participants exhibited a preference for either speech commands or manual navigation of videos, using mainly one or the other. This highlights the importance of providing both options in a multimodal system, especially as peoples' preferences and needs may change depending on their task or environment [19, 31]. In total, participants played videos by clicking 60 times, and by saying one of the *"play"* commands 61 times. Participants paused manually 60 times, and spoke the *"pause"* command 46 times. Participants manually navigated to points in the video by dragging on the timeline 124 times, by clicking ReMap's timeline markers 28 times, and by saying the *"next/previous/repeat marker"* commands 60 times.

Participants who preferred navigating videos with speech said it helped them watch and control the videos while simultaneously working on their task in Canva. It also allowed them to follow along with videos at their own pace, which prior work has demonstrated the importance of [5]. The timeline markers were rated as ReMap's most helpful multimodal feature on average (Table 1), echoing RePlay's finding that timeline markers provide a useful shortcut to potentially relevant moments in the video [9]. The speech commands for skipping between markers made it easy for participants to back-up or fast-forward the video to a reasonable point. This is easier than having to specify a time interval to skip between, which Chang *et al.* [5] found can be difficult. Participants who preferred manual navigation felt it was just as fast or easy, and didn't require them to remember the speech commands.

DISCUSSION

Overall, most participants were positive about multimodal help search, and many of the challenges they exhibited stemmed from either implementation issues or unfamiliarity with using speech and pointing for search and navigation. It is likely that an improved implementation combined with more time spent using ReMap would further increase users' success. This section outlines how ReMap's usability and implementation could be improved based on the study findings.

Usability Challenges With Speech for Search

How to Indicate a Query is Finished?

Several participants were frustrated by ReMap issuing a search before they had finished speaking their query, usually because they had paused briefly to think about what to say next and the Web Speech API interpreted this as the end of a phrase. Most voice assistants also automatically detect when the user is finished speaking (*e.g.*, Siri, Alexa), and they do sometimes cut the user off early. Jiang *et al.* [14] found that these "system interruptions" accounted for about 10% of all transcription errors. In our study, they accounted for about 39% (7/18) of all transcription errors, including speech recognition errors and unintentional searches. For multimodal help-seeking, it may be preferable to require the user to explicitly press a button, say a keyword, or use a keyboard shortcut to indicate that they are finished speaking. While this adds an extra step, it would likely prevent errors and remove the pressure some participants felt to finish speaking. Future work should explore these trade-offs.

How to Correct Mistakes in a Speech Query?

Correcting speech commands or queries is a difficult problem [14, 26, 29]. Errors may be caused by the system (*e.g.*, transcription errors) or by the user (*e.g.*, saying a word they didn't intend). ReMap participants mostly made corrections by either typing edits or repeating the entire query over again, echoing previous findings [14, 26]. One participant corrected their query in real time by repeating it while speaking, much as one might correct oneself in normal conversation, which led to the entire utterance being transcribed as one long query. A smarter system might recognize this repetition and automatically extract the corrected version. Future work should explore how this and other methods for correction might be implemented. For example, Jiang *et al.* [14] recommend letting users specify and repeat a portion of a query to correct. Shokouhi *et al.* [36] showed that (at least on mobile) people prefer not to switch between speech and text when correcting a query, so one option could be to support speech commands like *"change X to Y"* to replace incorrect words using speech.

How Much to Show Users and When?

Two participants noted that seeing their speech transcribed in real-time made it hard to focus on Canva to make deictic references. Indeed, Kalyuga *et al.* [15] showed that splitting one's attention in the same modality increases cognitive load; in this case, users' visual attention was split between the software and ReMap's search field. We have since updated ReMap with a setting that specifies whether to show the query as it is transcribed or only when it is finished, as the best approach may depend on personal preference. Seeing one's speech transcribed in real time (as many mobile voice assistants do) assures the user that the system is actively listening and gives them immediate feedback. But in the case where the search interface is not the primary application being used, it may add more distraction than the assurance is worth.

How Might Multimodal Help Search be Used in Everyday Life?

To further understand the benefits and challenges of multimodal help search, future work should explore how people might use multimodal search for their own personal tasks over a longer period of time, as well as compare ReMap's efficacy with other prominent search tools. One potential challenge with real-world use of ReMap is the use of voice input in public spaces or open offices where people may not want to speak out loud, but recent work shows promising techniques for overcoming this challenge, such as enabling near-silent speech detection [10].

Improving the Robustness of System-Wide Assistance

ReMap detects actions and resolves deictic references across software applications by leveraging accessibility APIs. However, this approach only works when application developers provide the necessary accessibility labels for their software, which they sometimes do not. This section discusses how future work might address these gaps.

Leveraging Accessibility on Mobile Systems

ReMap can only detect interface elements that have been labeled with accessibility information by the application developer, which as prior work has shown, varies widely across applications [13]. This paper's study also found that even in one of the most thoroughly labeled creative applications (Canva), some accessibility labels were still missing (*e.g.,* charts). Accessibility labeling tends to be more common for interface "widgets" such as menu items and tools, and less common for the "insides" of applications such as editing areas or canvases [13]. However, newer operating systems for mobile platforms such as Android and iOS tend to have more accessibility features built in and more standardized interface components, so ReMap may work more reliably on mobile platforms. Prior work has shown how mobile systems can leverage accessibility information to enable programming by demonstration [20]; ReMap could similarly use accessibility to provide contextual support across tablet and smartphone applications. Speech and pointing are also more common with mobile devices, which may make ReMap's multimodal interaction feel more natural.

Alternative Approaches for Detecting Interface Elements

For web applications, one alternative to accessibility APIs could be to leverage built-in properties of the DOM such as element types and classes. As for desktop software, prior work has used computer vision to recognize interface elements [6, 13]. Computer vision may in some cases be *more* effective than accessibility APIs as it could allow for higher-level interpretation of the elements being clicked. As this paper's study found, most deictic references participants made were for canvas elements rather than interface tools. However, even when such elements have accessibility labels, they tend to be very general (*e.g.*, *"image"*). Higher-level semantics about the element may be more useful when searching for help (*e.g.*, knowing that an image contains a person). Of course, computer vision may also have limitations compared to accessibility APIs; the description or action associated with an element is not always apparent from its visual attributes alone. For example, many tools in creative applications are represented by icons only. Since no approach will be perfect, future work should also explore how systems could "fail gracefully" when an element's name is unknown. For example, the system could prompt the user to enter the element's name manually and use this to improve the labeling over time.

CONCLUSION

This paper introduced an approach for quick, in-context help-seeking that leverages the strengths of multiple modalities. We presented ReMap, a system that allows users to search for and navigate videos using speech, and include application-specific terminology in queries using deixis. An initial study showed that ReMap helps people stay focused on their task while navigating help resources, and highlighted several important directions for future work. As the tasks people can do with software become increasingly complex, the ability to ask questions about software easily and naturally is becoming especially important. Leveraging peoples' natural communication strategies as well as relevant context is key to improving virtual assistance.

ACKNOWLEDGEMENTS

We thank Srishti Palani for her help with formative exploration, and the UCSD ERSP program for supporting this research.

REFERENCES

[1] 2019. Top Global Consumer Trends in 2020. (2019). https://www.globalwebindex.com/reports/trends-2020

[2] Richard A. Bolt. 1980. “Put-that-there”: Voice and gesture at the graphics interface. *ACM SIGGRAPH Computer Graphics* 14, 3 (jul 1980), 262–270. DOI: http://dx.doi.org/10.1145/965105.807503

[3] Horatiu Bota, Adam Fourney, Susan T. Dumais, Tomasz L. Religa, and Robert Rounthwaite. 2018. Characterizing Search Behavior in Productivity Software. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval - CHIIR ’18*. ACM Press, New York, NY, USA, 160–169. DOI: http://dx.doi.org/10.1145/3176349.3176395

[4] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-centric programming: Integrating Web Search into the Development Environment. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI ’10*. ACM Press, New York, NY, USA, 513. DOI: http://dx.doi.org/10.1145/1753326.1753402

[5] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. 2019. How to Design Voice Based Navigation for How-To Videos. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19*. ACM Press, New York, NY, USA, 1–11. DOI: http://dx.doi.org/10.1145/3290605.3300931

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

[7] Michael Ekstrand, Wei Li, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Searching for software learning resources using application context. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST ’11*. ACM Press, New York, NY, USA, 195. DOI: http://dx.doi.org/10.1145/2047196.2047220

[8] Adam Fourney, Ben Lafreniere, Parmit Chilana, and Michael Terry. 2014. InterTwine: creating interapplication information scent to support coordinated use of software. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST ’14*. ACM Press, New York, NY, USA, 429–438. DOI:http://dx.doi.org/10.1145/2642918.2647420

[9] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19*. ACM Press, New York, NY, USA, 1–13. DOI: http://dx.doi.org/10.1145/3290605.3300527

[10] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable voice input by ingressive speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology - UIST ’18*. ACM Press, New York, NY, USA, 237–246. DOI: http://dx.doi.org/10.1145/3242587.3242603

[11] Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI ’10*. ACM Press, New York, NY, USA, 1515. DOI: http://dx.doi.org/10.1145/1753326.1753552

[12] Ido Guy. 2018. The Characteristics of Voice Search: Comparing Spoken with Typed-in Mobile Web Search Queries. *ACM Transactions on Information Systems* 36, 3 (apr 2018), 1–28. DOI: http://dx.doi.org/10.1145/3182163

[13] Amy Hurst, Scott E. Hudson, and Jennifer Mankoff. 2010. Automatically identifying targets users interact with during real world tasks. In *Proceedings of the 15th international conference on Intelligent user interfaces - IUI ’10*. ACM Press, New York, NY, USA, 11. DOI: http://dx.doi.org/10.1145/1719970.1719973

[14] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How do users respond to voice input errors?: lexical and phonetic query reformulation in voice search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’13*. ACM Press, New York, NY, USA, 143. DOI:http://dx.doi.org/10.1145/2484028.2484092

[15] Slava Kalyuga, Paul Chandler, and John Sweller. 1999. Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology* 13, 4 (aug 1999), 351–371. DOI: http://dx.doi.org/10.1002/(SICI)1099-0720(199908)13:4<351::AID-ACP589>3.0.CO;2-6

[16] Yea-Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. 2019. Vocal Shortcuts for Creative Experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19*. ACM Press, New York, NY, USA, 1–14. DOI: http://dx.doi.org/10.1145/3290605.3300562

[17] Benjamin Lafreniere, Andrea Bunt, and Michael Terry. 2014. Task-centric interfaces for feature-rich software. In *Proceedings of the 26th Australian Computer-Human Interaction Conference - OzCHI ’14*. ACM Press, New York, NY, USA, 49–58. DOI: http://dx.doi.org/10.1145/2686612.2686620

[18] Gierad P. Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. PixelTone: a multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13*. ACM Press, New York, NY, USA, 2185. DOI: http://dx.doi.org/10.1145/2470654.2481301

[19] Joseph J. LaViola Jr., Sarah Buchanan, and Corey Pittman. 2014. Multimodal Input for Perceptual User Interfaces. In *Interactive Displays*. John Wiley & Sons, Ltd, Chichester, UK, 285–312. DOI: http://dx.doi.org/10.1002/9781118706237.ch9

[20] Toby Jia-Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating Multimodal Smartphone Automation by Demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, New York, NY, USA, 6038–6049. DOI: http://dx.doi.org/10.1145/3025453.3025483

[21] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011a. Ambient help. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, New York, NY, USA, 2751. DOI: http://dx.doi.org/10.1145/1978942.1979349

[22] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011b. IP-QAT: in-product questions, answers, & tips. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, New York, NY, USA, 175. DOI: http://dx.doi.org/10.1145/2047196.2047218

[23] Sven Mayer, Gierad Laput, and Chris Harrison. 2020. Enhancing Mobile Voice Assistants with WorldGaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems - CHI '20*. ACM Press, New York, NY, USA, 1–10. DOI: http://dx.doi.org/10.1145/3313831.3376479

[24] Rishabh Mehrotra, Ahmed Hassan Awadallah, Ahmed El Kholy, and Imed Zitouni. 2016. Hey Cortana! Exploring the use cases of a Desktop based Digital Assistant. In *Proceedings of ACM, Tokyo, Japan, August 2017 (CAIR'17)*. 5.

[25] Naomi Miyake and Donald A. Norman. 1979. To ask a question, one must know enough to know what is not known. *Journal of Verbal Learning and Verbal Behavior* 18, 3 (jun 1979), 357–364. DOI: http://dx.doi.org/10.1016/S0022-5371(79)90200-7

[26] Chelsea Myers, Anushay Furqan, Jessica Nebolsky, Karina Caro, and Jichen Zhu. 2018. Patterns for How Users Overcome Obstacles in Voice User Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, New York, NY, USA, 1–7. DOI: http://dx.doi.org/10.1145/3173574.3173580

[27] David G. Novick, Oscar D. Andrade, and Nathaniel Bean. 2009. The micro-structure of use of help. In *Proceedings of the 27th ACM international conference on Design of communication - SIGDOC '09*. ACM Press, New York, NY, USA, 97. DOI: http://dx.doi.org/10.1145/1621995.1622014

[28] Sharon Oviatt. 1999. Ten myths of multimodal interaction. *Commun. ACM* 42, 11 (nov 1999), 74–81. DOI:http://dx.doi.org/10.1145/319382.319398

[29] Tim Paek, Bo Thiesson, Yun-Cheng Ju, and Bongshin Lee. 2008. Search Vox: leveraging multimodal refinement and partial knowledge for mobile voice search. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*. ACM Press, New York, NY, USA, 141. DOI: http://dx.doi.org/10.1145/1449715.1449738

[30] Randy Pausch and James H. Leatherby. 1991. An Empirical Study: Adding Voice Input to a Graphical Editor. *Journal of the American Voice Input/Output Society* 9 (1991), 2–55.

[31] Leah M. Reeves, Jean-Claude Martin, Michael McTear, TV Raman, Kay M. Stanney, Hui Su, Qian Ying Wang, Jennifer Lai, James A. Larson, Sharon Oviatt, T. S. Balaji, Stéphanie Buisine, Penny Collings, Phil Cohen, and Ben Kraal. 2004. Guidelines for multimodal user interface design. *Commun. ACM* 47, 1 (jan 2004), 57. DOI:http://dx.doi.org/10.1145/962081.962106

[32] Daniel M. Russell. 2011. Making the Most of Online Searches. *APS Observer* 24, 4 (apr 2011). https://www.psychologicalscience.org/observer/making-the-most-of-online-searches

[33] Jana Sedivy and Hilary Johnson. 1999. Supporting creative work tasks: The potential of multimodal tools to support sketching. In *Proceedings of the third conference on Creativity & Cognition - C&C '99*. ACM Press, New York, NY, USA, 42–49. DOI: http://dx.doi.org/10.1145/317561.317571

[34] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. 2016. Eviza: A Natural Language Interface for Visual Analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*. ACM Press, New York, NY, USA, 365–377. DOI: http://dx.doi.org/10.1145/2984511.2984588

[35] Anirudh Sharma, Sriganesh Madhvanath, Ankit Shekhawat, and Mark Billinghurst. 2011. MozArt: a multimodal interface for conceptual 3D modeling. In *Proceedings of the 13th international conference on multimodal interfaces - ICMI '11*. ACM Press, New York, NY, USA, 307–310. DOI: http://dx.doi.org/10.1145/2070481.2070538

[36] Milad Shokouhi, Rosie Jones, Umut Ozertem, Karthik Raghunathan, and Fernando Diaz. 2014. Mobile query reformulations. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14*. ACM Press, New York, NY, USA, 1011–1014. DOI: http://dx.doi.org/10.1145/2600428.2609497

[37] Laton Vermette, Parmit Chilana, Michael Terry, Adam Fourney, Ben Lafreniere, and Travis Kerr. 2015. CheatSheet: A Contextual Interactive Memory Aid for Web Applications. In *Proceedings of the 41st Graphics Interface Conference (GI '15)*. Canadian Information Processing Society, Canada, 241–248.