WorkCache: Salvaging siloed knowledge

Scott Carter, Laurent Denoue, Matthew Cooper
FX Palo Alto Laboratory, Inc.
3174 Porter Drive
Palo Alto, California 94304 USA
carter,denoue,cooper@fxpal.com

ABSTRACT
The proliferation of workplace multimedia collaboration applications has meant on one hand more opportunities for group work but on the other more data locked away in proprietary interfaces. We are developing new tools to capture and access multimedia content from any source. In this demo, we focus primarily on new methods that allow users to rapidly reconstitute, enhance, and share document-based information.

CCS Concepts
Information systems → Information systems applications → Multimedia information systems → Multimedia content creation

Keywords
Multimedia capture and access; image processing; video processing; interactive documents

1. INTRODUCTION
New multimedia messaging tools (such as Slack) and an explosion of real-time video services have changed the landscape of everyday knowledge work, which is now increasingly collaborative and distributed. At the same time, this proliferation of group support applications has increased the amount of multimedia data tied to specific protocols and representations, counter-intuitively siloing data within work cliques that use a particular subset of tools. Overall, these trends can make it more difficult to share knowledge between disparate work groups, potentially limiting productivity as well as mitigating the type of spontaneous connections that can drive innovation [1].

Similarly, previous work we conducted investigating the uptake of real-time collaboration tools in the enterprise showed that workers use an array of different synchronous and asynchronous multimedia communication tools [2]. Typically, users have a set of applications that they prefer for personal use, but have to use a different set of applications for different work contexts because of various institutional or third-party demands. Furthermore, study participants reported that they often struggled to extract meaningful information from these siloed, and often unstructured data sources.

As a first step toward remedying this issue, we present WorkCache, a tool to capture, analyze, and index any screen-based content. With WorkCache, users can capture either multimedia or document-based content using a browser extension (Figure 1). Multimedia content is uploaded or streamed directly to a backend server, which then runs image- and video-based analysis to automatically index multimedia content (see [3] for details). Document-based content is handled using a new tool we developed to capture documents via screen recordings. Using image stitching techniques, the system analyses the screen recording and produces a copy of the original document that is immediately available online as a Web page, viewable on any device with a basic web browser. This web-based viewer shows each page as an image, including any interaction and voice comments that the author added while recording.

Figure 1. WorkCache includes an extension that allows users to capture multimedia screen content or documents. Screen content is uploaded to a back end server, indexed automatically, and added to a search-based user interface. Documents are sent to a separate service (see Figure 3).

2. CREATING WORKCACHE DOCUMENTS
To record a document, users launch the document capture tool from the WorkCache extension and pick one window on their desktop that displays the document to capture, e.g. a window showing a Word file, and click a "Start" recording button.

They then interact freely with the document inside the application window, scrolling up and down through pages they wish to capture and share. During the recording, they have full access to the functions provided by the application, such as selecting text passages and moving their pointer over areas of the document. The system is able to recognize the two major kinds of document viewing metaphors: scrolling and pagination. All word processors or PDF viewers typically implement scrolling, while pagination is used for showing slide decks, e.g. PowerPoint. Using the same system, users can also pick a browser window showing a video of a lecture, in which the speaker is showing slides. At the end of the session, the user will have a reconstituted copy of the slides that were shown during the video lecture.

During users' interactions, the system analyses in real-time every captured frame of the window and turns them into an enhanced copy of the original document[1].

[1] See video at https://youtu.be/MTMpHwXj4dY

MM '16, October 15-19, 2016, Amsterdam, Netherlands
ACM 978-1-4503-3603-1/16/10.
http://dx.doi.org/10.1145/2964284.2973809

In order to generate this copy, the system applies four important steps: 1) Region of Interest (ROI) detection, 2) Image stitching, 3) Interaction lifting, and 4) Document generation.

ROI detection: Once the user is done recording, the system processes the first frame, binarizes it, and looks for long vertical and horizontal segments. It combines them to determine the largest rectangle as the ROI for that window.

If the system has correctly detected this ROI, the user simply clicks over the identified region and the system proceeds to the next step. Alternately, users simply drag a rectangle to manually specify what ROI to use. Some document types such as web pages do not have clear paginated layout, making it hard or impossible for an automatic ROI detector to find the correct rectangle.

Image stitching: Image stitching is a critical analytical step for the system. Brute force comparisons between frames to determine their vertical shift would be very CPU intensive. Instead, we borrow techniques from the image stitching literature [4] that use key point detection and matching as the basis for finding generic transformations between pairs of images and adapt it to our domain (Figure 2). These methods are fast and robust to some noise, making them appealing for a real-time implementation.

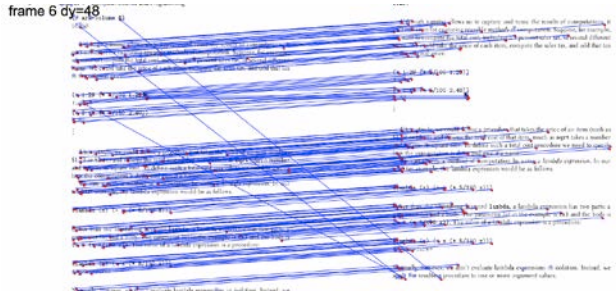

Figure 2. Vectors are matched in a vertical shift up: left previous frame, right next frame; blue lines join matching key-points (3 outliers); red dots show detected key-points.

Interaction lifting: Because the system records the screen as a video, users can also talk at the same time, e.g. to describe a figure that needs to be modified or express a feeling about a particular passage in the document. Authors can also interact with the document. These two features allow authors to add annotations to the document, a key component of active reading [5]. For example, an author can highlight specific words for revision with a co-author, or a teacher can explain a chart or figure by moving her cursor, or a lawyer can circle a whole section with her mouse cursor and ask a question verbally.

To detect these actions, it is enough to recognize when people only move their cursor and stop scrolling or paginating through the document.

Document generation: Given the set of frames and estimated shifts between them, the system generates a single composite image. Each frame is then copied at its corresponding Y position based on the accumulated delta offsets until that index. Special attention needs to be paid for cases when the user started recording her window at a later page and scrolls up.

Once the composite image is created, the system identifies likely page breaks: long horizontal lines that cross the whole ROI's width. Special attention is paid to small interrupted fragments; while recording, the mouse cursor has sometimes been found to overlap page boundaries, thus creating little discontinuities in the horizontal segments. To speed up implementation, our fast

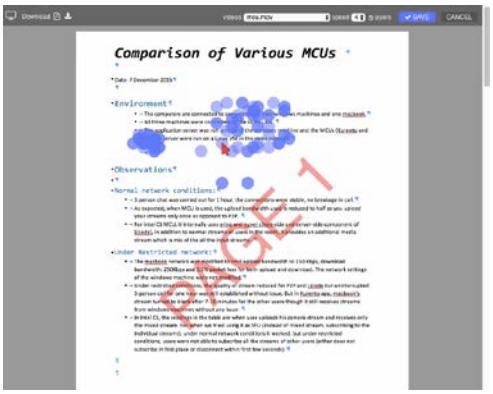

Figure 3. A sample reconstituted document; mouse actions and pages are preserved.

binarization step only computes the vertical gradient of the composite image so that it only detects horizontal edges.

Once page breaks have been found, the system cuts the tall image into as many smaller page images as necessary, padding the last one with white space in case the user had not completely captured it, thus giving the inferred document pages a more uniform look.

Finally, detected actions are overlaid on the corresponding page images as a DIV element that depicts a mouse pointer (Figure 3). When clicked, the original motion path is used to play an animation of the fake mouse cursor. The corresponding voice segment is played at the same time by seeking the audio track to the start time and playing it until the end of the segment.

All image-processing algorithms run in real-time inside the user's web browser in JavaScript. The video frames are captured using the MediaDevices API available to modern web apps, allowing us to capture either the full desktop or individual windows.

3. CONCLUSION

We built WorkCache to ameliorate the increasingly siloed nature of multimedia content. In this demonstration, we will present our capture and analysis methods, focusing in particular on features to reconstitute and annotate digital documents.

4. REFERENCES

[1] I. Nonaka and H. Takeuci. The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford Press, Oxford. 1945.

[2] J. Marlow, S. Carter, N. Good, and J-W. Chen. Beyond Talking Heads: Multimedia Artifact Creation, Use, and Sharing in Distributed Meetings. In Proceedings of ACM CSCW, 1703-1715. 2016.

[3] S. Carter, L. Denoue, and M. Cooper. Searching and Browsing Live, Web-based Meetings. In Proceedings of ACM MM, 791-792. 2015.

[4] L. Zhu, Y. Wang, B. Zhao and X. Zhang. A Fast Image Stitching Algorithm Based on Improved SURF. In proceedings of CIS, 171-175. 2014.

[5] A. Adler, A. Gujar, B. L. Harrison, K. O'Hara, and A. Sellen. A diary study of work-related reading: design implications for digital reading devices. In Proceedings of ACM CHI, 241–248. 1998.