ReMap: Multimodal Help-Seeking

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

ABSTRACT
ReMap is a multimodal interface that enables searching for learning videos using speech and in-task pointing. ReMap extends multimodal interaction to help-seeking for complex tasks. Users can speak search queries, adding app-specific terms deictically. Users can navigate ReMap's search results via speech or mouse. These features allow people to stay focused on their task while simultaneously searching for and using help resources. Future work should explore how to implement more robust deictic resolution and more modalities.

Author Keywords
multimodal interaction; speech; deixis; contextual search

INTRODUCTION
We introduce ReMap, a multimodal interface for users to search for learning videos using speech and pointing, without taking their hands (or mouse) off their current task. ReMap builds on the existing video search interface RePlay [3].

People often seek help via online resources like discussion fora or video tutorials. However, this requires switching mental context, visual attention, and input focus away from the task at hand. Furthermore, users must compose a search query using the same terms the resources do. Lastly, users must switch their attention back and forth between the resource they find and their task to follow along with instructions.

This demo explores how multimodal interaction might alleviate these help-seeking challenges. Different types of information lend themselves better to different modalities; leveraging the strengths of multiple modalities and integrating them smoothly can be extremely effective [7]. For example, combining speech and pointing allows people to communicate more precisely and efficiently by using deictic terms (*e.g.,* "this", "here") to refer to objects and locations [1, 6]. Furthermore, using multiple modalities simultaneously can improve efficiency; *e.g.,* navigating tutorial videos with speech while one's hands are busy with a physical task [2]. Finally, multimodal systems can also enable more natural interaction; *e.g.,* letting users describe photo edits in their own words and inferring the appropriate commands [6], or activating software commands with speech rather than memorizing keyboard shortcuts [5].

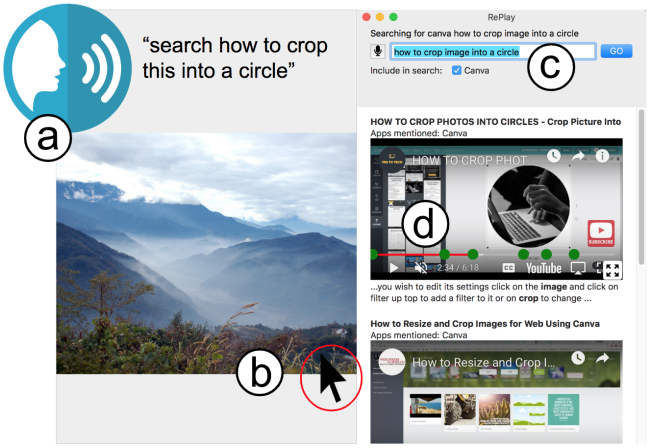

Figure 1. ReMap is a multimodal search interface for finding learning videos. a) The user speaks their query. b) The user clicks on an image on the canvas while saying the word "this." c) ReMap automatically changes the word "this" to "image." d) ReMap highlights relevant moments by placing markers on the timeline of each video result.

ReMap's multimodal help search demonstrates three main design insights:

1. Users can initiate and dictate a search at anytime using speech, to avoid context-switching.

2. Users can point at elements in the software they are using to include their names in the search query, removing the need to remember app-specific terminology.

3. Users can play, pause, and navigate video results using speech, allowing them to simultaneously work on their task and follow along with a video tutorial.

A study with 13 participants found that ReMap allows people to stay focused on their task while help-seeking. Future work should explore how to enable more robust deictic resolution.

REMAP SYSTEM DESIGN AND IMPLEMENTATION
ReMap (Figure 1) extends the RePlay contextual search system [3]. RePlay enables users to search for learning videos in context while working in software, and it highlights relevant moments in video results based on the user's context and query. While RePlay helps people find results faster, the attentional cost of switching to RePlay discouraged its being used as often as it could. ReMap lowers the switching cost and load by introducing three main improvements over RePlay.

UIST '19 Adjunct, October 20-23, 2019, New Orleans, LA, USA.

ACM ISBN 978-1-4503-6817-9/19/10.
http://dx.doi.org/10.1145/3332167.3356884

Searching for help using speech

ReMap uses the Web Speech API to detect speech by opening a browser page in the background when launched. The page's JavaScript invokes continuous listening and sends the current phrase to ReMap's custom web server whenever a new word is detected. Web Speech automatically determines when the user starts and finishes speaking, returning each phrase separately. If a phrase begins with *"search"*, ReMap initiates a search (Figure 1a), using the rest of the phrase as the query. Otherwise, it checks if the phrase matches any video navigation commands. If it does not, ReMap ignores it.

Making deictic references in a search query

Especially with new software, people are often unfamiliar with an application's vocabulary but can point at goal-relevant application elements. To alleviate the challenge of remembering app-specific terms, ReMap allows users to deictically reference interface elements and objects. If the user says *"this"* or *"that"* while clicking on a detectable element, ReMap replaces the pronoun with the reference element's name (Figure 1b-c).

ReMap uses the MacOS Accessibility API to resolve element names. This API can get the name and description of any element with accessibility labels [3]. Many modern applications have labeled menus, buttons, and other interface elements. Some also label canvas elements (such as text boxes, images, and graphics) though many do not. The examples in this demo use 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 and ReMap has obtained the final query from the server, it replaces all occurrences of *"this"* and *"that"* with the element names in the order they were clicked.

Navigating video results using speech commands

The user can speak commands to navigate video results, inspired by Chang *et al.*'s recommendations [2]. ReMap currently supports the following commands: *"play"* (plays the first or most-recently played video), *"play {next, previous, last}"* (plays the next/previous/last video in the list), *"play {first, second, third, fourth, fifth} video"*, *"{next, previous, repeat} marker"* (skips to the next or previous timeline marker, or re-starts from the current marker), and *"pause"* (pauses the currently playing video).

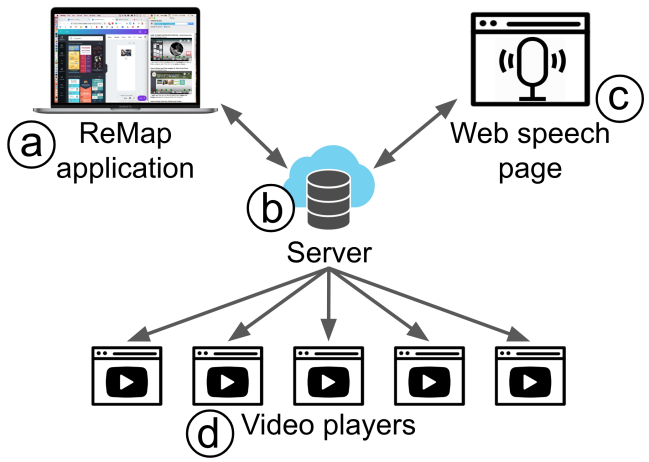

Figure 2. The ReMap system architecture. a) ReMap is a MacOS application that uses the Accessibility API to detect user context. b) ReMap connects to a web server, which opens c) a webpage for speech recognition, and d) a video player webpage for each result to embed in ReMap.

Implementation

ReMap is implemented as a MacOS Swift application (Figure 2a). It uses socket.io to communicate between the web server and the three client interfaces (the ReMap app, web speech engine, and video players). The web server (Figure 2b) is implemented in Node.js. The speech engine (Figure 2c) uses the Web Speech API, and the video player (Figure 2d) uses the YouTube Player API to load and control videos. When ReMap receives a video navigation command, it passes it to the server along with which video it applies to; the server then sends the command to the appropriate video player.

REMAP USAGE IN THE LAB AND IN THE WILD

To gain an initial understanding of how people use multimodal search for help, we conducted a think-aloud lab study with thirteen participants at a university. Participants were given a design to re-create in Canva and used ReMap to search for help as they worked.

Participants issued a total of 125 search queries; 118 used speech. Most participants used multimodal features to work and search or watch videos simultaneously. 7 of 13 participants used deictic references at least once. Only 6 of 24 deictic references were successfully resolved to a name, mainly because some canvas objects were not recognizable by ReMap (*e.g.,* graphs). More thorough accessibility labeling or integrated application plugins could improve this functionality.

Since ReMap's speech detection is always on, participants may encounter a "Midas touch" problem [4] of searching unintentionally. This happened occasionally but not frequently; future iterations of ReMap will more thoroughly explore the impact of such design decisions.

We have also demoed ReMap at two open events, both in large spaces with over 100 attendees. ReMap's speech recognition requires a clear signal of the user's speech, mostly free of interference from sound output or background conversations. We have found commodity headsets to be sufficient, high-quality headsets to be optimal, and built-in laptop microphones insufficient. Overall, attendees who tried ReMap were excited about its multimodal features, particularly the potential of deictic resolution.

CONCLUSION

ReMap demonstrates multimodal interaction for quick, in-context help-seeking by leveraging the strengths of multiple modalities. Users can search for videos using speech, use deixis to include app-specific terminology, and use speech to navigate videos. Initial usage showed that ReMap helps people stay focused on their task while navigating help resources, but further research is needed to provide robust deictic resolution and fully explore its potential for help-seeking.

REFERENCES

[1] 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

[2] 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, New York, USA, 1–11. DOI: http://dx.doi.org/10.1145/3290605.3300931

[3] 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, New York, USA, 1–13. DOI: http://dx.doi.org/10.1145/3290605.3300527

[4] Robert J. K. Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI ’90*. ACM Press, New York, New York, USA, 11–18. DOI:http://dx.doi.org/10.1145/97243.97246

[5] 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, New York, USA, 1–14. DOI: http://dx.doi.org/10.1145/3290605.3300562

[6] Jason Linder, Gierad Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, and Eytan Adar. 2013. PixelTone: A multimodal interface for image editing. In *CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA ’13*. ACM Press, New York, New York, USA, 2829. DOI: http://dx.doi.org/10.1145/2468356.2479533

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