Juho Kim
juhokim@mit.edu

November 24, 2014

Faculty Search Committee
Department of Communication Studies
Northwestern University

Dear Faculty Search Committee Members:

I am a Ph.D. candidate in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, focusing on Human-Computer Interaction research. I work in the User Interface Design Group in the Computer Science and Artificial Intelligence Laboratory, advised by Professor Robert C. Miller at MIT CSAIL and Professor Krzysztof Z. Gajos at the Harvard School of Engineering and Applied Sciences. I expect to complete my dissertation by June 2015, and I am interested in applying for the advertised tenure-track faculty position in Computation and Communication (Search No. 24144).

I build interactive technologies powered by large-scale user data. I am interested in designing and studying sociotechnical systems that support massive-scale collaboration, information sharing, and learning. My thesis research on learnersourcing proposes methods and design paradigms in which the byproduct of learning generates structured information difficult to collect at scale. My research uniquely combines crowdsourcing, social computing, visual and textual content analysis, and learning science, and introduces computational and social mechanisms to collect, process, and present structured information at scale.

I am passionate about teaching and mentoring. I plan to apply active learning methods to my courses, focusing on supporting high interactivity, learning by doing, and immediate feedback. I would be happy to teach courses in HCI and intro-level computer science and programming, as well as graduate-level courses covering crowdsourcing, social computing, and learning at scale.

I have asked that five letters of recommendation be sent to you on my behalf. The writers are:

Robert C. Miller (co-advisor):	rcm@mit.edu
Krzysztof Z. Gajos (co-advisor):	kgajos@eecs.harvard.edu
Meredith Ringel Morris:	merrie@microsoft.com
Haoqi Zhang:	hq@northwestern.edu
Philip J. Guo:	pg@cs.rochester.edu

My application materials, including my CV, research statement, teaching statement, and representative papers are enclosed. You can also find copies of my application materials at http://juhokim.com/. The best way to contact me is via email at juhokim@mit.edu, or by phone at (650) 796-9759.

Thanks in advance for your consideration, and please contact me if you need additional information.

Sincerely,
Juho Kim

JUHO KIM

MIT CSAIL | 32 Vassar Street, 32-G707, Cambridge, MA 02139
650.796.9759 | juhokim@mit.edu | www.juhokim.com

EDUCATION

Massachusetts Institute of Technology Cambridge, MA
Ph.D. candidate in Electrical Engineering and Computer Science (CSAIL) Sep.2010–Present
Advisors: Robert C. Miller & Krzysztof Z. Gajos. Expected: June 2015

Stanford University Stanford, CA
M.S. in Computer Science (Specialization: Human-Computer Interaction) Sep.2008–Jun.2010
Advisor: Scott R. Klemmer

Seoul National University Seoul, Korea
B.S. in Computer Science and Engineering Jun.2008
Graduated with honors (Summa Cum Laude)

RESEARCH INTERESTS

Human-Computer Interaction, Learning at scale, Data-driven interaction, Crowdsourcing, Learnersourcing

EMPLOYMENT

User Interface Design Group, MIT CSAIL Cambridge, MA
Research Assistant. Mentor: Robert C. Miller Sep.2010–Present

Microsoft Research Redmond, WA
Research Intern. Mentors: Meredith R. Morris & Andrés Monroy-Hernández May 2014–Aug. 2014

Learning Sciences Team, edX Cambridge, MA
Research Intern. Mentor: Piotr Mitros May 2013–Aug.2013

Creative Technologies Lab, Adobe Systems Inc. San Francisco, CA
Research Intern. Mentor: Joel Brandt May 2011–Sep.2011

USER Group, IBM Almaden Research Center San Jose, CA
Research Intern. Mentors: Eser Kandogan & Thomas P. Moran May 2010–Aug.2010

HCI Group, Stanford University Stanford, CA
Research Assistant. Mentor: Scott R. Klemmer Apr.2009–Jun.2010

SystemBase Co., Ltd. Seoul, Korea
Project Manager & Embedded Software Engineer Dec.2004–Sep.2007

Samsung Electronics Co., Ltd. Software Center Suwon, Korea
Summer Intern Jul.2003–Aug.2003

PUBLICATIONS

Conference Papers

[c.13] Learnersourcing Subgoal Labels for How-to Videos.
Sarah Weir, **Juho Kim**, Krzysztof Z. Gajos, & Robert C. Miller.
CSCW 2015: ACM Conference on Computer-Supported Cooperative Work and Social Computing.
To appear. (28.3% acceptance rate, 11 pages, with revise-and-resubmit cycle)

[c.12] Data-Driven Interaction Techniques for Improving Navigation of Educational Videos.
Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, & Robert C. Miller.
UIST 2014: ACM Symposium on User Interface Software and Technology. (22.2% acceptance rate, 10 pages)

[c.11] Content-Aware Kinetic Scrolling for Supporting Web Page Navigation.
Juho Kim, Amy X. Zhang, Jihee Kim, Robert C. Miller, & Krzysztof, Z. Gajos.
UIST 2014: ACM Symposium on User Interface Software and Technology. (22.2% acceptance rate, 5 pages)

[c.10] Attendee-sourcing: exploring the design space of community-informed conference scheduling.
Anant Bhardwaj, **Juho Kim**, Steven P. Dow, David Karger, Sam Madden, Robert C. Miller, & Haoqi Zhang.
HCOMP 2014: AAAI Conference on Human Computation & Crowdsourcing. (32% acceptance rate, 9 pages)

[c.9] Crowdsourcing step-by-step information extraction to enhance existing how-to videos.
Juho Kim, Phu Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, & Krzysztof Z. Gajos.
CHI 2014: ACM Conference on Human Factors in Computing Systems. (22.8% acceptance rate, 10 pages)
Honorable Mention Award (top 5%).

[c.8] Frenzy: collaborative data organization for creating conference sessions.
Lydia Chilton, **Juho Kim**, Paul André, Felicia Cordeiro, James Landay, Dan Weld, Steven P. Dow, Robert C. Miller, & Haoqi Zhang.
CHI 2014: ACM Conference on Human Factors in Computing Systems. (22.8% acceptance rate, 10 pages)
Honorable Mention Award (top 5%).

[c.7] Understanding in-video dropouts and interaction peaks in online lecture videos.
Juho Kim, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, & Robert C. Miller.
L@S 2014: ACM Conference on Learning at Scale. (35% acceptance rate, 10 pages)

[c.6] How video production affects student engagement: an empirical study of MOOC videos.
Philip J. Guo, **Juho Kim**, & Rob Rubin.
L@S 2014: ACM Conference on Learning at Scale. (35% acceptance rate, 10 pages)

[c.5] Community clustering: leveraging an academic crowd to form coherent conference sessions.
Paul André, Haoqi Zhang, **Juho Kim**, Lydia B. Chilton, Steven P. Dow, & Robert C. Miller.
HCOMP 2013: AAAI Conference on Human Computation & Crowdsourcing. (30% acceptance rate, 8 pages)
Notable Paper Award.

[c.4] Cobi: a community-informed conference scheduling tool.
Juho Kim, Haoqi Zhang, Paul André, Lydia B. Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C. Miller, & Steven P. Dow.
UIST 2013: ACM Symposium on User Interface Software and Technology. (20% acceptance rate, 10 pages)

[c.3] Social visualization and negotiation: effects of feedback configuration and status.
Michael Nowak, **Juho Kim,** Nam Wook Kim, & Clifford Nass.
CSCW 2012: ACM Conference on Computer-Supported Cooperative Work.
(40% acceptance rate, 10 pages, with revise-and-resubmit cycle)

[c.2] How a freeform spatial interface supports simple problem solving tasks.
Eser Kandogan, **Juho Kim,** Thomas P. Moran, & Pablo Pedemonte.
CHI 2011: ACM Conference on Human Factors in Computing Systems. (26% acceptance rate, 10 pages)

[c.1] Evolutionary topic maps.
Juho Kim, Won-Wook Hong, & Robert Ian McKay.
KHCI 2009: HCI Korea Conference. (5 pages)

Conference Papers in Submission (under review)

[u.6] RIMES: Embedding Interactive Multimedia Exercises in Lecture Videos.
Juho Kim, Elena L. Glassman, Andrés Monroy-Hernández, Meredith Ringel Morris.
In submission to CHI 2015 (under review)

[u.5] Mudslide: A Spatially Anchored Census of Student Confusion for Online Lecture Videos.
Elena L. Glassman, **Juho Kim**, Andrés Monroy-Hernández, Meredith Ringel Morris.
In submission to CHI 2015 (under review)

[u.4] Factful: Engaging Taxpayers in the Public Discussion of a Government Budget.
Juho Kim, Eun-Young Ko, Jonghyuk Jung, Chang Won Lee, Nam Wook Kim, Jihee Kim.
In submission to CHI 2015 (under review)

[u.3] BudgetMap: Supporting Issue-Driven Navigation for Government Budget.
Nam Wook Kim, Chang Won Lee, Jonghyuk Jung, Eun-Young Ko, **Juho Kim**, Jihee Kim.
In submission to CHI 2015 (under review)

[u.2] Mining Attendee Data to Improve Academic Conferences.
Anant Bhardwaj, **Juho Kim**, Steven P. Dow, David Karger, Sam Madden, Robert C. Miller, Haoqi Zhang.
In submission to CHI 2015 (under review)

[u.1] Apparition: Crowdsourced User Interfaces That Come To Life As You Sketch Them.
Walter Lasecki, **Juho Kim**, Nick Rafter, Onkur Sen, Jeffery Bigham, Michael Bernstein.
In submission to CHI 2015 (under review)

Posters, Demos, and Workshop Papers

[p.15] Workshop on Connecting Collaborative & Crowd Work with Online Education.
Joseph Jay Williams, Markus Krause, Praveen Paritosh, Jacob Whitehill, Justin Reich, **Juho Kim**, Piotr Mitros, & Neil Heffernan.
CSCW 2015. (to appear)

[p.14] Leveraging video interaction and content to improve video learning.
Juho Kim.
HCOMP 2014 Workshop on Crowdsourcing, Online Education, and Massive Open Online Courses.

[p.13] Leveraging video interaction and content to improve video learning.
Juho Kim, Shang-Wen (Daniel) Li, Carrie J. Cai, Krzysztof Z. Gajos, & Robert C. Miller.
CHI 2014 Workshop on Learning Innovation at Scale.

[p.12] Interaction peaks and data-driven interfaces for online lecture videos.
Juho Kim.
Quanta-CSAIL 2014 workshop poster.

[p.11] Enhancing how-to videos with crowdsourcing and learnersourcing.
Juho Kim.
Quanta-CSAIL 2014 workshop poster.

[p.10] Cobi: community-informed conference scheduling.
Juho Kim, Haoqi Zhang, Paul André, Lydia B. Chilton, Anant Bhardwaj, David Karger, Steven P. Dow, & Robert C. Miller.
HCOMP 2013 demo.

[p.9] User interfaces and crowdsourcing workflows for enhancing the video learning experience.
Juho Kim, Phu Nguyen, Robert C. Miller, & Krzysztof Z. Gajos.

SoCS 2013 Doctoral Symposium.

[p.8] Learnersourcing subgoal labeling to support learning from how-to videos.
Juho Kim, Robert C. Miller, & Krzysztof Z. Gajos.
CHI2013 Extended Abstracts. (32% acceptance rate)

[p.7] ToolScape: enhancing the learning experience of how-to videos.
Juho Kim.
CHI2013 Extended Abstracts. (32% acceptance rate)
2nd place, Student Research Competition.

[p.6] Generating annotations for how-to videos using crowdsourcing.
Phu Nguyen, **Juho Kim**, & Robert C. Miller.
CHI2013 Extended Abstracts. (32% acceptance rate)

[p.5] Cobi: communitysourcing large-scale conference scheduling.
Haoqi Zhang, Paul André, Lydia Chilton, **Juho Kim**, Steven P. Dow, Robert C. Miller, Wendy MacKay, & Michel Beaudouin-Lafon.
CHI2013 Interactivity. (32% acceptance rate)

[p.4] Mechanical Turk is Not Anonymous.
Matthew Lease, Jessica Hullman, Jeffrey P. Bigham, Michael S. Bernstein, **Juho Kim**, Walter S. Lasecki, Saeideh Bakhshi, Tanushree Mitra, & Robert C. Miller.
Social Science Research Network (SSRN) Online, March 6, 2013. SSRN ID: 2228728.

[p.3] Photoshop with friends: a synchronous learning community for graphic design.
Juho Kim, Ben Malley, Joel Brandt, Mira Dontcheva, Diana Joseph, Krzysztof Z. Gajos, & Robert C. Miller.
CSCW 2012 Interactive Demo.

[p.2] Crowdsourcing interface for collecting correspondences of web pages.
Juho Kim, Ranjitha Kumar, & Scott R. Klemmer.
UIST 2009 Poster.

[p.1] Automatic retargeting of web page content.
Ranjitha Kumar, **Juho Kim,** & Scott R. Klemmer.
CHI 2009 Extended Abstracts.

AWARDS & HONORS

Honorable Mention Award Apr.2014
CHI 2014. Among the top 5% of all submissions. [c.9]

Honorable Mention Award Apr.2014
CHI 2014. Among the top 5% of all submissions. [c.8]

Notable paper award Nov.2013
HCOMP 2013. [c.5]

2nd Place, Student Research Competition May 2012
CHI 2013, $300 award + $500 support. [p.7]

1st Place Technical Lecture Award Jan.2012
The 8th Annual Young Generation Technical and Leadership Conference

Best Talk Award Jun.2011
Samsung Scholarship Open Talk at the 2011 Samsung Scholarship Academic Camp

The Samsung Scholarship 2010–2015

Tuition and living costs covered for Ph.D. studies ($50K per year, for 5 years)

The Samsung Scholarship 2008–2010

Tuition and living costs covered for Master's studies ($50K per year, for 2 years)

Independent Study Scholarship Oct.2007

$1K Research grant awarded by Center for Teaching & Learning, Seoul National University

Seoul National University Scholarship Aug.2001–Aug.2004

Merit-based scholarships for 6 semesters

TEACHING & MENTORING

Instructor

- User Interface Design & Implementation (MIT 6.813/6.831) Spring 2015
  Delivering lectures, designing and facilitating in-class activities, hiring and supervising teaching assistants, and redesigning curriculum, problem sets, and final projects. Co-instructor: Prof. Robert C. Miller

Teaching Assistant

- User Interface Design & Implementation (MIT 6.813/6.831) Spring 2012
  Mentored student teams in semester-long interface design projects. Graded research and programming assignments, problem sets, and quizzes. Instructor: Prof. Robert C. Miller

Research Mentoring

- Peter Githaiga (UROP, MIT undergraduate) Nov.2014–Present
  Integrating LectureScape into the edX platform. Co-mentoring with Piotr Mitros at edX.
- Nam Wook Kim (Harvard SEAS Ph.D. student),
  Eun-Young Ko, Chang Won Lee, Jonghyuk Jung (KAIST undergraduates) Dec.2013–Present
  BudgetWiser: Crowdsourcing budgetary opinions and promoting online discussions.
- Sarah Weir (Super UROP, MIT undergraduate) Sep.2013–May 2014
  Applied learnersourcing to label subgoal structure from how-to videos ([c.13]). Sarah won 2nd place in Student Research Competition at CHI2014 & Robert M. Fano UROP Award at MIT EECS.
- Carolyn Chang & Danny Sanchez (MIT undergraduates) Jul.2013–Aug.2013
  Lecture video annotation tool (part of 6.mitx).
- Megan O'Leary & Joseph Hanley (MIT undergraduates) Jul.2013–Aug.2013
  edX student activity visualization (part of 6.mitx).
- Michelle Johnson & Ashley Cho (MIT undergraduates) Jul.2013–Aug.2013
  edX grade distribution visualization (part of 6.mitx).
- Phu Nguyen (Super UROP, MIT undergraduate) Sep.2012–May 2013
  Crowdsourcing step-by-step information from how-to videos ([c.9] and [p.6]).

INVITED TALKS

Data-driven interaction techniques for improving navigation of educational videos

- Boston University Image and Video Computing group seminar Oct. 27th, 2014
- HarvardX Research Colloquium Sep. 26th, 2014

The future of video

• Samsung Economics Research Institute	Dec. 30th, 2013

Video learning at scale with crowdsourcing and learnersourcing

• UVR Lab, Graduate School of Culture Technologies, KAIST	Dec. 27th, 2013
• Dept. of Knowledge Service Engineering, KAIST	Dec. 23rd, 2013
• UI/UX Inventor Club, Korea	Dec. 20th, 2013
• Korean Society of Design Science Seminar Series, Sungkyunkwan University	Dec. 19th, 2013
• HCI Lab, Seoul National University	Dec. 17th, 2013

Enhancing the learning experience of how-to videos

• Graphics Group, MIT CSAIL	Apr. 10th, 2013
• edX Learning Sciences Team	Mar. 7th, 2013
• Computer Science Department, KAIST	Dec. 21st, 2012
• ROSAEC Seminar Series, Seoul National University	Dec. 18th, 2012
• Epoch Foundation @ MIT	Oct. 17th, 2012

Online education and massive open online courses (MOOCs)

• Open Entrepreneur Center	Dec. 27th, 2012
• Todam: Boston Korean Students Meeting	Oct. 13th, 2012
• Samsung Scholarship Open Talk	Jun. 26th, 2012

Crowdsourcing: engineering collective intelligence

• Korean Society of Design Science Seminar Series, Yonsei University	Dec. 20th, 2012
• Department of Software, Sungkyunkwan University	Jul. 3rd, 2012
• Young Generation Technical and Leadership, KSEA	Jan. 6th, 2012
• Todam: Boston Korean Students Meeting	Oct. 8th, 2011
• Samsung Scholarship Open Talk	Jun. 29th, 2011

Creativity support tools

• HCI Lab, Seoul National University	Jan. 4th, 2011
• Department of Information Convergence at Seoul National University GSCST	Jan. 5th, 2011

SELECTED PRESS

Forbes | 2014.08.11

MIT Team Turns 6.9 Million Clicks Into Insights To Improve Online Education
http://www.forbes.com/sites/peterhigh/2014/08/11/mit-team-turns-6-9-million-clicks-into-insights-to-improve-online-education/

eCampus News | 2014.08.05

Learning from MOOC mistakes, one click at a time
http://www.ecampusnews.com/top-news/mooc-learning-767/

TICBeat | 2014.08.02 (Spanish)

El MIT investiga el camino hacia el aprendizaje online más eficiente
http://www.ticbeat.com/tecnologias/mit-investiga-aprendizaje-onine-eficiente/

Bostinno | 2014.07.31

MIT-Spun 'YouTube for MOOCs' is Solving a Major Problem Plaguing Online Education
http://bostinno.streetwise.co/2014/07/31/how-do-online-learners-watch-videos-lecturescape-mits-youtube-for-moocs-516442/

News Peppermint | 2014.07.29 (Korean)

온라인 교육의 성패를 가르는 요인들
http://newspeppermint.com/2014/07/29/online_education/

MIT News Office | 2014.07.28

What 6.9 million clicks tell us about how to fix online education
http://newsoffice.mit.edu/2014/what-69-million-clicks-tell-us-about-how-fix-online-education

PROFESSIONAL SERVICE

Program Committee – Learning at Scale 2015
Posters Co-Chair – UIST 2015
Conference Organizer – Webmaster (CHI 2015)
Conference Organizer – Scheduling + communitysourcing (CHI 2013-2014, CSCW 2014-2015)
Workshop Organizer – Connecting Collaborative & Crowd Work with Online Education (CSCW 2015)
Workshop Organizer – CrowdCamp (HCOMP 2013-2014)
Reviewer – CHI 2008-2015, UIST (2012 & 2014), CSCW (2008 & 2011-2015), ICWSM 2014, EDM 2014, MobileHCI 2014, HCOMP 2013, IWIC 2009
Student Volunteer – CHI 2010-2012, UIST 2012, CSCW 2013, APCHI 2008
Head of Digital Learning Subcommittee – MIT Graduate Student Council May 2013-May 2014
Organizer – Todam interdisciplinary weekly seminar Aug. 2012-Aug.2014
Organizer – MIT CSAIL HCI Seminar Series 2011-Present
Organizer – BostonCHI Labs Research Consortium 2011-Present
President – MIT EECS Korean Graduate Student Association 2012-2013

REFERENCES

Robert C. Miller
Professor, Massachusetts Institute of Technology
rcm@mit.edu

Frédo Durand
Professor, Massachusetts Institute of Technology
fredo@mit.edu

Haoqi Zhang
Assistant Professor, Northwestern University
hq@northwestern.edu

Krzysztof Z. Gajos
Associate Professor, Harvard University
kgajos@eecs.harvard.edu

Meredith Ringel Morris
Senior Researcher, Microsoft Research
merrie@microsoft.com

Philip J. Guo
Assistant Professor, University of Rochester
pg@cs.rochester.edu

Juho Kim - Research Statement

As a human-computer interaction (HCI) researcher, I build interactive technologies powered by large-scale data from users. With an unprecedented scale of data generated by users' interactions with online platforms, there is an opportunity to leverage data to uncover rich and structured information, such as usage patterns, levels of engagement, preferences, and even diverse perspectives and opinions. However, raw interaction data is often noisy and unstructured. Furthermore, tools for collecting, interpreting, and acting on interaction data are rudimentary and ad hoc. As a result, data-driven discoveries rarely result in meaningful changes and often require extremely long cycles to directly impact the user experience. My research tackles this challenge by 1) extracting meaningful patterns from natural interaction traces, 2) eliciting specific information from users by designing microtasks that are inherently meaningful to them, and 3) changing the interface behavior immediately as more data becomes available. These data-driven methods demonstrate how interaction data can be powerful building blocks for enhancing massive-scale learning, planning, discussion, collaboration, and sensemaking online.

Specifically, my research has focused on educational videos in learning at scale platforms. My primary approach has been **learnersourcing**, in which learners collectively generate novel content and interfaces for future learners while engaging in a meaningful learning experience themselves. Millions of learners today use educational videos from online platforms such as YouTube, Khan Academy, Coursera, or edX. I believe learners can be a qualified and motivated crowd who can help improve the content and interfaces. What if learners' collective activity can help generate structured information for videos, such as identifying points of confusion or importance in a video, reconstructing a solution structure from a tutorial, or creating alternate explanations and examples? Learnersourcing can generate such structured information neither experts, nor computers, nor existing crowdsourcing methods can achieve at scale. My research demonstrates that interfaces powered by learnersourced structured information can enhance content navigation, create a sense of learning with others, and ultimately improve learning.

I draw on several fields to design learnersourcing applications: crowdsourcing to collect and handle a large amount of learner input; social computing to promote collaborative improvement; content-based video analysis techniques such as computer vision and natural language processing to complement learner input; and learning science to inform the design of learnersourcing tasks that are pedagogically meaningful. I explore two types of learnersourcing: **passive learnersourcing** uses data generated by learners' natural interaction with the learning platform, and **active learnersourcing** prompts learners to provide specific information.

Passive Learnersourcing: Natural learner interactions improve video learning

I created a thread of research that leverages natural learning interaction data to improve the learning experience, specifically using thousands of learners' second-by-second video player interaction traces (e.g., clicking the play button in the video player). I first conducted exploratory analyses to better understand the video clickstream data, and used the findings to design a set of data-driven video interaction techniques to directly help future learners.

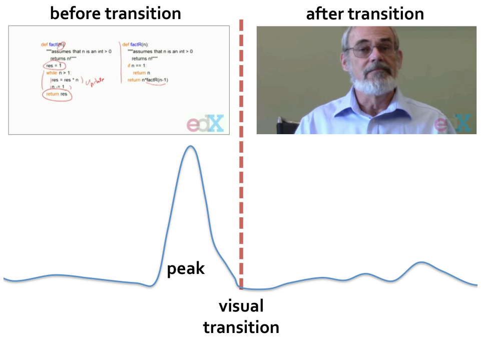
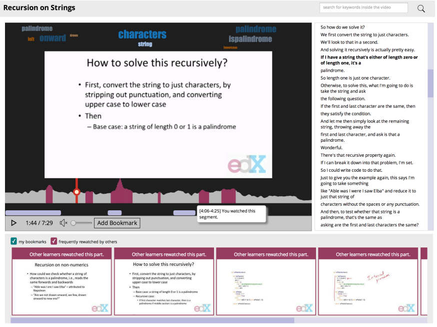


Figure 1. An example interaction peak near a scene transition in a lecture video.

Data analysis of 39 million MOOC video clicks. Exploratory data analyses of four massive open online courses (MOOCs) on the edX platform investigated 39 million video events and 6.9 million watching sessions from over 120,000 learners. Analyzing collective in-video interaction traces revealed video interaction patterns, one of which is *interaction peaks*, a burst of play button clicks around a point in a video indicating points of interest and confusion for many learners. A key observation was that 61% of the peaks accompany visual transitions in the video, e.g., a slide view to an instructor view (Figure 1). Extending this observation, I identified student activity patterns that can explain peaks, including playing from the beginning of a new material, returning to missed content, and replaying a brief segment [1]. A further analysis investigated how video production factors affect video engagement: shorter (less than six minutes long), informal (professors at a desk rather than behind a podium), and web-optimized (rather than in-class lecture captures) videos lead to higher engagement [2]. These analyses have implications for video authoring, editing, and interface design, and provide a richer understanding of video learning on MOOCs.



How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
  - Base case: a string of length 0 or 1 is a palindrome

[4:06-4:25] You watched this segment.

So how do we solve it?
We first convert the string to just characters.
We'll look to that in a second.
And solving it recursively is actually pretty easy.
If I have a string that's either of length zero or of length one, it's a
palindrome.
So length one is just one character.
Otherwise, to solve this, what I'm going to do is take the string and ask
the following question.
If the first and last character are the same, then
they satisfy the condition.
And let me then simply look at the remaining string, throwing away the
first and last character, and ask is that a palindrome.
Wonderful.
There's that recursive property again.
If I can break it down into that problem, I'm set.
So I could write code to do that.
Just to give you the example again, this says I'm going to take something
like "Able was I were I saw Elba" and reduce it to just that string of
characters without the spaces or any punctuation.
And then, to test whether that string is a palindrome, that's the same as
asking are the first and last characters the same?

Other learners rewatched this part.

Recursion on non-numerics

- How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards
  - "Able was I ere I saw Elba" – attributed to Napolean
  - "Are we not drawn onward, we few, drawn onward to new era?"

Other learners rewatched this part.

How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
  - Base case: a string of length 0 or 1 is a palindrome
  - Recursive case:
    - If first character matches last character, then is a palindrome if middle section is a palindrome

Other learners rewatched this part.

Other learners rewatched this part.

Figure 2. LectureScape: lecture video player powered by interaction data.

Data-driven video interface that evolves over time. I took a step further and explored ways to directly improve the video watching experience with interaction data. **LectureScape** (Figure 2) [3] is an enhanced video player for educational content online, powered by data on learners' collective video watching behavior. LectureScape dynamically adapts to thousands of learners' collective video watching patterns to make it easier to rewatch, skim, search, and review. By analyzing the viewing data as well as the content itself, LectureScape introduces a set of interaction techniques that augment existing video interface widgets: a 2D video timeline with an embedded visualization of collective navigation traces; dynamic and non-linear timeline scrubbing; data-enhanced transcript search and keyword summary; automatic display of relevant still frames next to the video; and a visual summary representing points with high learner activity. Participants in a user study commented that "it feels like watching with other students" and it was "more classroom-y" to watch videos with LectureScape, which shows how large-scale interaction data can support social and interactive video learning. This project has been published at UIST 2014 [3], a top-tier venue in HCI, and was featured by over 30 media outlets, including Forbes, eCampusNews, BostonInno, and MIT News Office. I am currently collaborating with edX to integrate the player into the edX platform, and will soon release it as open source.

Adapting interaction behavior to user interest. LectureScape demonstrates an example of how signals of user interest can be used to dynamically improve the user interface. I extended this idea to a scrolling technique for touchscreen devices. The **content-aware kinetic scrolling** (CAKS) technique [4] constructs a degree of interest model from web page content (e.g., number of likes, shares, or recommendations on social media) to dynamically modify the scrolling behavior. CAKS applies additional friction around points of high interest within the page. This draws user's attention to interesting content without cluttering the limited visual space.

Active Learnersourcing: Learner prompts contribute to new learning materials

How-to videos contain worked examples and step-by-step instructions for how to complete a task (e.g., math, cooking, programming, graphic design). My formative study showed the navigational, self-efficacy, and performance benefits of having step-by-step information about the solution. Education research also advocates for exposing the solution structure and shows the learning benefits of having labels for groups of steps (subgoals). However, such information is often not available for most existing how-to video online. I have created scalable methods for extracting steps and subgoals from existing videos, as well as an alternative video player where the solution structure is displayed alongside the video. These techniques actively prompt learners to contribute structured information in an in-video quiz format.

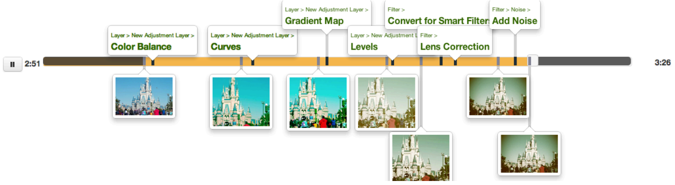


Figure 3. ToolScape: Step-aware tutorial video player.

Extracting step-by-step information from how-to videos. Based on the findings of the study, I built **ToolScape**, a video player that displays step descriptions and intermediate result thumbnails in the video timeline (Figure 3) [5]. To enable non-experts to successfully extract step-by-step structure from existing how-to videos at scale, I designed a three-stage crowdsourcing workflow. It applies temporal clustering, text processing, and visual analysis algorithms to merge crowd output. The workflow successfully annotated 75 cooking, makeup, and Photoshop videos on YouTube of varying styles, with a quality comparable to trained annotators across all domains.

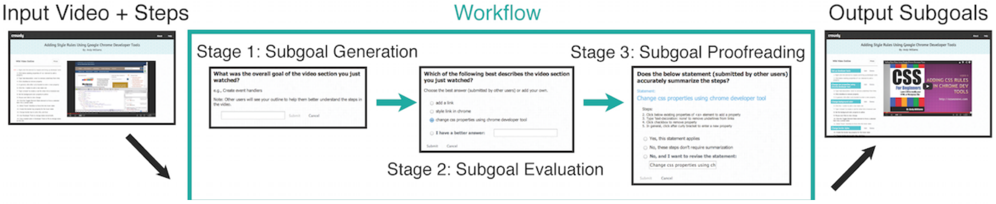

Input Video + Steps
Workflow
Output Subgoals
Stage 1: Subgoal Generation
What was the overall goal of the video section you just watched?
Stage 2: Subgoal Evaluation
Which of the following best describes the video section you just watched?
Stage 3: Subgoal Proofreading
Does the below statement (submitted by other users) accurately summarize the steps?
Change css properties using chrome developer tool
CSS

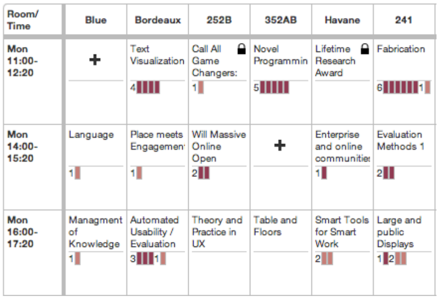
Figure 4. Crowdy: Learnersourcing workflow for summarizing steps in a how-to video.

Learnersourcing section summaries from how-to videos. Taking a step further, we asked if learners, both an intrinsically motivated and uncompensated crowd, can generate summaries of individual steps at scale. This research question resulted in a learnersourcing workflow that periodically prompts learners who are watching the video to answer one of the pre-populated questions, such as “what was the overall goal of the video section you just watched?” (Figure 4) [6]. The system determines which question to display depending on how much information has already been gathered for that section in the video, and the questions are designed to engage learners to reflect on the content. Future learners can navigate the video with the up-to-date solution summary. We deployed **Crowdy**, a live website with the learnersourcing workflow implemented on a set of introductory web programming videos. The 25-day deployment attracted more than 1,200 learners who contributed hundreds of subgoal labels and votes. A majority of learner-generated subgoals were comparable in quality to expert-generated ones, and learners commented that the system helped them grasp the material.

When combined, the two methods (ToolScape for individual steps and Crowdy for subgoal labels) can fully extract a hierarchical solution structure from existing how-to videos. I am currently conducting a controlled study designed to evaluate the motivational and learning benefits of participating in our learnersourcing workflow and interacting with a solution structure generated by learners.

Designing systems for communities and crowds that care

I have also explored other domains beyond education, in which a community of users perform inherently meaningful tasks while collaborating in the sensemaking and content generation process.



Room/ Time	Blue	Bordeaux	252B	352AB	Havane	241
Mon 11:00-12:20	+	Text Visualization 4	Call All Game Changers: 1	Novel Programmin 5	Lifetime Research Award	Fabrication 6 1
Mon 14:00-15:20	Language 1	Place meets Engagemen 1	Will Massive Online Open 2	+	Enterprise and online communitie 1	Evaluation Methods 1 2
Mon 16:00-17:20	Managment of Knowledge 1	Automated Usability / Evaluation 3 1	Theory and Practice in UX	Table and Floors	Smart Tools for Smart Work 2	Large and public Displays 1 2

Figure 5. Cobi: Conference scheduling tool powered by community input.

Community-driven conference scheduling. The **Cobi** project engages an entire academic community in planning a large-scale conference. Cobi elicits community members' preferences and constraints, and provides a scheduling tool that empowers organizers to take informed actions toward improving the schedule [7] (Figure 5). Community members’ self-motivated interactions and inputs guide the conflict resolution and schedule improvements. Because each group within a community has different information needs, motivations, and interests in the schedule, we designed custom applications for different groups: Frenzy [8] for program committee members to group and label papers sharing a common theme; authorsourcing [7, 9] for paper authors to indicate papers relevant to theirs; and Confer [10] for attendees to bookmark papers of interest. Cobi has scheduled CHI and CSCW, two of the largest conferences in HCI, since 2013. It has successfully resolved conflicts and incorporated preferences in the schedule, with input from hundreds of committee members, and thousands of authors and attendees. I was the main designer and developer for the authorsourcing and scheduling applications [7], and helped the design of Frenzy [8] and Confer [10].

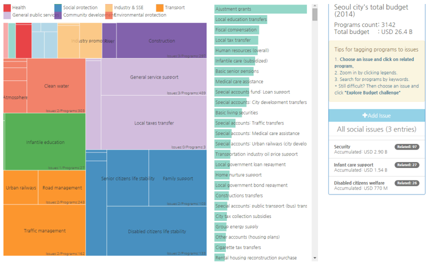
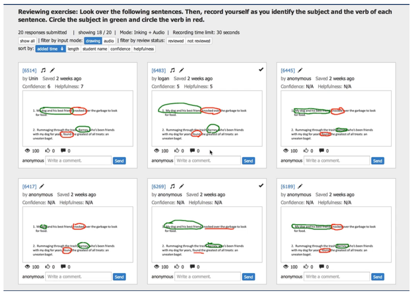


Figure 6. BudgetMap: Visual exploration of government budgets driven by social issues tagged by taxpayers.

Budgetary discussion support for taxpayers. The extensiveness and complexity of a government budget hinder taxpayers from understanding budgetary information and participating in deliberation. We designed interactive and collaborative web platforms in which users can contribute to the overall “pulse” of the public’s understanding and sentiment about budgetary issues. **Factful** [11] is an annotative article reading application that enhances the article with fact-checking support and contextual budgetary information. Users’ fact-checking requests and results are accumulated to help future users engage in fact-oriented discussions. **BudgetMap** (Figure 6) [12] allows users to navigate the government budget with social issues of their interest. Users can make links between government programs and social issues by tagging. In our 5-day live deployment, more than 1,600 users visited the site and made 697 links between social issues and budget programs. The government of South Chungcheong Province in South Korea has decided to officially adopt BudgetMap.

Future Research Agenda

My research has introduced computational mechanisms in which user's lightweight contributions serve a bigger cause: learners improve content and interfaces for future learners, paper authors and attendees help with conference scheduling, and taxpayers issue fact checking requests and link social issues to budget items. My research will continue to lower the bar on individuals’ participation and impact within a community, by engaging them in activities meaningful to both themselves and the community.



Reviewing exercise: Look over the following sentences. Then, record yourself as you identify the subject and the verb of each sentence. Circle the subject in green and circle the verb in red.

20 responses submitted | showing 18 / 20 | Mode: Inking + Audio | Recording time limit: 30 seconds

[6514] by Unin Saved 2 weeks ago Confidence: 6 Helpfulness: 7

[6483] by logan Saved 2 weeks ago Confidence: 5 Helpfulness: 5

[6445] by anonymous Saved 2 weeks ago Confidence: N/A Helpfulness: N/A

[6417] by anonymous Saved 2 weeks ago Confidence: N/A Helpfulness: N/A

[6269] by anonymous Saved 2 weeks ago Confidence: N/A Helpfulness: N/A

[6189] by anonymous Saved 2 weeks ago Confidence: N/A Helpfulness: N/A

1. My dog and his best friend knocked over the garbage to look for food.

2. Rummaging through the trash, Barney, who's been friends with my dog for years, found the greatest of all treats: an uneaten bagel.

show all
| filter by input mode: drawing audio
| filter by review status: reviewed not reviewed
sort by: added time length student name confidence helpfulness
100 0 0
anonymous
Write a comment.
Send

Figure 7. RIMES: Interactive multimedia exercises in lecture videos.

Innovative learnersourcing applications. I plan to push forward the boundaries of learnersourcing by broadening the application scope domain. I have already started exploring with RIMES (Figure 7), a system for easily authoring, recording, and reviewing interactive multimedia exercises embedded in lecture videos [13]. With RIMES, teachers can prompt learners to record their responses to an activity using video, audio, and inking while watching lecture videos. Teachers can then review and interact with all the learners’ responses in an aggregated gallery. I plan to extend RIMES to learnersource alternative self-explanations within a video, which can enable multiple learning paths and personalization for learners. A technical challenge will be in interpreting and labeling multimedia inputs from learners. Another avenue for future research is supporting diverse problem domains: a programming environment that learnersources code snippets and documentations; a writing tool that learnersources alternate expressions and phrases for inspiration and learning; and a graphical design tool that learnersources visual assets others can build on top of.

Learning platforms of the future. While existing online learning platforms have provided access to more learners, the learning experience is still limited to passively watching and answering canned questions. I envision educational technologies that truly scale: a course that is created, organized, and taught entirely by learners. Learners will actively engage in 1) creating various artifacts including quizzes, explanations, and examples, 2) providing feedback and iterating on the artifacts, and 3) labeling the artifacts with metadata for better search and organization. With a learnersourced course that builds itself, learners are at the center of both the instructional design and learning experience.

Science of learning at scale. Despite the rapid growth, we do not yet have good answers to whether, what, how, and why people learn in massive learning platforms. The field desperately needs to develop theories, instructional guidelines, and design principles unique to learning at scale settings. I am interested in building formal and data-driven frameworks for conducting controlled experiments, plugging in modularized content, and immediately modifying the learning interface. The frameworks will spur the design of in vivo experiments, new pedagogical formats, and tools for learning, which in turn can advance the emerging science of learning at scale.

Sociotechnical application design. My research program has uniquely applied insights from crowdsourcing (microtasks and workflow design) to lower the barrier to participation for individuals within a community. I believe this framework has potential for broader societal impact. I plan to design novel coordination and incentive mechanisms for broader domains including civic engagement, health care, and accessibility. I will investigate generalizable design principles and computational methods applicable across multiple domains. Furthermore, I am interested in capturing community interactions beyond clickstream and answers to given prompts. How can we capture discussion, collaboration, decision making, and creative processes by individuals and groups at scale? I will continue to build systems that are used by real people in answering the proposed research questions.

Collaboration. In graduate school, I have been fortunate to work with over 60 collaborators from over 20 different institutions, spanning a variety of domains including machine learning, natural language processing, computer vision, artificial intelligence, systems, communications, education, psychology, and economics. I will continue to aim for more ambitious research goals and impact with my collaborators.

References

[1] **Juho Kim**, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, Robert C. Miller. Understanding In-Video Dropouts and Interaction Peaks in Online Lecture Videos. In Proceedings of *L@S 2014: ACM Conference on Learning at Scale.*

[2] Philip J. Guo, **Juho Kim**, Rob Rubin. How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos. In Proceedings of *L@S 2014: ACM Conference on Learning at Scale.*

[3] **Juho Kim**, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, Robert C. Miller. Data-Driven Interaction Techniques for Improving Navigation of Educational Videos. In Proceedings of *UIST 2014: ACM Symposium on User Interface Software and Technology.*

[4] **Juho Kim**, Amy X. Zhang, Jihee Kim, Robert C. Miller, Krzysztof Z. Gajos. Content-Aware Kinetic Scrolling for Supporting Web Page Navigation. In Proceedings of *UIST 2014: ACM Symposium on User Interface Software and Technology.*

[5] **Juho Kim**, Phu Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, Krzysztof Z. Gajos. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. In Proceedings of *CHI 2014: ACM Conference on Human Factors in Computing Systems.* **Best of CHI Honorable Mention.**

[6] Sarah Weir, **Juho Kim**, Krzysztof Z. Gajos, Robert C. Miller. Learnersourcing Subgoal Labels for How-to Videos. In Proceedings of *CSCW 2015: ACM Conference on Computer-Supported Cooperative Work and Social Computing, to appear.*

[7] **Juho Kim**, Haoqi Zhang, Paul André, Lydia B. Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C. Miller, Steven P. Dow. Cobi: A Community-Informed Conference Scheduling Tool. In Proceedings of *UIST 2013: ACM Symposium on User Interface Software and Technology.*

[8] Lydia Chilton, **Juho Kim**, Paul André, Felicia Cordeiro, James Landay, Dan Weld, Steven P. Dow, Robert C. Miller, Haoqi Zhang. Frenzy: Collaborative Data Organization for Creating Conference Sessions. In Proceedings of *CHI 2014: ACM Conference on Human Factors in Computing Systems.* **Best of CHI Honorable Mention.**

[9] Paul André, Haoqi Zhang, **Juho Kim**, Lydia B. Chilton, Steven P. Dow, Robert C. Miller. Community Clustering: Leveraging an Academic Crowd to Form Coherent Conference Sessions. In Proceedings of *HCOMP 2013: AAAI Conference on Human Computation and Crowdsourcing.* **Notable Paper Award.**

[10] Anant Bhardwaj, **Juho Kim**, Steven P. Dow, David Karger, Sam Madden, Robert C. Miller, Haoqi Zhang. Attendee-sourcing: Exploring the Design Space of Community-Informed Conference Scheduling. In Proceedings of *HCOMP 2014: AAAI Conference on Human Computation and Crowdsourcing.*

[11] **Juho Kim**, Eun-Young Ko, Jonghyuk Jung, Chang Won Lee, Nam Wook Kim, Jihee Kim. Factful: Engaging Taxpayers in the Public Discussion of a Government Budget. *In submission to CHI 2015: ACM Conference on Human Factors in Computing Systems.* (under review)

[12] Nam Wook Kim, Chang Won Lee, Jonghyuk Jung, Eun-Young Ko, **Juho Kim**, Jihee Kim. BudgetMap: Supporting Issue-Driven Navigation for Government Budget. *In submission to CHI 2015: ACM Conference on Human Factors in Computing Systems.* (under review)

[13] **Juho Kim**, Elena L. Glassman, Andrés Monroy-Hernández, Meredith Ringel Morris. RIMES: Embedding Interactive Multimedia Exercises in Lecture Videos. *In submission to CHI 2015: ACM Conference on Human Factors in Computing Systems.* (under review)

Juho Kim - Teaching Statement

My goal in teaching and mentoring is to create an interactive and constructive learning environment for students. I have found it rewarding to see my students learn and grow, and realized that a learning environment makes a big difference in students' experiences. A good environment is more than well-delivered lectures and well-designed curricula: it's a culture. I will carefully build a culture that increases interactivity, promotes learning by doing, and provides feedback. **Interactivity** encourages higher comprehension and reflection, and helps students have more control in their learning. I care about increasing student-teacher, student-student, and student-content interactivity. **Learning by doing** promotes tinkering with physical and digital artifacts, provides an opportunity to turn abstract concepts into concrete examples, and naturally affords an iterative design process where students experience failing fast, often, and softly. Immediate and constructive **feedback** helps students discover missing links in their knowledge and adjust their understanding. I focus on creating an open environment where self- and peer-feedback is encouraged.

My teaching and mentoring experiences, as well as my research on online education, have helped prepare me to cultivate the supportive learning environment and become a good teacher and mentor. I would like to teach, advise, explore novel educational technologies, and provide community support for the next generation of builders and designers of interactive technologies.

Teaching

I will be a co-instructor for MIT's User Interface Design course (6.813/6.831) in Spring 2015, in which we are expecting around 300 undergraduate and graduate students. It is an intro-level Human-Computer Interaction (HCI) course that provides backgrounds in HCI methodology, design process, and hands-on user interface design and implementation experiences with a semester-long group project. It is a rare opportunity as a graduate student at MIT to be an official instructor for a regular course. My responsibilities include delivering lectures, designing and facilitating in-class activities, hiring and supervising teaching assistants, and redesigning problem sets. We are implementing a flipped classroom model in this course, where we ask students to read the course material for the upcoming class, take a short quiz in the beginning of the class, participate in various in-class activities, and attend weekly studio sections for peer feedback on group projects. It will be a great opportunity for me to gain an experience in planning and running a course before starting as faculty, and to practice various pedagogically effective approaches I believe in.

I also served as a teaching assistant for the same course in Spring 2012, for which I mentored 11 student project groups, facilitated multiple rounds of design critique sessions, and graded problem sets and project milestones. During four years in my undergraduate, I have worked as a private tutor in math, computer programming, English, and vocal training. This variety of teaching experiences helped me not only realize my passion in teaching, but also think about how to bring the benefits of one-to-one tutoring into classrooms with hundreds of students and even online classes with hundreds of thousands of students. I aim to apply active learning methods to my teaching, such as in-class discussions, small group tasks, frequent and targeted feedback, and peer instruction, while reducing one-way lecturing as much as possible. I also familiarize myself with novel education technologies and online learning support systems, which may enable more interactive experiences within and outside of the classroom.

Mentoring

I have been fortunate to mentor talented and motivated students in various research projects. I mentored nine MIT undergraduates in learning-related HCI projects, two of whom I mentored for a full academic year with weekly in-person meetings. Sarah Weir designed a learnersourcing workflow for generating summary labels for how-to videos, which resulted in a full paper at CSCW 2015, a premier venue in HCI, with Sarah as the first author. She is also a 2nd place winner in student research competition at CHI 2014, a premier venue in HCI, and received a Robert M. Fano UROP (Undergraduate Research Opportunities Program) award in the EECS department at MIT. Phu Nguyen worked on a crowdsourcing workflow for extracting step-by-step information from how-to videos, which resulted in a full paper at CHI 2014 with Phu as the second author, and a poster at CHI 2013 with Phu as the first author. I also mentored four students at Harvard and KAIST in a multi-institutional research project, BudgetWiser, which attempts to engage the public in a budgetary discussion online. I mentored the students in creating two live web interfaces and submitting two full papers to CHI 2015. Such diverse and successful mentoring experiences were highly rewarding, and I will continue to refine my mentoring approach to help my students grow as independent researchers.

Technologies for Teaching and Learning

I have a strong interest in improving the teaching and learning experience with technology, which also aligns with my own research domain and expertise. I am interested in designing and building interactive exercises, systems for supporting active learning, and reusable infrastructure for courses. An example is RIMES, a system I built for enabling instructors to insert interactive exercises into a lecture video. Students can record their responses with drawing, audio, and video. Instructors can then review the submissions using the gallery interface, provide personalized feedback, and share example answers with the entire class. I plan to actively create and employ such technologies to enhance the learning experience for my students.

Community

Learning comes from not just sitting in classrooms but also from interacting with peers and colleagues in a community. I have taken a leadership role in expanding the HCI community at MIT and in the Boston area. I have co-organized the HCI seminar at MIT CSAIL [http://groups.csail.mit.edu/uid/seminar.shtml] since 2011, where we invited speakers from diverse institutions and research backgrounds. The attendance has been not just from MIT CSAIL, but also from other programs at MIT, as well as local schools and industry labs. I have also served as a co-organizer for BostonCHILabs [http://bostonchilabs.org/] since 2011, an alliance of academic and industry HCI researchers in the Boston area, where I organized various academic and social events for the community. I also co-organized Todam, a weekly interdisciplinary seminar series for Korean students in the Boston area for two years. As faculty, I will continue my effort in building and growing an academic community within the institution and local area.

Example Courses

Having mastered the skills to provide a solid conceptual ground in both computer science and HCI, I am qualified and excited to teach courses in the following areas, as well as a broad range of intro-level CS courses:

- **Human-Computer Interaction:** Possible courses include introduction to HCI, interaction design studio, and graduate-level research topics in HCI. In all these courses I plan to incorporate a project component, where students follow the design process to define a problem, discover user needs, build multiple prototypes and iterate on them, evaluate with end users, and present the demo and findings.

- **Programming / Web Applications**: Possible courses include introduction to programming, software studio, and web applications. Topics would include the basics of programming, algorithms, data structures, and software engineering, as well as modern web development technologies and frameworks. All these courses will be project-based, with an emphasis on modularity, reusability, documentation, collaboration, and open source.

- **Crowdsourcing / Human Computation / Social Computing:** Possible courses include graduate-level research topics in crowdsourcing, human computation, and social computing. The courses will include a survey of key topics and research methods, with an emphasis on paper critiques and peer discussions.

- **Learning at Scale:** This graduate-level course will include a survey of key topics in this emerging area of research. It will examine existing massive open online courses (MOOCs), intelligent tutoring systems, and educational technologies, and encourage students to explore opportunities and limitations of technologies specifically designed for learning at scale. It will cover pedagogical theories, computational technologies, social learning models, and policy and legal issues around learning at scale.

Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos

Juho Kim[1] **Phu Nguyen**[1] **Sarah Weir**[1] **Philip J. Guo**[1,2] **Robert C. Miller**[1] **Krzysztof Z. Gajos**[3]

[1]MIT CSAIL
Cambridge, MA USA
{juhokim, phun, sweir, rcm}@mit.edu

[2]University of Rochester
Rochester, NY USA
pg@cs.rochester.edu

[3]Harvard SEAS
Cambridge, MA USA
kgajos@eecs.harvard.edu

ABSTRACT

Millions of learners today use how-to videos to master new skills in a variety of domains. But browsing such videos is often tedious and inefficient because video player interfaces are not optimized for the unique step-by-step structure of such videos. This research aims to improve the learning experience of existing how-to videos with *step-by-step annotations*.

We first performed a formative study to verify that annotations are actually useful to learners. We created ToolScape, an interactive video player that displays step descriptions and intermediate result thumbnails in the video timeline. Learners in our study performed better and gained more self-efficacy using ToolScape versus a traditional video player.

To add the needed step annotations to existing how-to videos at scale, we introduce a novel crowdsourcing workflow. It extracts step-by-step structure from an existing video, including step times, descriptions, and before and after images. We introduce the Find-Verify-Expand design pattern for temporal and visual annotation, which applies clustering, text processing, and visual analysis algorithms to merge crowd output. The workflow does not rely on domain-specific customization, works on top of existing videos, and recruits untrained crowd workers. We evaluated the workflow with Mechanical Turk, using 75 cooking, makeup, and Photoshop videos on YouTube. Results show that our workflow can extract steps with a quality comparable to that of trained annotators across all three domains with 77% precision and 81% recall.

Author Keywords

Crowdsourcing; how-to videos; video annotation.

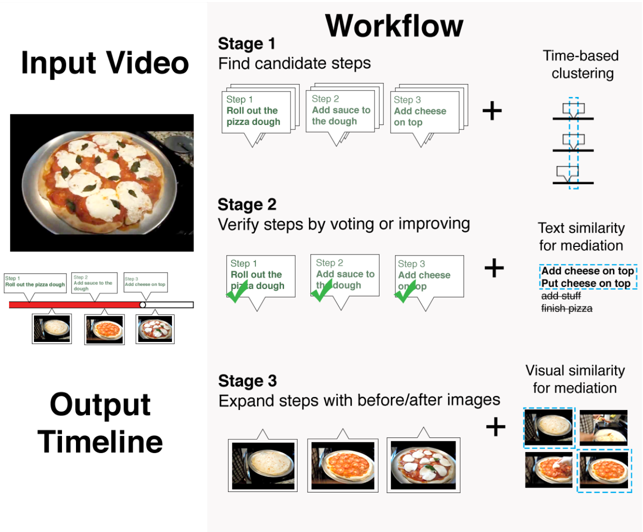
ACM Classification Keywords

H.5.2 User Interfaces: Graphical User Interfaces

INTRODUCTION

How-to videos on the web have enabled millions of learners to acquire new skills in procedural tasks such as folding origami, cooking, applying makeup, and using computer software. These videos have a unique step-by-step structure, which encourages learners to sequentially process and perform steps in the procedure [29]. While most text- and image-based tutorials (e.g., webpages) are naturally segmented into distinct steps, how-to video tutorials often contain a single continuous stream of demonstration. Because comprehensive and accurate step-by-step information about the procedure is often missing, accessing specific parts within a video becomes frustrating for learners. Prior research shows that higher interactivity with the instructional content aids learning [13, 28], and that the completeness and detail of step-by-step instructions are integral to task performance [11].

To better understand the role of step-by-step information in how-to videos, we ran a formative study where learners performed graphical design tasks with how-to videos. For this study, we designed ToolScape, an interactive how-to video player that adds step descriptions and intermediate result thumbnails to the video timeline. Learners using ToolScape showed a higher gain in self-efficacy and rated the quality of their own work higher, as compared to those using an ordinary video player. Moreover, external judges gave higher ratings to the designs produced by learners using ToolScape.



Input Video
Step 1 Roll out the pizza dough
Step 2 Add sauce to the dough
Step 3 Add cheese on top
Output Timeline
Workflow
Stage 1
Find candidate steps
Step 1 Roll out the pizza dough
Step 2 Add sauce to the dough
Step 3 Add cheese on top
+
Time-based clustering
Stage 2
Verify steps by voting or improving
Step 1 Roll out the pizza dough
Step 2 Add sauce to the dough
Step 3 Add cheese on top
+
Text similarity for mediation
Add cheese on top
Put cheese on top
~~add stuff~~
~~finish pizza~~
Stage 3
Expand steps with before/after images
+
Visual similarity for mediation

Figure 1. Our crowdsourcing workflow extracts step-by-step information from a how-to video with their descriptions and before/after images. It features the Find-Verify-Expand design pattern, time-based clustering, and text/visual analysis techniques. Extracted step information can be used to help learners navigate how-to videos with higher interactivity.

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2473-1/14/04..$15.00.
http://dx.doi.org/10.1145/2556288.2556986

Providing such navigation support for how-to videos requires extracting step-by-step information from them. One solution is to ask instructors to include this information at tutorial generation time, but this adds overhead for instructors and does not solve the problem for existing videos. Another approach uses automatic methods such as computer vision. Previous research [3, 26] has shown success in limited domains with extensive domain-specific customization. When working with "videos in the wild", however, vision-based algorithms often suffer from low-resolution frames and a lack of training data. A scalable solution applicable beyond limited task domains and presentation formats is not yet available.

To address the issues of high cost or limited scalability with existing methods, we introduce a crowdsourcing workflow for annotating how-to videos, which includes the Find-Verify-Expand design pattern shown in Figure 1. It collects step-by-step information from a how-to video in three stages: (1) find candidate steps with timestamps and text descriptions, (2) verify time and description for all steps, and (3) expand a verified step with before and after images. The workflow does not rely on domain-specific knowledge, works on top of existing videos, and recruits untrained, non-expert crowd workers. For quality control, the workflow uses time-based clustering, text processing, and visual analysis to merge results and deal with noisy and diverse output from crowd workers.

To validate the workflow with existing how-to videos, we asked crowd workers on Mechanical Turk to annotate 75 YouTube how-to videos spanning three domains: cooking, makeup, and graphics editing software. Results show that the crowd workflow can extract steps with 77% precision and 81% recall relative to trained annotators. Successfully extracted steps were on average 2.7 seconds away from ground truth steps, and external evaluators found 60% of before and after images to be accurately representing steps.

The contributions of this paper are as follows:

- A how-to video player interface and experimental results showing that increased interactivity in a video player improves learners' task performance and self-efficacy.
- A domain-independent crowd video annotation method and the Find-Verify-Expand design pattern for extracting step-by-step task information from existing how-to videos.
- A novel combination of time-based clustering, text processing, and visual analysis algorithms for merging crowd output consisting of time points, text labels, and images.
- Experimental results that validate the workflow, which fully extracted steps from 75 readily available videos on the web across three distinct domains with a quality comparable to that of trained annotators.

RELATED WORK

We review related work in crowdsourcing workflows, video annotation, and tutorials.

Crowdsourcing Workflows

Research on multi-stage crowd workflows inspired the design of our method. Soylent [5] has shown that splitting tasks into the Find-Fix-Verify stages improves the quality and accuracy of crowd workers' results. Other multi-stage crowdsourcing workflows were designed for nutrition information retrieval from food photos [24], activity recognition from streaming videos [21], and search engine answer generation [6]. These applications demonstrated that crowdsourcing can yield results comparable to those of experts at lower cost. Our work contributes to this line of research a novel domain, video annotation, by extending [18] and [23].

Video Annotation Methods

This work focuses on providing a scalable and generalizable video annotation solution without relying on trained annotators, experts, or video authors. Video annotation tools capture moments of interest and add labels to them. Many existing tools are designed for dedicated annotators or experts in limited context. Domain-specific plug-ins [8, 14, 15] automatically capture task information, but require direct access to internal application context (e.g., Photoshop plug-ins accessing operation history). But plug-ins do not exist for most procedural tasks outside of software applications (e.g., makeup), which limits the applicability of this method.

Crowdsourcing video annotation has recently gained interest as a cost-effective method without relying on experts while keeping humans in the loop. Existing systems were designed mostly to collect training data for object recognition [31], motion tracking [30], or behavior detection [25]. Rather than use crowdsourcing to support qualitative researchers, this work supports end users learning from videos. Adrenaline [4] uses crowdsourcing to find the best frame from a video in near real-time. While [4] and our work both aim to detect a time-specific event from a video stream, our work additionally labels the event and expands to capture surrounding context.

Interactive Tutorials

In designing user interfaces for instructional videos, higher interactivity with the content has been shown to aid learning [13, 28]. Tversky et al. [28] state that "stopping, starting and replaying an animation can allow reinspection", which in turn can mitigate challenges in perception and comprehension, and further facilitate learning. Semantic indices and random access have been shown to be valuable in video navigation [32, 22], and the lack of interactivity has been deemed a major problem with instructional videos [16]. This work introduces a user interface for giving learners more interactivity in video navigation, and a crowdsourcing method for acquiring metadata handles to create such an interface at scale.

Recent systems create interactive tutorials by either automatically generating them by demonstration [8, 14], connecting to examples [26], or enhancing the tutorial format with annotated information [8, 9, 18, 20]. Our crowdsourcing workflow can provide annotations required to create these interfaces and further enable new ways to learn from tutorials.

EFFICACY AND CHALLENGES OF VIDEO ANNOTATIONS

To motivate the design of our crowdsourcing workflow for how-to video annotations, we first performed a formative study to (1) verify that annotations are actually useful to

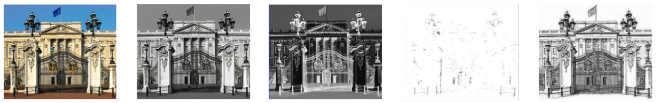


Figure 2. Progress in many how-to videos is visually trackable, as shown in screenshots from this Photoshop how-to video. Adding step annotations to videos enables learners to quickly scan through the procedure.

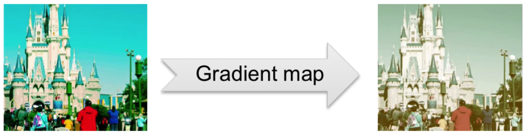


Figure 3. How-to videos often contain a series of task steps with visually distinct before and after states. Here the author applied the "Gradient map" tool in Photoshop to desaturate the image colors.

learners, and (2) reveal the challenges of manually annotating videos and show the need for a more scalable technique.

Annotations on How-To Videos

How-to videos often have a well-defined step-by-step structure [15]. A *step* refers to a low-level action in performing a procedural task. Literature on procedural tasks suggests that step-by-step instructions encourage learners to sequentially process and perform steps in the workflow [29] and improve task performance [11]. Annotations can make such structure more explicit. In this paper, we define *annotation* as the process of adding step-by-step information to a how-to video. In determining which information to annotate, we note two properties of procedural tasks. First, for many domains, task states are visually distinct in nature, so progress can be visually tracked by browsing through a video (Figure 2). Examples include food in cooking videos, a model's face in makeup videos, and an image being edited in Photoshop videos. Second, how-to videos contain a sequence of discrete steps that each advance the state of the task (Figure 3). Our annotation method uses these two properties to accurately capture a sequence of steps, extracting timestamps, textual descriptions, and before and after images for each step.

We manually created a corpus of annotations for 75 how-to videos in three procedural task domains: cooking, applying makeup, and using Photoshop. We used this corpus to create our interface in the formative study, and ground truth data for evaluating our crowdsourcing workflow. We collected videos from YouTube's top search results for "[domain] [task name]" (e.g., "cooking samosa", "Photoshop motion blur").

Annotation-Aware Video Player: ToolScape

To display step annotations, we created a prototype video player named ToolScape. ToolScape augments an ordinary web-based video player with a rich timeline containing links to each annotated step and its respective before and after thumbnail images (Figure 4). ToolScape is a Javascript library that manages a timestamped list of steps and before/after images, which can connect to any embedded video player with a "play from this time point" Javascript API call.

In the timeline, the top and bottom streams represent annotated steps and thumbnail images from the video, respectively (Figure 4(a), (c)). Clicking on a step or image moves the video player's slider to 5 seconds before the moment it occurred. The 5-second buffer, determined from pilot testing, helps learners catch up with the context preceding the indicated moment. Finally, ToolScape supports annotations of "dead times" at the beginning and end of videos (Figure 4(b)), which often contain introductory or concluding remarks. Pilot user observations showed that learners often skip to the main part of the tutorial. In our manually annotated video corpus, on average, 13.7% of time at the beginning and 9.9% at the end were "dead times" with no task progress.

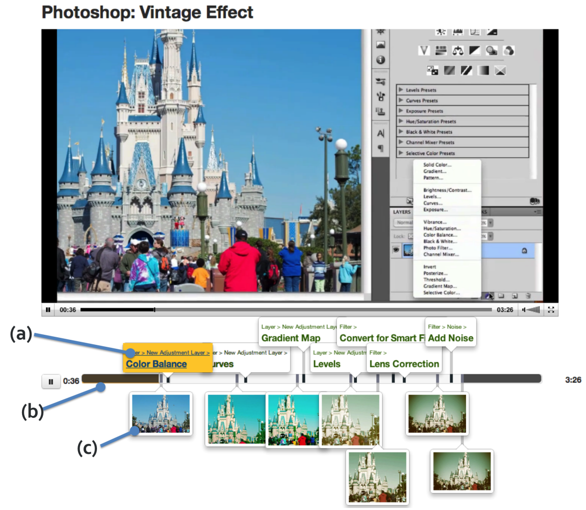

Photoshop: Vintage Effect
00:36
03:26
(a)
(b)
(c)
Layer > New Adjustment Layer >
Color Balance
Layer > New Adjustment Layer >
Curves
Layer > New Adjustment Layer >
Gradient Map
Layer > New Adjustment Layer >
Levels
Filter >
Convert for Smart Filters
Filter >
Lens Correction
Filter > Noise >
Add Noise
0:36
3:26

Figure 4. ToolScape augments a web-based video player with an interactive timeline. Annotations are shown above the timeline (a), screenshots of intermediate states are shown below the timeline (c), and the gray regions at both ends (b) show "dead times" with no meaningful progress (e.g., waiting for Photoshop to launch).

Formative Study Design

To assess the effects of step annotations, we ran a formative study on novice Photoshop learners watching how-to videos on image manipulation tasks. We compared the experiences of learners using ToolScape and a baseline video player without the interactive timeline. We hypothesized that interacting with step annotations provided by ToolScape improves both task performance and learner satisfaction. Specifically:

H1 Learners complete design tasks with a higher self-efficacy gain when watching how-to videos with ToolScape.

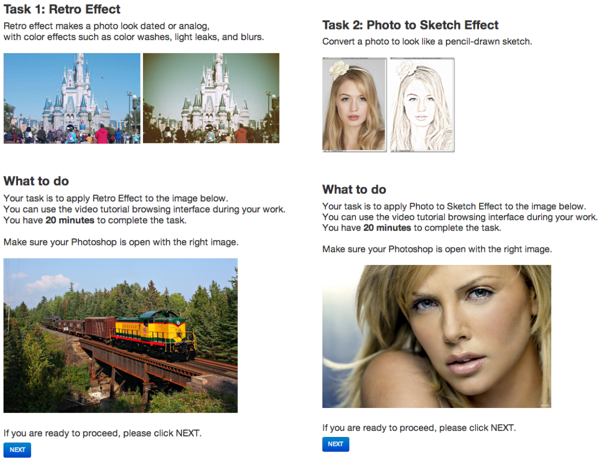
H2 Learners' self-rating of the quality of their work is higher when watching with ToolScape.

H3 Learners' designs when watching with ToolScape are rated higher by external judges.

H4 Learners show higher satisfaction with ToolScape.

H5 Learners perceive design tasks to be easier when watching with ToolScape.

In addition to external ratings (H3), our measures of success include self-efficacy (H1) and self-rating (H2). In the context of how-to videos, these measures are more significant than



Task 1: Retro Effect

Retro effect makes a photo look dated or analog,
with color effects such as color washes, light leaks, and blurs.

What to do

Your task is to apply Retro Effect to the image below.
You can use the video tutorial browsing interface during your work.
You have **20 minutes** to complete the task.

Make sure your Photoshop is open with the right image.

If you are ready to proceed, please click NEXT.

NEXT

Task 2: Photo to Sketch Effect

Convert a photo to look like a pencil-drawn sketch.

What to do

Your task is to apply Photo to Sketch Effect to the image below.
You can use the video tutorial browsing interface during your work.
You have **20 minutes** to complete the task.

Make sure your Photoshop is open with the right image.

If you are ready to proceed, please click NEXT.

NEXT

Figure 5. These task instructions are shown before the participant starts working on their image manipulation task. It includes a description of the effect to be implemented and a before-and-after example image pair.

just user preference. Educational psychology research shows that self-efficacy, or confidence in application of skills, is an effective predictor of motivation and learning [2, 33]. Positive self-rating has also been shown to accurately predict learning gains [27]. Finally, we chose not to count errors made in repeating tutorial steps as in [8], because our goal was to help users explore and learn new skills in open-ended design tasks.

Participants: We recruited twelve participants through university mailing lists and online community postings. Their mean age was 25.2 ($\sigma = 3.2$), with 8 males and 4 females. Most rated themselves as novice Photoshop users, but all had at least some experience with Photoshop. They received $30 for up to two hours of participation, on either a Mac or PC.

Tasks and Procedures: Our study had 2 x 2 conditions: two tasks each using ToolScape and baseline video players. We used a within-subject design with interface, task, and order counterbalanced. Each participant performed two image manipulation tasks in Photoshop: applying retro effect and transforming a photo to look like a sketch. In both interface conditions, we provided participants with the same set of how-to videos; the interface was the only difference. In addition, we disallowed searching for other web tutorials to ensure that any effect found in the study comes from the interaction method, not the content.

After a tutorial task covering all features of the video player interface, we asked participants self-efficacy questions adapted from Dow et al. [10], whose study also measured participants' self-efficacy changes in a design task. The questions asked: On a scale of 1 (not confident at all) to 7 (very confident), how confident are you with…

- solving graphic design problems?
- understanding graphic design problems?
- applying design skills in practice?
- incorporating skills from video tutorials in your design?

Next, participants attempted two 20-minute image manipulation tasks in Photoshop, with instructions shown in Figure 5. Participants could freely browse and watch the 10 how-to videos we provided (with annotations in the ToolScape condition). After each task, we asked questions on task difficulty, self-rating, and interface satisfaction. We also asked the self-efficacy questions again to observe any difference, followed by a 15-minute open-ended interview.

Finally, we asked four external judges to evaluate the quality of all transformed images by ranking them, blind to user and condition. They ranked the images from best to worst, based on how well each participant accomplished the given task.

Formative Study Results

H1 (higher self-efficacy for ToolScape) is supported by our study. For the four self-efficacy questions, we take the mean of the 7-point Likert scale ratings as the self-efficacy score. The participants' mean initial score was 3.8; with the baseline video player, the score after the task was 3.9 (+0.1) whereas with ToolScape the score was 5.2 (+1.4), which meant that learners felt more confident in their graphical design skills after completing tasks with ToolScape. (For H1, H2, and H4, differences between interfaces were significant at $p<0.05$ using a Mann-Whitney U test.)

H2 (higher self-rating for ToolScape) is supported. Participants rated their own work quality higher when using ToolScape (mean rating of 5.3) versus baseline (mean of 3.5).

H3 (higher external judge rating for ToolScape) is supported. The overall ranking was computed by taking the mean of the four judges' ranks. The mean rankings (lower is better) for output images in the ToolScape and Baseline conditions were 5.7 and 7.3, respectively. A Wilcoxon Signed-rank test indicates a significant effect of interface (W=317, Z=-2.79, $p<0.01$, r=0.29). Furthermore, nine of the twelve participants produced higher-rated images with ToolScape. The ranking method yielded high inter-rater reliability (Krippendorff's alpha=0.753) for ordinal data.

H4 (higher satisfaction with ToolScape) is supported. Mean ratings for ToolScape and Baseline were 6.1 and 4.5, respectively.

H5 (easier task difficulty perception for ToolScape) is not supported: The mean ratings for ToolScape and Baseline were 4.0 and 3.7, respectively. Combined with H2 and H3, this might indicate that participants did not find the tasks easier yet still produced better designs with greater confidence.

In conclusion, ToolScape had a significant effect on learners' belief in their graphical design skills and output quality. They also produced better designs as rated by external judges. Note that participants were watching the same video content in both conditions. Thus, the video annotation browsing interface affected design outcomes. Participants especially enjoyed being able to freely navigate between steps within a video by clicking on annotations.

Lessons for Video Browsing Interfaces

The features of ToolScape that provided higher interactivity and non-sequential access were highly rated and frequently used. In participants' responses to the 7-point Likert scale questions on the usability of interface features, the time-marked image thumbnails (6.4) and step links (6.3) were among the highest rated, as well as the graying out of "dead times" with no workflow progress (6.5). Participants noted, "It was also easier to go back to parts I missed.", "I know what to expect to get to the final result.", and "It is great for skipping straight to relevant portions of the tutorial."

All participants frequently used the ability to click on timeline links to navigate directly to specific images and steps. They clicked the interactive timeline links 8.9 times on average ($\sigma = 6.7$) in a single task. We also analyzed the tracking log, which records an event when the user clicks on an interactive link or a pause button, or drags the playhead to another position. The learners watched videos less linearly with ToolScape: The ToolScape condition recorded 150 such events, versus only 96 in the Baseline condition. In ToolScape, 107 out of 150 events were interactive link clicks and 43 were pause button clicks or direct scrubbing on the player. These findings indicate that interactive links largely replaced the need for pause or scrubbing, and encouraged the stepwise navigation of the procedure.

Lessons for How-To Video Annotation

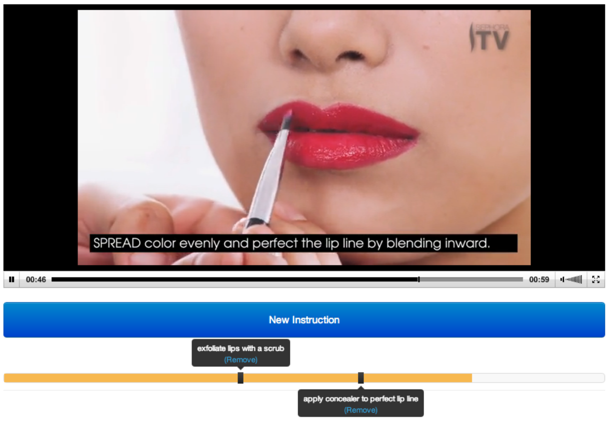
The study results suggest that annotated step information makes how-to videos much more effective for learners. However, the bottleneck is in obtaining the annotations. Here are some lessons from our experience annotating videos by hand:

- Extracting step information from how-to videos involves detecting timing, generating a natural language description of a step, and capturing before and after states.
- It often requires multi-pass watching, which adds to task complexity. Before knowing what each step is, the annotator cannot extract before and after thumbnail images. This experience supports a design choice to split the work into multiple stages so that in each stage, the annotator's attention is focused on a single, simple task.
- Hand annotation is time-consuming. Roughly three times the original video length was required by trained annotators to annotate each how-to video.
- Timing detection is difficult. Sometimes there is an interval between when a step is spoken and demonstrated. Also, if the goal is to find a starting time of a step, the annotator has to watch, verify, and scroll back to mark as a valid step.

These lessons informed the design of our crowdsourced how-to video annotation method, which we now present.

CROWDSOURCING WORKFLOW: FIND-VERIFY-EXPAND

Using lessons from our formative study, we designed a three-stage crowdsourcing workflow for annotating how-to videos with procedural steps, timings, textual descriptions, and before and after thumbnail images. This workflow works with any how-to video regardless of its domain, instructional style, and presentation. It also collects annotations with untrained crowd workers (e.g., workers on Mechanical Turk).

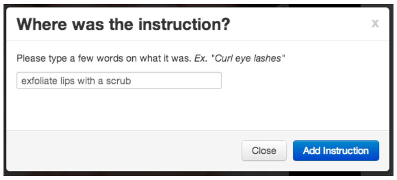


SPREAD color evenly and perfect the lip line by blending inward.

exfoliate lips with a scrub

apply concealer to perfect lip line

Figure 6. In the Find stage, the crowd worker adds new steps to the timeline by clicking on the "New Instruction" button.



Where was the instruction?

Please type a few words on what it was. *Ex. "Curl eye lashes"*

exfoliate lips with a scrub

Figure 7. Upon clicking on the "New Instruction" button, a popup window asks the worker to describe what the step is about in free-form text.

Inspired by crowd design patterns that segment a bigger task into smaller micro-tasks [5], our workflow decomposes the annotation task into three stages and each video into shorter segments. This design addresses the task complexity and multi-pass overhead problems of manual annotation.

We developed a generalizable crowd workflow pattern called **Find-Verify-Expand** (Figure 1) for detecting temporal and visual state changes in videos, such as steps in a how-to video, highlights from a sports game, or suspicious incidents from a surveillance video. The unique *Expand* stage captures surrounding context and causal relationships (e.g., before/after images for a step in a how-to video) by expanding on the detected event (e.g., a step in a how-to video). To better handle crowd output coming from timing detection and image selection, we apply clustering algorithms and text and visual analysis techniques to intelligently merge results from workers.

Stage 1: FIND candidate steps

This crowd task collects timestamps and text descriptions for possible steps from a video segment. While watching the video, the worker adds a step by clicking on the "New Instruction" button every time the instructor demonstrates a step (Figure 6). Each time the worker clicks on the button, the task prompts the worker to describe the step in free-form text (Figure 7). The same segment is assigned to three workers, whose results get merged to create candidate steps.

Pre-processing: A video is segmented into one-minute chunks. We learned from pilot runs that longer video segments lead to lower annotation accuracy toward the end and slower responses on Mechanical Turk. However, a drawback in using segmented video is the possibility of missing steps

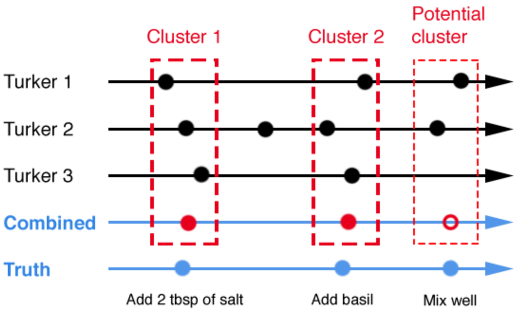
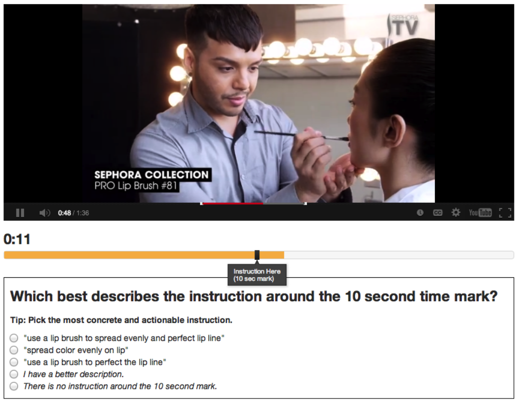


Figure 8. Our clustering algorithm groups adjacent time points into a candidate step. It further adds a potential cluster as a candidate, which might turn out to be a proper step once checked in the Verify stage. This inclusive strategy mitigates the effect of clustering errors.



Which best describes the instruction around the 10 second time mark?

Tip: Pick the most concrete and actionable instruction.

○ "use a lip brush to spread evenly and perfect lip line"
○ "spread color evenly on lip"
○ "use a lip brush to perfect the lip line"
○ *I have a better description.*
○ *There is no instruction around the 10 second mark.*

Figure 9. The Verify stage asks the worker to choose the best description of a candidate step. The options come from workers in the Find stage. Additional default options allow the worker to either suggest a better description or mark the step as invalid.

near segment borders. We address this issue by including a five-second overlap between segments, and attaching the final segment to the prior one if it is shorter than 30 seconds.

Task Design: For quality control, the task first ensures that the user has audio by giving a test that asks the worker to type in a word spoken from an audio file. Our pilot runs showed that labeling accuracy drops significantly when the worker does not listen to audio. Secondly, we disable the Submit button until the video playhead reaches the end to ensure that the worker watches the entire segment. Finally, when the worker clicks on the "New Instruction" button, the video pauses and a dialog box pops up to ask what the step was. Our initial version simply added a tick on the timeline and continued playing without pausing or asking for a label. But this resulted in workers clicking too many times (as many as 100 for a 60-second chunk) without thinking. The prompt adds self-verification to the task, which encourages the worker to process the workflow by each step. The prompt also includes an example label to show the format and level of detail they are expected to provide (Figure 7).

Post-Processing: The workflow intelligently merges results from multiple workers to generate step candidates. To cluster nearby time points given by different workers into a single step, we use the DBSCAN clustering algorithm [12] with a timestamp difference as the distance metric. The clustering idea is shown in Clusters 1 and 2 in Figure 8. The algorithm takes ϵ as a parameter, which is defined by the maximum distance between two points that can be in a cluster relative to the distance between farthest points. We train ϵ once initially on a small set of pilot worker data and ground truth labels. Our tests show that the values between 0.05 and 0.1 yield high accuracy, regardless of domain or video. We configured the algorithm to require at least two labels in every cluster, similar to majority voting among the three workers who watched the segment. We considered other clustering algorithms such as K-Means, but many require the number of clusters as an input parameter. In video annotation, the number of steps is neither known a priori nor consistent across videos.

Depending on videos and parameters, the DBSCAN algorithm might over-generate (false positive) or under-generate (false negative) clusters. We bias the algorithm to over-generate candidate steps ('potential cluster' in Figure 8) and aim for high recall over high precision, because the first stage is the only time the workflow generates new clusters. We improve the initial clusters in three ways, with the goal of higher recall than precision. First, we take into account the textual labels to complement timing information. The clustering initially relies on workers' time input, but using only time might result in incorrect clusters because steps are distributed unevenly time-wise. Sometimes there are steps every few seconds, and other times there might be no step for a minute. We run a string similarity algorithm between text labels in border points in clusters, to rearrange them to the closer cluster. Second, we break down clusters that are too large by disallowing multiple labels from one worker to be in a cluster. Finally, if there are multiple unclustered points within ϵ between clusters, we group them into a candidate cluster. For each cluster, we take a mean timestamp as the representative time to advance to the Verify stage.

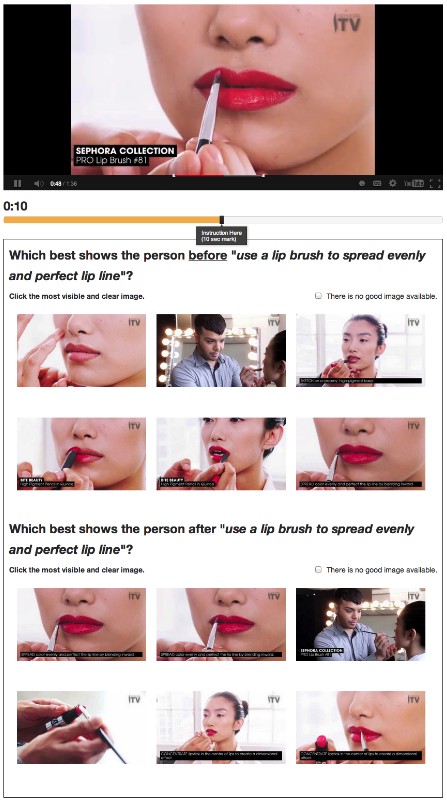
Stage 2: VERIFY steps

Here the worker's verification task is to watch a 20-second clip that includes a candidate step and textual descriptions generated from the prior stage, and vote on the best description for the step (Figure 9). The workflow assigns three workers to each candidate step, whose votes are later merged.

Pre-processing: For each of the candidate steps from Stage 1, the workflow segments videos into 20-second clips around each step (10 seconds before and after).

Task Design: To prevent workers from selecting the first result without reviewing all options, we randomize the order of options presented each time. We also lowercase all labels to prevent capitalized descriptions from affecting the decision. Also, the Submit button becomes clickable only after the worker finishes watching the 20-second clip.

In addition to candidate text descriptions, two additional options are presented to workers: "I have a better description",


SEPHORA COLLECTION
PRO Lip Brush #81
0:48 / 1:36
Instruction Here
(10 sec mark)
BITE BEAUTY
High Pigment Pencil in Quince
SKETCH on a creamy, high-pigment base.
SPREAD color evenly and perfect the lip line by blending inward.

0:10

Which best shows the person before "*use a lip brush to spread evenly and perfect lip line*"?

Click the most visible and clear image. □ There is no good image available.

Which best shows the person after "*use a lip brush to spread evenly and perfect lip line*"?

Click the most visible and clear image. □ There is no good image available.

Figure 10. The Expand stage asks the worker to choose the best before and after images for a step. The worker visually reviews the thumbnail options and clicks on images to decide.

which improves the step label, and "There is no instruction", which filters out false positives from Stage 1.

Post-Processing: Two possible outputs of this stage are 1) finalizing the timestamp and description for a valid step, or 2) removing a false step. The workflow uses majority voting to make the final decision: If two or more workers agreed on a description, it becomes the final choice. If workers are split between three different options, it checks if some of the selected text descriptions are similar enough to be combined. We first remove stop words for more accurate comparisons, and then apply the Jaro-Winkler string matching algorithm [17]. If the similarity score is above a threshold we configured with initial data, we combine the two descriptions with a longer one. If not, it simply picks the longest one from the three. The decision to pick longer description for tie-breaking comes from a pilot observation that longer descriptions tend to be more concrete and actionable (e.g., "grate three cups of cheese" over "grate cheese").

Stage 3: EXPAND with before and after images for steps

This final stage collects the before and after images of a step, which visually summarize its effect. This stage captures surrounding context and causal relationships by expanding on what is already identified in Find and Verify. The worker's task here is to watch a 20-second video clip of a step, and select a thumbnail that best shows the work in progress (e.g., food, face, or Photoshop image) before and after the step (Figure 10). The workflow assigns three workers to each step.

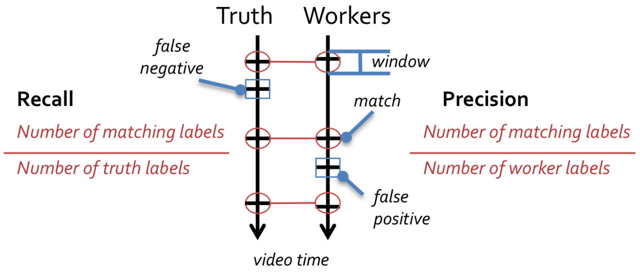


Figure 11. Our evaluation uses the Hungarian method to match extracted and ground truth steps with closest possible timestamp within a 10-second window size. Then we compute precision and recall, which indicate if our workflow over- or under-extracted steps from ground truth.

Pre-processing: This stage uses a 20-second video clip of a step verified in Stage 2, and uses its final text label to describe the step. It creates thumbnails at two-second intervals to present as options, 10 seconds before and after the step.

Task Design: Our initial design asked workers to click when they see good before and after images, but this resulted in low accuracy due to variable response time and the lack of visual verification. We then simplified the task to a multiple choice question. Selecting from static thumbnail images makes the task easier than picking a video frame.

Post-Processing: Similar to the Verify stage, we apply majority voting to determine the final before and after images. For merging and tie breaking, we use Manhattan distance, an image similarity metric that computes pixel differences between two images.

EVALUATION

We deployed our annotation workflow on Mechanical Turk and evaluated on:

- **Generalizability**: Does the workflow successfully generate labels for different types of how-to tasks in different domains with diverse video production styles?
- **Accuracy**: Do collected annotations include all steps in the original video, and avoid capturing too many (false positive) or too few (false negative)? How do textual descriptions generated by crowd workers compare to those generated by trained annotators?

Methodology

We used our workflow and Mechanical Turk to fully extract step information from the 75 how-to videos in our annotation corpus, with 25 videos each in cooking, makeup, and graphics editing software (Photoshop). We did not filter out videos based on use of subtitles, transitions, or audio, to see if our annotation workflow is agnostic to presentation styles. Out of 75 videos in our set, 7 did not have audio, and 27 contained text overlays. For each domain, we picked five tasks to cover diverse types of tasks: Cooking – pizza margherita, mac and cheese, guacamole, samosa, and bulgogi; Makeup – bronze look, reducing redness, smokey eyes, bright lips, and summer glow; Photoshop: motion blur, background removal, photo to sketch, retro effect, and lomo effect. The mean video length was 272 seconds, summing to over 5 hours of videos.

Results

Our evaluation focuses on comparing the quality of step information produced by our crowdsourcing workflow against ground truth annotations from our corpus.

The Turk crowd and trained annotators (two co-authors with educational video research experience) generated similar numbers of steps (Table 1). In Stage 1, 361 one-minute video segments were assigned to Turkers, who generated 3.7 candidate steps per segment, or 53.6 per video. Clustering reduced that to 16.7 steps per video. Stage 2 further removed over-generated steps, resulting in 15.7 per video, which is nearly equivalent to the ground truth of 15 steps per video.

Precision indicates how accurate extracted steps are compared to ground truth, while *recall* shows how comprehensively the workflow extracted ground truth steps (Figure 11). We present precision and recall results considering only the timing of steps (Stage 1), and both the timing and the textual description accuracy (Stage 2). For matching crowd-extracted steps to ground truth steps, we use the Hungarian method [19] whose cost matrix is filled with a time distance between steps.

Evaluating Stage 1: FIND

We consider only precision and recall of times in the Stage 1 evaluation because final textual descriptions are not yet determined. Detecting the exact timing of a step is not straightforward, because most steps take place over a time period, verbal and physical steps are commonly given with a time gap.

To more accurately account for the timing issue, we set a highest threshold in time difference that accepts a Turker-marked point as correct. We set the threshold to 10 seconds, which indicates that a step annotation more than 10 seconds off is discarded. This threshold was based on heuristics from step intervals in our corpus: We hand-annotated, on average, one step every 17.3 seconds in our video corpus (mean video length / number of steps in ground truth = 272/15.7), so a maximum 10-second difference seems reasonable.

The mean distance between ground truth steps and extracted steps (ones within 10 seconds of the ground truth) was only 2.7 seconds. This suggests that for matched steps, the time-based clustering successfully detected the timing information around this distance. When considering only time accuracy, our workflow shows 0.76 precision and 0.84 recall (Table 2).

Evaluating Stage 2: VERIFY

Here we combine the accuracy of both timing and text descriptions. Precision for this stage captures what fraction of steps identified by the workflow are both placed correctly on the time line and whose description reasonably matches the ground truth. The analysis shows 0.77 precision and 0.81 recall over all the videos (Table 2).

For text accuracy measurement, we use the string similarity algorithm to see if a suggested description is similar enough to a description from ground truth. We apply the same threshold as what we configured in the workflow for tie breaking in the Verify stage. The precision and recall both go down when the text similarity condition is added, but precision recovers from the post-processing of steps in this stage. Two enhancements contribute to this recovery: removing steps that workers indicated as “no instruction” from the task, and merging nearby steps that have identical descriptions.

By Turkers	After Stage1	After Stage2	Ground Truth
53.6	16.7	**15.7**	**15.0**

Table 1. The mean number of steps generated by the workflow in each stage. At the end the workflow extracted 15.7 steps per video, which is roughly equivalent to 15.0 from ground truth. Stage 1 clusters the original Turker time points, and Stage 2 merges or removes some.

Stage 1. Time only		Stage 2. Time + Text	
Precision	Recall	Precision	Recall
0.76	0.84	0.77	0.81

Table 2. When considering time information only, recall tends to be higher. When considering both time and text descriptions, incorrect text labels lower both precision and recall, but removing unnecessary steps in Stage 2 recovers precision.

In 76% of the steps, two or more Turkers agreed on a single description. For the rest, the tie breaking process determined the final description. For 13% of the steps, Turkers provided their own description.

Evaluating Stage 3: EXPAND

This evaluation should judge if crowd-selected before and after images correctly capture the effect of a step. Because this judgment is subjective, and there can be multiple correct before and after images for a step, we recruited six external human evaluators to visually verify the images. We assigned two evaluators to each domain based on their expertise and familiarity with the domain, and gave a one-hour training session on how to verify before and after images. For each workflow-generated step, we presented an extracted text description along with a before and after image pair. Their task was to make binary decisions (yes / no) on whether each image correctly represents the before or after state of the step.

We used Cohen’s Kappa to measure inter-rater agreement. The values were 0.57, 0.46, 0.38, in cooking, makeup, and Photoshop, respectively, which show a moderate level of agreement [1]. Results show that on average, both raters marked 60% of before and after images as correct. At least one rater marked 81.3% as correct.

Cost, time, and tasks

We created a total of 8,355 HITs on Mechanical Turk for annotating 75 videos. With three workers on each task and a reward of $0.07, $0.03, and $0.05 for Find, Verify, and Expand, respectively, the average cost of a single video was $4.85, or $1.07 for each minute of a how-to video. The Expand stage was more costly ($2.35) than the first two; thus, time points and text descriptions can be acquired at $2.50 per video. The average task submission time was 183, 80, and 113 seconds for Find, Verify, and Expand, respectively.

Results summary

In summary, our workflow successfully extracted step information from 75 existing videos on the web, generalizing to three distinct domains. The extracted steps on average showed 77% precision and 81% recall against ground truth, and were 2.7 seconds away from ground truth. Human evaluators found 60% of before and after images to be accurate.

DISCUSSION AND LIMITATIONS

We now discuss qualitative findings from the experiment, which might have practical implications for future researchers designing crowd workflows.

Detecting precise timing of a step. We observed that Turkers add new steps with higher latency than trained annotators, resulting in Turker-labeled time points being slightly later than those by annotators for the same step. The trained annotators often rewinded a few seconds to mark the exact timing of a step after seeing the step, whereas most Turkers completed their tasks in a single pass. While this might be a limitation of the workflow, our results show that a reasonable window size mitigates such differences. We will explore time-shifting techniques to see if timing accuracy improves.

Handling domain and video differences. Extraction accuracy in our workflow was consistent across the three domains with different task properties. This finding validates our domain-agnostic approach based on the general properties of procedural tasks. Photoshop videos were often screencasts, whereas cooking and makeup videos were physical demonstrations. Cooking videos contained higher number and density of steps than makeup or Photoshop videos, while Photoshop and makeup videos often had longer steps that required fine-grained adjustments and tweaking. Also, some videos were studio-produced with multiple cameras and high-quality post-processing, while others were made at home with a webcam. Our workflow performed robustly despite the various differences in task properties and video presentation styles.

Extracting steps at different conceptual levels. Video instructors present steps at different conceptual levels, and this makes it difficult to keep consistent the level of detail in Turkers' step detection. In a makeup video, an instructor said “Now apply the bronzer to your face evenly”, and shortly after applied the bronzer to her forehead, cheekbones, and jawline. While trained annotators captured this process as one step, our workflow produced four, including both the high-level instruction and the three detailed steps. Turkers generally captured steps at any level, but our current approach only constructs a linear list of steps, which sometimes led to redundancy. Previous research suggests that many procedural tasks contain a hierarchical solution structure [7], and we plan to extend this work to hierarchical annotation.

POSSIBLE APPLICATIONS AND GENERALIZATION

We list possible applications that leverage step information extracted from our workflow, and discuss ways to generalize the Find-Verify-Expand pattern beyond how-to videos.

Our scalable annotation workflow can enable a series of novel applications in addition to the ToolScape player. First, better video search can be made possible with finer-grained video indices and labels. For example, ingredient search for cooking or tool name search for Photoshop can show all videos and time points that cover a specific tutorial element. Furthermore, video players can present alternative examples to a current step. If a learner is watching how to apply the eyeliner, the interface can show just the snippets from other videos that include demonstrations of the eyeliner. This allows the learner to hop between different use cases and context for the step of interest, which can potentially improve learning outcomes.

We believe the Find-Verify-Expand pattern can generalize to annotating broader types of metadata beyond steps from how-to videos. For example, from a soccer video this pattern can extract goal moments with Find and Verify, and then use Expand to include a crucial pass that led to the goal, or a ceremony afterward. Generally, the pattern can extract metadata that is human-detectable but hard to completely automate. It is a scalable method for extracting time-sensitive metadata and annotating streaming data, which can be applied to video, audio, and time-series data.

CONCLUSION AND FUTURE WORK

This paper presents a scalable crowdsourcing workflow for annotating how-to videos. The Find-Verify-Expand pattern efficiently decomposes the complex annotation activity into micro-tasks. Step information extracted from the workflow can enable new ways to watch and learn from how-to videos. We also present ToolScape, an annotation-enabled video player supporting step-by-step interactivity, which is a potential client of this workflow. Our lab study shows the value of accessing and interacting with step-by-step information for how-to videos. Participants watching videos with ToolScape gained higher self-efficacy, rated their own work higher, and produced higher-rated designs.

Our future work will explore applying the workflow to additional procedural task domains such as origami, home DIY tasks, and Rubik's cube. We will also explore procedural tasks that require a conceptual understanding of the underlying concept, such as solving algorithm or physics problems.

Another direction for research is collecting task information using learners as crowd. We believe learners can potentially provide more advanced, higher-level, and richer information not possible with Turkers, if their learning interactions can naturally provide useful input to the system. Combining crowdsourcing with “learnersourcing” can extract rich annotations from existing resources while enhancing learning.

ACKNOWLEDGMENTS

The authors would like to thank the members of the UID Group at MIT CSAIL for their feedback. This work was funded in part by Quanta Computer. Juho Kim is supported by the Samsung Fellowship.

REFERENCES

1. Altman, D. G. *Practical statistics for medical research*, vol. 12. CRC Press, 1991.

2. Bandura, A. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review 84*, 2 (1977), 191.

3. Banovic, N., Grossman, T., Matejka, J., and Fitzmaurice, G. Waken: Reverse engineering usage information and interface structure from software videos. In *UIST '12* (2012).

4. Bernstein, M. S., Brandt, J., Miller, R. C., and Karger, D. R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *UIST '11*, ACM (2011).

5. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *UIST '10* (2010), 313–322.

6. Bernstein, M. S., Teevan, J., Dumais, S., Liebling, D., and Horvitz, E. Direct answers for search queries in the long tail. In *CHI '12'* (2012), 237–246.

7. Catrambone, R. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General 127*, 4 (1998), 355.

8. Chi, P.-Y., Ahn, S., Ren, A., Dontcheva, M., Li, W., and Hartmann, B. Mixt: Automatic generation of step-by-step mixed media tutorial. In *UIST '12* (2012).

9. Chi, P.-Y. P., Liu, J., Linder, J., Dontcheva, M., Li, W., and Hartmann, B. Democut: generating concise instructional videos for physical demonstrations. In *UIST '13*, ACM (2013).

10. Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., and Klemmer, S. R. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput.-Hum. Interact. 17*, 4 (Dec. 2010), 18:1–18:24.

11. Eiriksdottir, E., and Catrambone, R. Procedural instructions, principles, and examples: how to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Hum Factors 53*, 6 (2011), 749–70.

12. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, vol. 96 (1996).

13. Ferguson, E. L., and Hegarty, M. Learning with real machines or diagrams: application of knowledge to real-world problems. *Cognition and Instruction 13*, 1 (1995), 129–160.

14. Grabler, F., Agrawala, M., Li, W., Dontcheva, M., and Igarashi, T. Generating photo manipulation tutorials by demonstration. In *SIGGRAPH '09* (2009), 1–9.

15. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories. In *UIST '10* (2010).

16. Hadidi, R., and Sung, C.-H. Students' acceptance of web-based course offerings: an empirical assessment. *AMCIS 1998* (1998).

17. Jaro, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association 84*, 406 (1989), 414–420.

18. Kim, J. Toolscape: enhancing the learning experience of how-to videos. In *CHI EA '13* (2013), 2707–2712.

19. Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.

20. Lafreniere, B., Grossman, T., and Fitzmaurice, G. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *CHI '13* (2013), 1779–1788.

21. Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. Real-time crowd labeling for deployable activity recognition. In *CSCW '13* (2013), 1203–1212.

22. Li, F. C., Gupta, A., Sanocki, E., He, L.-w., and Rui, Y. Browsing digital video. In *CHI '00*, ACM (2000).

23. Nguyen, P., Kim, J., and Miller, R. C. Generating annotations for how-to videos using crowdsourcing. In *CHI EA '13*, ACM (2013), 835–840.

24. Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. Platemate: crowdsourcing nutritional analysis from food photographs. In *UIST '11* (2011), 1–12.

25. Park, S., Mohammadi, G., Artstein, R., and Morency, L.-P. Crowdsourcing micro-level multimedia annotations: The challenges of evaluation and interface. In *CrowdMM Workshop* (2012).

26. Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. Pause-and-play: Automatically linking screencast video tutorials with applications. In *UIST 2011* (2011).

27. Schunk, D. Goal setting and self-efficacy during self-regulated learning. *Educational psychologist 25*, 1 (1990), 71–86.

28. Tversky, B., Morrison, J. B., and Betrancourt, M. Animation: can it facilitate? *International journal of human-computer studies 57*, 4 (2002), 247–262.

29. van der Meij, H., Blijleven, P., and Jansen, L. What makes up a procedure? *Content & Complexity* (2003).

30. Vondrick, C., Ramanan, D., and Patterson, D. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV 2010*. Springer, 2010, 610–623.

31. Yuen, J., Russell, B., Liu, C., and Torralba, A. Labelme video: Building a video database with human annotations. In *ICCV 2009*, IEEE (2009), 1451–1458.

32. Zhang, D., Zhou, L., Briggs, R. O., and Jr., J. F. N. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management 43*, 1 (2006), 15 – 27.

33. Zimmerman, B. J., Bandura, A., and Martinez-Pons, M. Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal 29*, 3 (1992), 663–676.

Data-Driven Interaction Techniques for Improving Navigation of Educational Videos

Juho Kim[1] **Philip J. Guo**[1,2] **Carrie J. Cai**[1] **Shang-Wen (Daniel) Li**[1]
Krzysztof Z. Gajos[3] **Robert C. Miller**[1]

[1]MIT CSAIL
Cambridge, MA USA
{juhokim, cjcai, swli, rcm}@mit.edu

[2]University of Rochester
Rochester, NY USA
pg@cs.rochester.edu

[3]Harvard SEAS
Cambridge, MA USA
kgajos@eecs.harvard.edu

ABSTRACT

With an unprecedented scale of learners watching educational videos on online platforms such as MOOCs and YouTube, there is an opportunity to incorporate data generated from their interactions into the design of novel video interaction techniques. Interaction data has the potential to help not only instructors to improve their videos, but also to enrich the learning experience of educational video watchers. This paper explores the design space of data-driven interaction techniques for educational video navigation. We introduce a set of techniques that augment existing video interface widgets, including: a 2D video timeline with an embedded visualization of collective navigation traces; dynamic and non-linear timeline scrubbing; data-enhanced transcript search and keyword summary; automatic display of relevant still frames next to the video; and a visual summary representing points with high learner activity. To evaluate the feasibility of the techniques, we ran a laboratory user study with simulated learning tasks. Participants rated watching lecture videos with interaction data to be efficient and useful in completing the tasks. However, no significant differences were found in task performance, suggesting that interaction data may not always align with moment-by-moment information needs during the tasks.

Author Keywords

Video learning; Interaction peaks; Video summarization; MOOCs; Multimedia learning; Video content analysis.

ACM Classification Keywords

H.5.1. Multimedia Information Systems: Video

INTRODUCTION

Millions of people watch free educational videos online on platforms such as Khan Academy, Coursera, edX, Udacity, MIT OpenCourseWare, and YouTube. For example, the "Education" channel on YouTube currently has over 10.5 million subscribers, and a typical MOOC has thousands of video-watching learners. In addition, learners also take paid video-centric courses on commercial platforms such as Lynda, Udemy, and numerous university e-learning initiatives.

The server logs of these platforms contain fine-grained, second-by-second data of learners' interactions with videos, which we refer to as *interaction traces*. This data is now being used for real-time analytics to optimize business metrics such as viewer engagement time. Researchers have also used this data to perform retrospective empirical analyses. For example, video analytics studies on MOOCs have compared the effects of video production methods on learner engagement [13] and identified common causes of peaks in learner activity within videos [20].

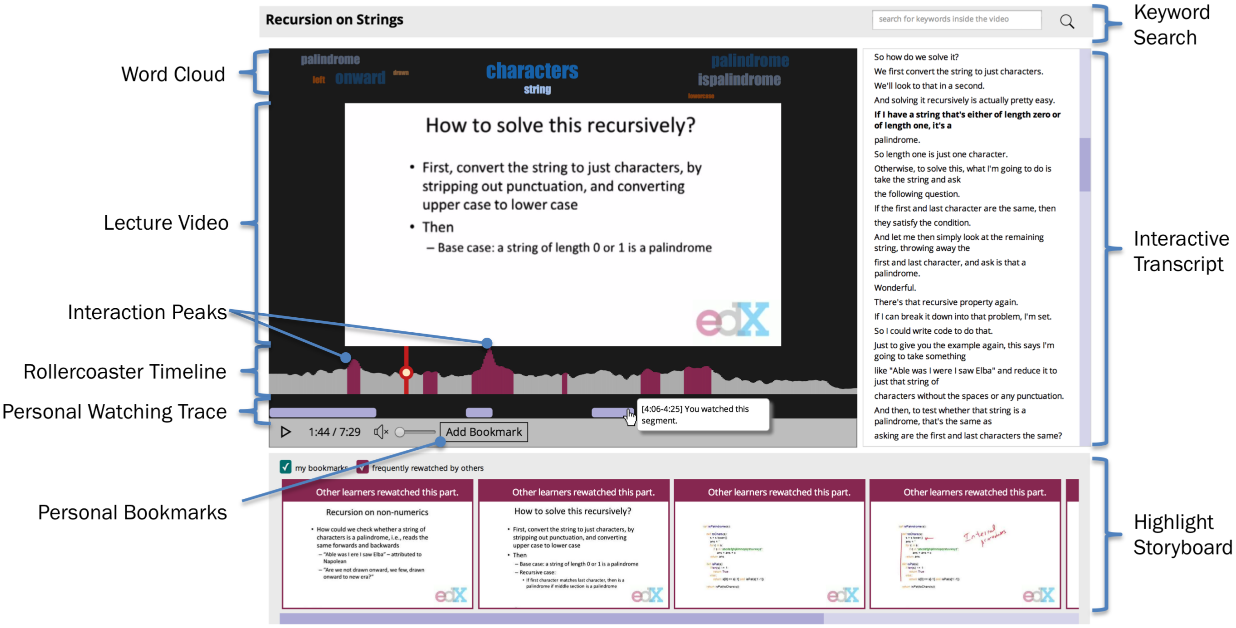
Interaction data provides a unique opportunity to understand collective video watching patterns, which might indicate points of learner interest, confusion, or boredom in videos. However, to our knowledge, researchers have not yet attempted to feed these patterns back into the video navigation interface to support learners. While learners might have diverse goals in navigating through a video, existing video interfaces do not provide customized navigation support beyond scrubbing on a linear timeline slider with thumbnail previews and synchronizing with a textual transcript. Adapting to collective video watching patterns can lead to richer social navigation support [10].

This paper explores the design space of navigation techniques for educational videos that leverage interaction data. We introduce novel data-driven interaction techniques that process, visualize, and summarize interaction data generated by many learners watching the same video. For instance, in a typical MOOC, at least a few thousand learners watch each video. Based on prior findings about learner intent and typical formats of educational videos [13, 20], we have designed these techniques to support fluid and diverse video navigation patterns. Typical video watching scenarios include:

- Rewatch: "Although I understand the high-level motivation, I didn't quite get the formal definition of 'admissible heuristic' the first time I watched this lecture. So I want to rewatch the section explaining the formal definition."
- Textual search: "I want to jump to where the instructor first mentioned the phrase 'alpha-beta pruning.'"

UIST'14, October 5–8, 2014, Honolulu, HI, USA.

ACM 978-1-4503-3069-5/14/10 $15.00.
http://dx.doi.org/10.1145/2642918.2647389



Recursion on Strings

How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
  – Base case: a string of length 0 or 1 is a palindrome

[4:06-4:25] You watched this segment.

So how do we solve it?
We first convert the string to just characters.
We'll look to that in a second.
And solving it recursively is actually pretty easy.
If I have a string that's either of length zero or of length one, it's a
palindrome.
So length one is just one character.
Otherwise, to solve this, what I'm going to do is take the string and ask
the following question.
If the first and last character are the same, then
they satisfy the condition.
And let me then simply look at the remaining string, throwing away the
first and last character, and ask is that a palindrome.
Wonderful.
There's that recursive property again.
If I can break it down into that problem, I'm set.
So I could write code to do that.
Just to give you the example again, this says I'm going to take something
like "Able was I were I saw Elba" and reduce it to just that string of
characters without the spaces or any punctuation.
And then, to test whether that string is a palindrome, that's the same as
asking are the first and last characters the same?

☑ my bookmarks ☑ frequently rewatched by others

Other learners rewatched this part.

Recursion on non-numerics

- How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards
  – "Able was I ere I saw Elba" – attributed to Napolean
  – "Are we not drawn onward, we few, drawn onward to new era?"

Other learners rewatched this part.

How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
  – Base case: a string of length 0 or 1 is a palindrome
  – Recursive case:
    - If first character matches last character, then is a palindrome if middle section is a palindrome

Other learners rewatched this part.

Other learners rewatched this part.

Internal procedures

Figure 1. This paper presents three sets of novel interaction techniques to improve navigation of educational videos. 1) Dynamic timelines (Rollercoaster Timeline, Interaction Peaks, and Personal Watching Trace), 2) Enhanced in-video search (Keyword Search and Interactive Transcript), 3) Highlights (Word Cloud, Personal Bookmarks, Highlight Storyboard). All techniques are powered by interaction data aggregated over all video watchers.

- Visual search: "I remember seeing this code example in a diagram somewhere in the video. I want to find it again."
- Return: "Hey, that was annoying! I don't want to see the instructor's talking head. I'm not done looking at this PowerPoint slide yet. I want the slide back!"
- Skim: "This lecture seems somewhat trivial. I'll skim to see if there's something I probably shouldn't miss."

Specifically, we developed interaction techniques to augment a traditional Web video player with 1) a Rollercoaster timeline that expands the video timeline to a 2D space, visualizes collective interaction traces of all learners, and dynamically applies non-linear scrolling to emphasize interaction peaks, 2) enhanced in-video search that visualizes and ranks occurrences on the timeline and recommends salient keywords for each video section, and 3) a video summarization method that captures frames that are frequently viewed by other learners. These techniques combine learners' collective interaction traces with text and visual content analysis.

We package all of these techniques together in LectureScape, a prototype Web-based video interface shown in Figure 1. In a laboratory study with simulated search and skimming tasks, we observed that participants employ diverse video navigation patterns enabled by our techniques. Specifically, they noted that LectureScape helped them to quickly scan the video and efficiently narrow down to parts to direct their focus. They also found interaction data to be useful in identifying important or confusing pedagogical points within videos. However, no significant differences were found in task performance, suggesting that interaction data may not always align with moment-by-moment information needs participants had for the study tasks.

This paper makes the following contributions:

- a conceptual design approach for interaction techniques that leverages information about other learners' behavior to improve the video learning experience,
- a set of novel video interaction techniques powered by real log data from learners in a MOOC platform, introducing 1) a 2D, non-linear timeline, 2) enhanced in-video search, and 3) a data-driven video summarization method,
- and an empirical evaluation of the techniques with learners, which enabled fluid and diverse video navigation.

RELATED WORK

We review previous research in leveraging interaction history to improve user interfaces and video navigation.

Leveraging Interaction History

There is a rich thread of research in using interaction history data to analyze usage patterns and improve users' task performance. Interaction history data is automatically collected by applications during normal usage. Examples include Web browsers logging Web page visit history, search engines capturing query history, and video players storing video interaction clickstreams such as play and pause events. Read Wear [14] presented a visionary idea in this space to visualize users' read and edit history data in the scrollbar. Chronicle [12] captured and provided playback for rich, contextual user interaction history inside a graphical application. Dirty Desktops [16] applied magnetic forces to each interaction trace, which improved target selection for commonly used widgets. Patina [26] separated individual and collective history and added overlays on top of the GUI, to help people find commonly used menu items and discover new ways of completing desktop-related tasks. Causality [30] introduced an application-independent conceptual model for working with interaction history. This paper uses video interaction history to support common navigation tasks in video-based learning.

To model user interest in video watching, researchers have proposed features such as viewership [38], scrubbing [39], zooming and panning [7], and replaying and skipping [9] activities. SocialSkip [9] applied signal processing to replaying activity data in order to infer interesting video segments. Other researchers have used more explicit input from video watchers, including user ratings [33], annotations [38], and the "this part is important" button [36]. Most existing approaches introduce a modeling technique or data visualization. We take this data further to build new interaction techniques for video navigation, which prior work has not done. Also, we combine both implicit user history data and explicit user bookmarks to support diverse learning tasks, which extends prior work on supporting social navigation for lecture videos [28].

Video Navigation Techniques

To improve video navigation with interaction data, we designed novel techniques to 1) add richer interactions to the video timeline, 2) support enhanced in-video search, and 3) automatically summarize video content. We now review related work for each of the three techniques.

To improve video scrubbing, YouTube displays thumbnail previews for quickly skimming local frames, and Swift [25] overlays low-resolution thumbnails to avoid network latency delays. The content-aware timeline [34] extracts keyframes with content analysis and plays a video snippet around these points when the user scrubs the timeline. Elastic interfaces [24] use the rubber band analogy to control scrubbing speed and support precise navigation, as seen in the PV Slider [35] and Apple's iOS video interface. We extend this line of research by asking: "What if the scrubbing behavior adapts to learners' watching patterns, as collected from interaction history data?" To our knowledge, no video scrubbing technique has leveraged interaction history data.

Another thread of research introduced techniques to support navigation of how-to videos, a sub-genre of educational video that includes procedural, step-by-step instructions about completing a specific task. Existing systems reveal step-by-step structure by adding rich signals to the video timeline, such as tool usage and intermediate results in graphical applications [8, 21]. Classroom lecture videos tend to be less structured than how-to videos, which makes capturing clear structural signals harder. We instead turn to interaction data that is automatically logged for learners as they watch the video.

Popular GUI applications such as Web browsers and text editors have incremental search features where the scrollbar and text visually highlight locations of search term occurrences. Also, video players on educational platforms such as edX show a synchronized transcript alongside the currently playing video. Learners can search for text in the transcript and then click to jump to the corresponding spot in the video. We improve these interfaces by augmenting search results with interaction data and visualizing them on the video timeline.

Existing video summarization techniques use video content analysis to extract keyframes [3], shot boundaries [22], and visual saliency [15]. To provide an overview of the entire clip at a glance and support rapid navigation, recent research has used a grid layout to display pre-cached thumbnails [27], short snippets [18] in a single clip, personal watching history for multiple clips [1], a conceptual hierarchy visualization [19], or a 3D space-time cube display [31]. For educational lecture videos, Panopticon [18] has been shown to shorten task completion time in seeking information inside videos [32]. For blackboard-style lecture videos, NoteVideo [29] reverse-engineers a rendered video to create a summary image and support spatial and temporal navigation. This paper introduces a new summarization technique that uses *interaction peaks*, points in a video with significantly high play button click activity, to generate highlight frames of a clip.

DESIGN GOALS

This work focuses on supporting video navigation patterns common in online education, which differ from watching, say, a movie or TV show in a sequential, linear manner. Our designs are informed by quantitative and qualitative findings from analyses of educational videos, which suggest that learners often re-watch and find specific information from videos. Prior work in video clickstream analysis on four edX MOOCs found many *interaction peaks*, i.e., concentrated bursts in play/pause button clicks during certain segments of a video [20]. 70% of automatically-detected peaks coincided with visual transitions (e.g., switching between an instructor's head and a slide) and topic transitions [20]. A challenge in using interaction data to support learning is that the meaning of an interaction peak can be ambiguous (e.g., interest, confusion, or importance). In this paper, we do not assume a specific meaning behind interaction peaks, but do assume they are worth emphasizing regardless of the real cause. If a peak indicates importance, it would make sense to highlight it for future learners. If it indicates confusion, it may still make sense to emphasize so that learners would pay more attention.

To discover unsupported needs in lecture video navigation, we also conducted multiple rounds of feedback sessions with learners using our initial prototypes. The data analysis and interviews led to three high-level goals that informed our design of data-driven video interaction techniques.

Provide easy access to what other learners frequently watched. Our observations suggest that learners find it hard to identify and navigate to important parts of information-dense educational videos. To help a learner make more informed decisions about which part of the video to review, we leverage other learners' interaction traces, especially interaction peaks. We designed navigation techniques to emphasize these points of interest while the learner visually scans the video or physically scrubs the timeline.

Support both personal and collective video summaries. To prepare for homework assignments or exams, learners often take notes and watch videos multiple times to create a meaningful summary. Since there are often thousands of learners watching each video, we explore ways to present collective interaction traces as an alternative summary to complement each learner's personal summary. We extend prior work on social navigation in videos [28], history visualization, and revisitation mechanisms by supporting both manual bookmarking and automatic personal and collective watching traces.

Support diverse ways to search inside of a video. In our formative studies, learners described different ways they look for specific information inside a video. They would rely on both textual cues (e.g., topic and concept names) and visual cues (e.g., an image or a slide layout) to remember parts of the video. A more challenging case is when they cannot remember what the cue was for their particular information need. This observation inspired us to support both active search (e.g., when the learner has a clear search term), and ambient recommendations (e.g., when the learner does not know exactly what to search for). We designed techniques to enhance existing search mechanisms with interaction data, which provide social cues to serve both search scenarios.

DATA-DRIVEN VIDEO NAVIGATION TECHNIQUES

We introduce three interaction techniques to improve navigation of educational videos: an alternative timeline, search interface, and summarization method. Our main insight is to use the non-uniform distribution of learner activity within a video to better support common navigation patterns. Although the prototypes shown in this paper use videos on edX, a MOOC (Massive Open Online Course) platform, the techniques can be implemented for other video platforms such as YouTube because they use only standard Web technologies.

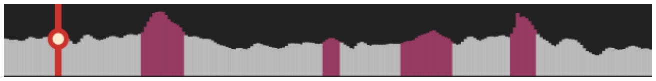


Figure 2. The 2D Rollercoaster timeline that appears below each video instead of a traditional 1D timeline. The height of the timeline at each point shows the amount of navigation activity by learners at that point. The magenta sections are automatically-detected interaction peaks.

The Rollercoaster Timeline: 2D, Non-Linear Timeline

To help learners identify and navigate to important parts of the video, we introduce the rollercoaster timeline. Unlike a traditional 1D timeline, the rollercoaster timeline is 2D with an embedded visualization of second-by-second learner interaction data (Figure 2). It visualizes the navigation frequency as a proxy of importance, as revealed by the behavior of other learners, and modifies the timeline scrubbing behavior to make precise navigation in important regions easier.

Navigation events are logged when the learner pauses and resumes the video, or navigates to a specific point. The Rollercoaster timeline uses navigation event counts as the vertical dimension. This visualization can also show other kinds of interaction events, including the number of viewers, rewatchers, unique viewers, or play or pause button clicks.

2D timeline

If the learner wants to jump to a specific point in the video, he can click on any point in the 2D timeline, which will capture the x coordinate of the click and update the playhead. The embedded peak visualization shows the intensity and range of each peak, and the overall distribution of the peaks within a video. Since interaction peaks are highlighted in magenta and span a wider region than other points, the learner can visually review and navigate to the commonly revisited parts in the video. We use the Twitinfo [23] peak detection algorithm to detect peaks in the server log data.

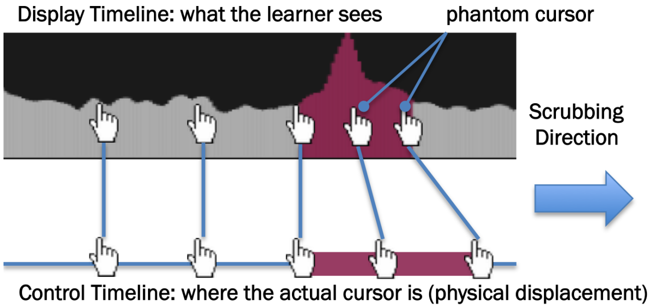

Display Timeline: what the learner sees
phantom cursor
Scrubbing Direction
Control Timeline: where the actual cursor is (physical displacement)

Figure 3. Non-linear scrubbing in the Rollercoaster timeline. To draw the learner's attention to content around interaction peaks, the phantom cursor decelerates scrubbing speed when the cursor enters a peak range.

Non-linear scrubbing with the phantom cursor

This timeline also enables dynamic, non-linear scrubbing, which takes advantage of interaction peaks. The basic idea is to apply friction while scrubbing around peaks, which leads to prolonged exposure so that learners can get a more comprehensive view of the frames near the peaks even when scrubbing quickly. Friction also makes it easier to precisely select specific frames within the range, since it lowers the frame update rate. It is an example of control-display ratio adaptation [5, 16, 37], dynamically changing the ratio between physical cursor movement and on-screen cursor movement.

Previous techniques have applied elastic, rubber band-like interactions to scrubbing [17, 24, 34, 35]. Our technique differs in that 1) it uses interaction data instead of content-driven keyframes, 2) elasticity is selectively applied to parts of the timeline, and 3) the playhead and the cursor are always synchronized, which reduced user confusion in our pilot studies.

When the mouse cursor enters a peak region while dragging, the dragging speed slows down relative to the dragging force, creating the sense of friction. The faster the dragging, the weaker the friction. We achieve this effect by temporarily hiding the real cursor, and replacing it with a *phantom cursor* that moves slower than the real cursor within peak ranges (Figure 3). The idea of enlarging the motor space around targets is inspired by Snap-and-Go [4].

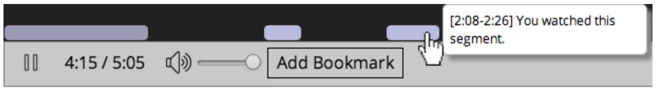

[2:08-2:26] You watched this segment.

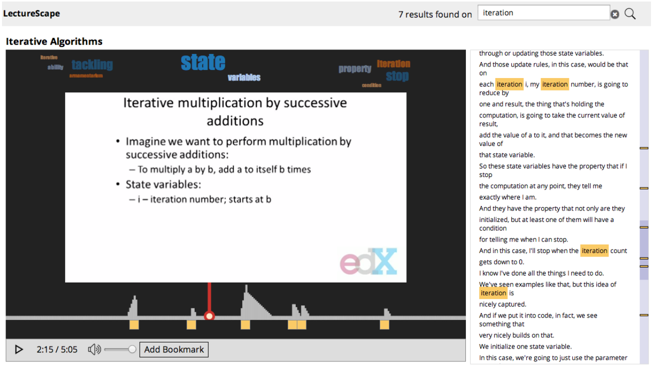
Figure 4. The real-time personal watching traces visualize segments of the video that the learner has watched.

Personal watching trace visualization

When we observed pilot study users navigating videos with our timeline, a common desire was to keep track of which parts of the video they personally watched, which might not align with the aggregate interaction peaks collected over all learners. Thus, we added another stream under the timeline to visualize each learner's personal watching traces. Previous research has separated personal and collective history traces to support GUI command selection [26], and added history indicators to a document scrollbar [2], which improved task performance in information finding. We extend these approaches to video navigation by using personal watching traces to support revisitation. Once the learner pauses the video or jumps to a new point, the current watching segment is visualized on a separate track below the timeline (Figure 4). Clicking on a generated segment replays the segment. More recent segments are displayed with higher opacity to further emphasize them over older ones. These traces can be stored on a per-user basis to help learners quickly find points of interest when they return to re-watch a video at a later date.

Keyword Search and Visualization

To better support searching for relevant information inside of a video, we use interaction data to power keyword search and transcript analysis. Instead of weighing all occurrences equally, our search technique rewards results in sections of the video where more learners watched. Since key concepts often appear dozens of times in a video, this feature helps the learner prioritize which parts of the video to review. Furthermore, to support novice learners who do not necessarily have the vocabulary to translate their information needs into a direct search query, we suggest major topics discussed in each section of the video in a word cloud. These topics serve as a keyword summary that can help learners recognize and remember the main topics discussed in each video.



Iterative multiplication by successive additions

- Imagine we want to perform multiplication by successive additions:
  - To multiply a by b, add a to itself b times
- State variables:
  - i – iteration number; starts at b

through or updating those state variables.
And those update rules, in this case, would be that on
each iteration i, my iteration number, is going to reduce by
one and result, the thing that's holding the
computation, is going to take the current value of result,
add the value of a to it, and that becomes the new value of
that state variable.
So these state variables have the property that if I stop
the computation at any point, they tell me
exactly where I am.
And they have the property that not only are they
initialized, but at least one of them will have a condition
for telling me when I can stop.
And in this case, I'll stop when the iteration count
gets down to 0.
I know I've done all the things I need to do.
We've seen examples like that, but this idea of iteration is
nicely captured.
And if we put it into code, in fact, we see something that
very nicely builds on that.
We initialize one state variable.
In this case, we're going to just use the parameter

Figure 5. Our interaction data-driven keyword search brings in more context for the learner to decide where to focus on when searching for a keyword. For instance, the learner can visually check the distribution of when the lecturer said the keyword in the current video, which is useful in seeing where it was heavily discussed versus simply introduced.

Keyword search

If the learner has a keyword to search for, she can type it in the search field (top right in Figure 5), which searches over the full transcript of a video clip, and displays results both on the timeline and in the transcript. When the learner enters a search query, the timeline dynamically displays the search results instead of the default interaction visualization (see Figure 6). In this way, the timeline serves as a dynamic space for supporting different learner tasks by changing the peak points it emphasizes. Each result renders as a pyramid-shaped distribution, whose range is the duration of the sentence the word belongs to and whose peak is where the term is spoken. Figure 7 shows how hovering over the result displays a tooltip, and clicking on the result plays the video from the beginning of the sentence that includes the search term. This sentence-level playback provides the learner with more context surrounding the term.

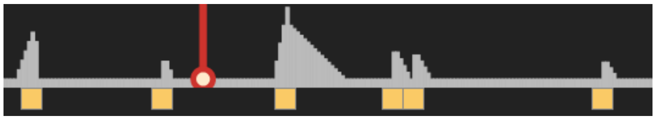
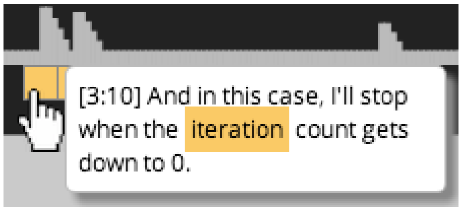


Figure 6. The search timeline appears below the video after each search. It visualizes the positions of search results, as well as the relative importance of each. Here the high peak in the middle indicates both that it contains the search term and that lots of learners watched that part.



[3:10] And in this case, I'll stop when the iteration count gets down to 0.

Figure 7. Hovering on a search result displays a tooltip with the transcript sentence that contains the search term. Clicking on the result plays the video starting at the beginning of the sentence to assist the learner with comprehending the surrounding context.

Because key terms are repeated many times even during a short video, it can be hard for the learner to locate the most important search result. For example, in a 5-minute edX video on iterative algorithms, "variable" is mentioned 13 times, and "state" 12 times. We use interaction data to rank search results, with the assumption that parts of the video more frequently watched by previous learners are likely to reflect the current learner's interest. Our ranking algorithm analyzes learner activity around a search result and assigns a weight to the result, giving higher weights to sentences that were viewed by more learners. It then computes the relevance score by combining this weight with term frequency within the sentence. To support quick visual inspection of search results, we represent the computed score as the height on the timeline (gray peaks in Figure 6). If the term was mentioned multiple times in a sentence, we convolve the distributions for all occurrences and assign the maximum score to it.

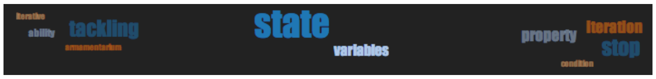


Figure 8. The word cloud displays automatically-extracted topics for the currently visible section of the video, providing a keyword-based summarization of the video. Clicking on any word triggers a search for it.

Word cloud: topic summarization and visualization

To address the low visual variation between frames in many videos and to help learners recognize and remember major topics in the clip, we use word clouds to dynamically display keywords in different segments of the video. We use TF-IDF (term frequency-inverse document frequency) scores for extracting keywords and weighing their importance. To compute the TF-IDF scores for the keywords in a transcript, we define a document as the transcription sentences between two consecutive interaction peaks, and the background corpus as the collection of all video transcripts in the entire course. This user activity-driven mechanism extracts self-contained segments from each video.

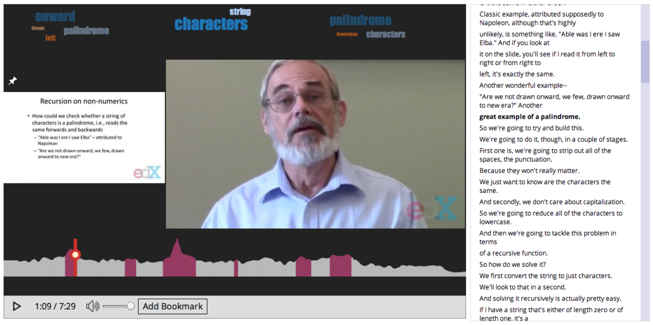
The visualization is positioned directly above the video as a panel consisting of three word clouds (see Figure 8), and gets updated at every interaction peak. The center cloud corresponds to the present segment being viewed, and two additional clouds represent the previous and upcoming segments, respectively. These displays are intended to give a sense of how lecture content in the current segment compares to that of surrounding segments. To bring focus to the current word cloud, we de-emphasize the previous and next word clouds by decreasing their opacity and word size relative to the current cloud. Clicking on any keyword in the cloud triggers a search using that term, visualizing the occurrences in the transcript as well as on the timeline.

Video Summarization with Highlights

To enable a quick overview of important points, we present a strip of visual highlights of selected video frames. Consistent with our design goal of providing access to both personal and collective interaction traces in the timeline, we support both collective and personal highlights. Collective highlights are captured by interaction peaks, while personal highlights are captured by the learner bookmarking a frame of interest.

Interaction peak highlights

We capture interaction peaks and provide one-click access to them to support common watching patterns such as jumping directly to these points. The storyboard-style display of the peak frames allows the learner to visually scan the video's progress (Figure 1). These highlights are visually-oriented, while the word cloud of Figure 8 is text-oriented.



Recursion on non-numerics

- How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards
  - "Able was I ere I saw Elba" – attributed to Napolean
  - "Are we not drawn onward, we few, drawn onward to new era?"

Classic example, attributed supposedly to Napoleon, although that's highly

unlikely, is something like, "Able was I ere I saw Elba." And if you look at

it on the slide, you'll see if I read it from left to right or from right to

left, it's exactly the same.

Another wonderful example--

"Are we not drawn onward, we few, drawn onward to new era?" Another

great example of a palindrome.

So we're going to try and build this.

We're going to do it, though, in a couple of stages.

First one is, we're going to strip out all of the spaces, the punctuation.

Because they won't really matter.

We just want to know are the characters the same.

And secondly, we don't care about capitalization.

So we're going to reduce all of the characters to lowercase.

And then we're going to tackle this problem in terms

of a recursive function.

So how do we solve it?

We first convert the string to just characters.

We'll look to that in a second.

And solving it recursively is actually pretty easy.

If I have a string that's either of length zero or of length one, it's a

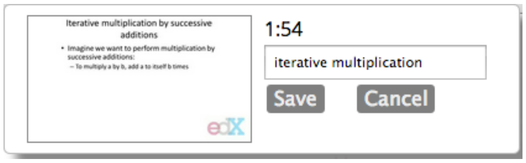
Figure 9. Our pinning algorithm analyzes interaction peaks and visual transitions in the video to display a smaller static frame (on the left) next to the video (on the right). Learners can manually pin any frame as well.

Pinning video frames

Most existing video players display only one frame at a time. This ubiquitous interface is sensible for the narrative structure of general-purpose videos such as TV shows, where a sequential flow is natural. However, educational videos are information-heavy, and active learning involves skimming and scrubbing [20]. For example, an instructor might verbally refer to the formula she described in the previous PowerPoint slide, but the formula might no longer be available on screen. A learner who wants to refer to that formula has to scrub the video timeline to go back to a frame with the relevant formula slide. To support watching patterns that are not easily supported by existing players, our video player pins a relevant frame next to the video stream for easy reference (Figure 9).

Our video player automatically determines a frame to pin. A relevant pinned frame should 1) not be identical to what is currently shown in the video, 2) include important content that is worth referencing, and 3) contain readable content such a textual slide or technical diagram, not merely a static frame of the instructor's face or students sitting in a classroom. Otherwise, juxtaposing a static frame next to the video might cause distraction and visual clutter. To meet these requirements, our pinning algorithm uses both interaction peaks and visual transitions. It automatically pins an interaction peak frame if there is a visual transition shortly after the peak. Checking for a visual transition ensures that the pinned frame is not visually identical to the frames right after the transition. Also, pinning an interaction peak frame ensures that the frame at least includes content viewed by many others.

The learner can also manually pin a frame by clicking the pin icon attached to each peak frame, which replaces the current pinned frame with the learner's. While the system attempts its best effort to show the most relevant frame at a given time, the learner also has the flexibility to control what gets displayed.



Iterative multiplication by successive additions

- Imagine we want to perform multiplication by successive additions:
  - To multiply a by b, add a to itself b times

1:54

iterative multiplication

Figure 10. The learner can add a labeled bookmark, which is added to the highlights stream below the video for visual preview and revisitation.

Personal bookmarks

While peak frames might be a reasonable summary of a video, individual learners might have summarization needs that are not captured by the collective traces. The learner can add personal bookmarks by clicking on the "Add Bookmark" button. The learner can see the captured frame at the point of click and add their own label for future reference (Figure 10). Once a bookmark is saved, it is added to the highlights stream, chronologically ordered along with other bookmarks and interaction peak highlights. This view allows the learner to choose between naturally-formed interaction peaks by other learners as well as self-generated bookmarks.

LECTURESCAPE: WEB-BASED PROTOTYPE

This section introduces LectureScape, a prototype lecture video player that combines all of the techniques in a unified interface (see Figure 1). The main goal of LectureScape is to give learners more control and flexibility in deciding how to navigate educational videos. LectureScape features the video player in the main view, along with the Rollercoaster timeline below the video and the word cloud above it. The interactive transcript is in the right sidebar, and the highlights are positioned at the bottom of the screen. The search box at the top enables keyword search. The widgets are all collapsible to reduce visual clutter and to hide unused features.

We implemented LectureScape using standard Web technologies: HTML5, CSS3, and JavaScript. The word cloud and Rollercoaster timeline are rendered with D3 [6]. Our customized HTML5 video player does not require additional re-encoding or streaming, and is independent of the encoding or streaming method used. It only requires interaction data for each video to activate the data-driven interaction techniques. Our data processing pipeline formats interaction traces and generates visual assets from an existing video.

First, three data streams are stored for each video: interaction data, TF-IDF results on the transcript, and visual transition data. Our peak detection algorithm runs on page load, which allows dynamic parameter tuning. Peak detection is run on both the interaction data and visual transition data, which returns interaction peaks and shot boundaries, respectively.

Second, the system generates thumbnail images for each second of a video. Because many of our interactions require displaying a thumbnail of a video frame on demand, low latency in image loading is crucial in supporting seamless interactions. It is especially important for educational videos whose visual changes are more subtle than in movies or TV shows, and whose on-screen information matters for learners to read and comprehend. Upon a page load, the system preloads all thumbnails for a video. When a learner drags the timeline, instead of loading the video each time a dragging event is triggered, our player pauses the video and displays an overlay screenshot. This results in much less latency and smoother dragging, similar to the benefits reported by Swift [25].

EVALUATION

To assess the feasibility of using interaction data to enhance video navigation, we conducted a user study comparing video players with and without our data-driven interaction techniques. We explored three research questions:

- **RQ1.** How do learners navigate lecture videos with LectureScape in typical kinds of learning tasks such as search and summarization?
- **RQ2.** How do learners interpret interaction data presented in LectureScape?
- **RQ3.** Are LectureScape's features useful and learnable?

Study Design

The study was a within-subjects design, where each learner used both LectureScape and a baseline interface that stripped off all interaction data-related features from LectureScape. The baseline interface still included the interactive transcript and preview thumbnails on hover, to emulate what is available in platforms such as edX or YouTube. To maintain uniformity in look and feel for our comparative study, the baseline interface had the same layout and visual design as LectureScape.

Learners performed three types of learning tasks for lecture videos: visual search, problem search, and summarization. These tasks represent realistic video watching scenarios from our observations, and match common evaluation tasks used in the literature on video navigation interfaces [11, 32].

- **Visual search** tasks involved finding a specific piece of visual content in a video. These tasks emulated situations when a learner remembers something visually and wants to find where it appeared in a video. For example, for a video about tuples in Python, a visual search task asked: *"Find a slide where the instructor displays on screen examples of the singleton operation."* Targets were slides that appeared briefly (less than 20 seconds) in the video. We mixed tasks that had targets around interaction peaks and non-peaks to investigate how LectureScape fares even when the target is not near a peak. For all visual search tasks, we provided learners with only the video timeline (linear timeline in baseline, 2D timeline in LectureScape) and removed all other features (e.g., transcript, word cloud) to restrict video navigation to timelines only. Learners were told to pause the video as soon as they navigated to the answer.
- **Problem search** tasks involved finding an answer to a given problem. These tasks emulated a learner rewatching a relevant video to answer a discussion forum question or to solve a homework problem. For example, for a video about approximation methods, a problem search task asked: *"If the step size in an approximation method decreases, does the code run faster or slower?"* Learners were asked to find the part in the video that discussed the answer, and then state their answer.
- **Summarization** tasks required learners to write down the main points of a video while skimming through it. We gave learners only three minutes to summarize videos that were seven to eight minutes long, with the intent of motivating learners to be selective about what parts to watch.

All videos used in the study were from an introductory computer science course on edX. Interaction data was collected from server logs during the first offering of the course in fall 2012. The course has been recurring every semester since then. Each of the eight tasks in the study used different videos to minimize learning effects. We also chose videos of similar length, difficulty, and style within each task type to control for differences across videos.

Participants

We recruited 12 participants (5 male, mean age 25.6, stdev=11.0, max=49, min=18) via a recruitment flyer on the course discussion forum on edX and the on-campus course website, both with consent from instructors. We recruited only learners who were currently enrolled in the introductory CS course (either on edX or on campus) to which the videos belong. The on-campus version of the course shares the same curriculum but is taught by different instructors. Furthermore, we picked videos for lessons given earlier in the semester, so that participants were likely to have already been exposed to that material before coming to our study, as is often the case in video re-watching scenarios. Four participants were enrolled in a current or previous edX offering of the course, while six were taking the on-campus version. Two had previously registered in the online offering but were currently taking the on-campus course. Participants received $30 for their time.

Procedure

A 75-minute study session started with 15-minute tutorials on both interfaces. Next, participants performed eight learning tasks: four visual search tasks, two problem search tasks, and two summarization tasks. After each task, they answered questions about confidence in their answer and prior exposure to the video. After each task type, we interviewed them about their task strategy. For each task type, we counterbalanced the order of the interfaces and the assignment of videos to tasks. After completing all the tasks, participants completed a questionnaire on the usability of each interface and their experience and opinions about interaction data. All click-level interactions were logged by the server for analysis.

Results

RQ1. Navigation Patterns for Search and Summarization

In **visual search**, most participants in the baseline 1D timeline sequentially scanned the video using thumbnail previews or dragging. In contrast, participants using the 2D Rollercoaster timeline often jumped between interaction peaks to reach the general area of the answer. But the latter strategy did not help in many cases because interaction data represents collective interests in a video, not results for a search query.

For the two out of four tasks where the search targets were located near interaction peaks, it took participants in both conditions similar amounts of time (LectureScape: μ=85 seconds, σ=51, baseline: μ=80, σ=73). This difference was not statistically significant with the Mann-Whitney U (MWU) test (p>0.4, Z=-0.9). For the other two tasks where the search targets were outside of an interaction peak range, it took participants in the LectureScape condition longer to complete (μ=117, σ=44) than in the baseline condition (μ=90, σ=50), although the MWU test showed no statistical significance (p>0.1, Z=-1.5) The longer time might be due to the fact that many participants navigated by clicking on the peaks to see if the answer existed around peaks. Nonetheless, results show that LectureScape did not adversely affect task completion times even when the answer was not in peak ranges.

Because **problem search** tasks required some understanding of the context to provide an answer, participants often tackled the problem by narrowing down to a section of the video that mentioned relevant concepts and then watching the section until they found the answer. In the process of narrowing down to a video section, most participants in the baseline condition relied on searching for keywords (10 out of 12) and clicking on transcript text (11 / 12), while participants with LectureScape used search (6 / 12) and clicking on transcript text (6 / 12) less frequently, and additionally clicked on interaction peaks on the timeline (6 / 12) or highlights below the video (6 / 12). Over all problem search tasks, participants in the LectureScape condition completed them slightly faster (μ=96, σ=58) than participants in the baseline condition (μ=106, σ=58), although the difference was not significant (MWU test, p=0.8, Z=0.3).

In comparison to the other tasks, participants in **summarization** tasks did not rely on keyword search (1 / 12 in both conditions), because the task required them to quickly scan the entire video for the main points. Many participants with LectureScape scanned the peak visualization, word cloud, and highlights for an overview, and clicked on interaction peaks (9 / 12) or highlights (6 / 12) for a detailed review. In one video, all six participants with LectureScape visited an interaction peak almost at the end of the video (located at 6:35 in the 7:00 clip). This slide summarized main ideas of variable binding, which was the topic of the video. In contrast, in the baseline condition, only one learner navigated to this section of the video. Most participants spent majority of their time in the earlier part of the video in the baseline condition.

Despite inconclusive evidence on quantitative differences in task completion time, participants believed that they were able to complete the tasks faster and more efficiently with LectureScape than with the baseline interface. Answers to 7-point Likert scale questions on the overall task experience revealed significant differences between the two interfaces in participants' belief in speed (LectureScape: μ=5.8, Baseline: μ=4.8) and efficiency (LectureScape: μ=6.1, Baseline: μ=4.8). The MWU test shows that both differences were significant at p<0.05 for these questions.

RQ2. Perception of Interaction Data

Generally, participants' comments about watching videos augmented by others' interaction data were positive. Participants noted that *"It's not like cold-watching. It feels like watching with other students."*, and *"[interaction data] makes it seem more classroom-y, as in you can compare yourself to what how other students are learning and what they need to repeat."*

In response to 7-point Likert scale questions about the experience of seeing interaction data, participants indicated that they found such data to be "easy to understand" (μ=5.9), "useful" (5.3), "enjoyable" (5.2), that interaction peaks affected their navigation (5), and that interaction peaks matched their personal points of interest in the video (4.4).

In an open-ended question, we asked participants why they thought interaction peaks occurred. Common reasons provided were that these parts were "confusing" (mentioned by 8 / 12), "important" (6 / 12), and "complex" (4 / 12). Identifying the cause of a peak might be useful because they can enable more customized navigation support. While most participants mentioned that highlighting confusing and important parts would be useful, some noted that personal context may not match the collective patterns. One said, *"If it were a topic where I was not very confident in my own knowledge, I would find it very helpful to emphasize where others have re-watched the video. If however it was a topic I was comfortable with and was watching just to review, I would find it frustrating to have the physical scrolling be slowed down due to others' behavior while watching the video."*

RQ3. Perceived Usability of LectureScape

Many participants preferred having more options in navigating lecture videos. As one learner noted, *"I like all the extra features! I was sad when they got taken away [for the baseline condition]."* Also, when asked if the interface had all the functions and capabilities they expected, participants rated LectureScape (6.4) significantly higher than the baseline interface (4.3) (p<0.001, Z=-3.2 with the MWU test).

However, some expressed that LectureScape was visually complex, and that they would have liked to hide some widgets not in use at the moment. They found it more difficult to use than the baseline (ease of use: 4.7 vs. 6.3, $p<0.001$, $Z=2.7$ with the MWU test). This perception leaves room for improvement in the learnability of the system. A participant commented: *"[LectureScape] is fairly complex and has a lot of different elements so I think it may take a bit of time for users to fully adjust to using the interface to its full potential."* These comments are consistent with our design decision to support collapsible widgets to reduce visual clutter, although the version of LectureScape used in the study had all features activated for the purpose of usability testing.

Due to the limitations of a single-session lab study, few participants actively used personal bookmarks or personal history traces. A longitudinal deployment might be required to evaluate the usefulness of these features.

Navigation Pattern Analysis

Now we provide a detailed analysis of how participants navigated videos during the study. In all tasks, most participants' strategy was to start with an overview, and then focus on some parts in detail. Participants alternated between the overview and focus stages until they found what they were looking for, or covered all major points in the video for summarization.

While the high-level strategy was similar in the two video interfaces, participants' navigation patterns within each of the two stages differed noticeably. With LectureScape, participants used more diverse options for overview and focused navigation, making more directed jumps to important points in the video. With the baseline interface, most participants sequentially scanned the video for overview and clicked on the timeline or transcript for focusing. Another common pattern in the baseline condition was to make short and conservative jumps on the timeline from the beginning of the video, in order not to miss anything important while moving quickly.

In the **overview** stage, most participants tried to scan the video to grasp the general flow and select a few points to review further. One learner described her strategy with LectureScape in this stage: *"having this idea of 'here's where other people have gone back and rewatched, being able to visually skim through very quickly and see titles, main bullet points, and following along with the transcript a little bit as well was definitely helpful."* Although visual scanning did not result in click log entries, our interviews with participants confirm that it was a common pattern. They pointed out three main features in LectureScape that supported overview:

- the 2D timeline with an overall learner activity visualization: *"I could use the 2D overlay to scroll through... I think I took a quick scan through and saw the general overview of what was on the slides."*
- highlight summaries with a strip of screenshots: *"They would get me close to the information I needed. They also made it easier to quickly summarize."*
- the word cloud with main keywords for sections in the video: *"I just looked at the top keywords, then I watched the video to see how [the instructor] uses those keywords."*

After scanning for an overview, participants chose a point in the video to watch further. All of the methods described above provide a single-click mechanism to directly jump to an "important" part of the video. Participants reviewed data-driven suggestions from LectureScape to make informed decisions. The log analysis reveals that participants using LectureScape made direct jumps such as clicking on a specific point in the timeline or a highlight 8.4 times on average per task, in contrast to 5.6 times in the baseline condition.

In the **focus** stage, participants watched a segment in the video and reviewed if content is relevant to the task at hand. In this stage they relied on methods for precise navigation: scrubbing the timeline a few pixels for a second-by-second review, and re-watching video snippets multiple times until they fully comprehend the content. With LectureScape, participants had options to use the slowed-down scrubbing around peaks in the rollercoaster timeline, automatic pinning of the previous slide, and sentence-level playback in search. To navigate back to the previously examined point, participants frequently used timestamped anchors attached to search results, interaction peaks, and highlights.

In summary, with LectureScape, participants used more navigation options in both the overview and focus stages. A learner commented that *"[LectureScape] gives you more options. It personalizes the strategy I can use in the task."* They had more control in which part of the video to watch, which might have led them to believe that they completed the tasks faster and more efficiently.

DISCUSSION AND FUTURE WORK

Availability of interaction data: Discussion throughout this paper assumes the availability of large-scale interaction data. With modern Web technologies, clickstream logging can be easily added with APIs for video event handling.

There remain unanswered questions around interaction data, such as "Will using the data-driven techniques bias the data so that it reinforces premature peak signals and ignores other potentially important ones?", and "How many data points are required until salient peaks and patterns emerge?" Our future work will address these questions through a live deployment on a MOOC platform such as edX. We will also explore other types of interaction data such as active bookmarking and content streams such as voice to enrich video-based learning.

Adaptive video UI: The data-driven techniques introduced in this paper open opportunities for more adaptive and personalized video learning experiences. In this paper, we demonstrated how collective viewership data can change the video interface dynamically, influencing the physical scrubbing behavior, search ranking algorithm, and side-by-side frame display. We envision future video UIs that adapt to collective usage. Also, incorporating interaction data can lead to personalized video learning. Because interaction data is likely to represent an average learner, comparing personal history traces against collective traces may help model the current user more accurately and improve personalization.

Beyond MOOC-style lecture videos: While this paper used MOOC-style lecture videos for demonstration, we believe our techniques can generalize to other types of educational videos such as programming tutorial screencasts, how-to demonstration videos, and health education videos. We expect to apply the techniques introduced in this paper to these videos.

CONCLUSION

This paper introduces a novel concept of designing video interaction techniques by leveraging large-scale interaction data. We present three sets of data-driven techniques to demonstrate the capability of the concept: 2D, non-linear timeline, enhanced in-video search, and a visual summarization method. In a lab study, participants found interaction data to draw attention to points of importance and confusion, and navigated lecture videos with more control and flexibility.

Ultimately, the design techniques we have presented provide enriched alternatives to conventional video navigation. We envision engaging a community of learners in creating a social, interactive, and collaborative video learning environment powered by rich community data.

ACKNOWLEDGMENTS

The authors thank the members of the UID Group at MIT CSAIL for their feedback, and the 6.01x course staff for study advertisement. This work was funded in part by Quanta Computer. The Samsung Fellowship has supported Juho Kim.

REFERENCES

1. Al-Hajri, A., Miller, G., Fong, M., and Fels, S. S. Visualization of personal history for video navigation. In *CHI '14* (2014).
2. Alexander, J., Cockburn, A., Fitchett, S., Gutwin, C., and Greenberg, S. Revisiting read wear: Analysis, design, and evaluation of a footprints scrollbar. In *CHI '09* (2009), 1665–1674.
3. Arman, F., Depommier, R., Hsu, A., and Chiu, M.-Y. Content-based browsing of video sequences. In *MULTIMEDIA '94* (1994), 97–103.
4. Baudisch, P., Cutrell, E., Hinckley, K., and Eversole, A. Snap-and-go: Helping users align objects without the modality of traditional snapping. In *CHI '05* (2005), 301–310.
5. Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *CHI '04* (2004), 519–526.
6. Bostock, M., Ogievetsky, V., and Heer, J. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2301–2309.
7. Carlier, A., Charvillat, V., Ooi, W. T., Grigoras, R., and Morin, G. Crowdsourced automatic zoom and scroll for video retargeting. In *Multimedia '10* (2010), 201–210.
8. Chi, P.-Y. P., Liu, J., Linder, J., Dontcheva, M., Li, W., and Hartmann, B. Democut: generating concise instructional videos for physical demonstrations. In *UIST '13*, ACM (2013).
9. Chorianopoulos, K. Collective intelligence within web video. *Human-centric Computing and Information Sciences 3*, 1 (2013), 10.
10. Dieberger, A., Dourish, P., Höök, K., Resnick, P., and Wexelblat, A. Social navigation: Techniques for building more usable systems. *Interactions 7*, 6 (Nov. 2000), 36–45.
11. Ding, W., and Marchionini, G. A study on video browsing strategies. Tech. rep., College Park, MD, USA, 1997.
12. Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories. In *UIST '10* (2010).
13. Guo, P. J., Kim, J., and Rubin, R. How video production affects student engagement: An empirical study of mooc videos. In *L@S '14* (2014), 41–50.
14. Hill, W. C., Hollan, J. D., Wroblewski, D., and McCandless, T. Edit wear and read wear. In *CHI '92* (1992), 3–9.
15. Hou, X., and Zhang, L. Saliency detection: A spectral residual approach. In *CVPR '07* (2007), 1–8.
16. Hurst, A., Mankoff, J., Dey, A. K., and Hudson, S. E. Dirty desktops: Using a patina of magnetic mouse dust to make common interactor targets easier to select. In *UIST '07* (2007), 183–186.
17. Hürst, W., Götz, G., and Jarvers, P. Advanced user interfaces for dynamic video browsing. In *MULTIMEDIA '04* (2004), 742–743.
18. Jackson, D., Nicholson, J., Stoeckigt, G., Wrobel, R., Thieme, A., and Olivier, P. Panopticon: A parallel video overview system. In *UIST '13* (2013), 123–130.
19. Jansen, M., Heeren, W., and van Dijk, B. Videotrees: Improving video surrogate presentation using hierarchy. In *CBMI 2008*, IEEE (June 2008), 560–567.
20. Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., and Miller, R. C. Understanding in-video dropouts and interaction peaks in online lecture videos. In *L@S '14* (2014), 31–40.
21. Kim, J., Nguyen, P., Weir, S., Guo, P., Gajos, K., and Miller, R. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *CHI '14* (2014).
22. Li, F. C., Gupta, A., Sanocki, E., He, L.-w., and Rui, Y. Browsing digital video. In *CHI '00* (2000), 169–176.
23. Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. Twitinfo: aggregating and visualizing microblogs for event exploration. In *CHI '11* (2011), 227–236.
24. Masui, T., Kashiwagi, K., and Borden, IV, G. R. Elastic graphical interfaces to precise data manipulation. In *CHI '95* (1995), 143–144.
25. Matejka, J., Grossman, T., and Fitzmaurice, G. Swift: Reducing the effects of latency in online video scrubbing. In *CHI '12* (2012), 637–646.
26. Matejka, J., Grossman, T., and Fitzmaurice, G. Patina: Dynamic heatmaps for visualizing application usage. In *CHI '13* (2013), 3227–3236.
27. Matejka, J., Grossman, T., and Fitzmaurice, G. Swifter: Improved online video scrubbing. In *CHI '13* (2013), 1159–1168.
28. Mertens, R., Farzan, R., and Brusilovsky, P. Social navigation in web lectures. In *HYPERTEXT '06* (2006), 41–44.
29. Monserrat, T.-J. K. P., Zhao, S., McGee, K., and Pandey, A. V. Notevideo: Facilitating navigation of blackboard-style lecture videos. In *CHI '13* (2013), 1139–1148.
30. Nancel, M., and Cockburn, A. Causality: A conceptual model of interaction history. In *CHI '14* (2014).
31. Nguyen, C., Niu, Y., and Liu, F. Video summagator: An interface for video summarization and navigation. In *CHI '12* (2012), 647–650.
32. Nicholson, J., Huber, M., Jackson, D., and Olivier, P. Panopticon as an elearning support search tool. In *CHI '14* (2014).
33. Olsen, D. R., and Moon, B. Video summarization based on user interaction. In *EuroITV '11* (2011), 115–122.
34. Pongnumkul, S., Wang, J., Ramos, G., and Cohen, M. Content-aware dynamic timeline for video browsing. In *UIST '10* (2010), 139–142.
35. Ramos, G., and Balakrishnan, R. Fluid interaction techniques for the control and annotation of digital video. In *UIST '03* (2003), 105–114.
36. Risko, E., Foulsham, T., Dawson, S., and Kingstone, A. The collaborative lecture annotation system (clas): A new tool for distributed learning. *Learning Technologies, IEEE Transactions on 6*, 1 (2013), 4–13.
37. Scott MacKenzie, I., and Riddersma, S. Effects of output display and control–display gain on human performance in interactive systems. *Behaviour & Information Technology 13*, 5 (1994), 328–337.
38. Shaw, R., and Davis, M. Toward emergent representations for video. In *Multimedia '05* (2005), 431–434.
39. Yew, J., Shamma, D. A., and Churchill, E. F. Knowing funny: genre perception and categorization in social video sharing. In *CHI '11* (2011), 297–306.

Cobi: A Community-Informed Conference Scheduling Tool

Juho Kim[1] **Haoqi Zhang**[1,2] **Paul André**[3] **Lydia B. Chilton**[4] **Wendy Mackay**[5]
Michel Beaudouin-Lafon[6] **Robert C. Miller**[1] **Steven P. Dow**[3]

[1]MIT CSAIL
Cambridge, MA, USA
{juhokim, rcm}@mit.edu

[2]Northwestern University
Evanston, IL, USA
hq@eecs.northwestern.edu

[3]HCI Institute, CMU
Pittsburgh, PA, USA
{pandre, spdow}@cs.cmu.edu

[4]University of Washington
Seattle, WA, USA
hmslydia@cs.uw.edu

[5]INRIA
Orsay, France
mackay@lri.fr

[6]Université de Paris-Sud
Orsay, France
mbl@lri.fr

ABSTRACT

Effectively planning a large multi-track conference requires an understanding of the preferences and constraints of organizers, authors, and attendees. Traditionally, the onus of scheduling the program falls on a few dedicated organizers. Resolving conflicts becomes difficult due to the size and complexity of the schedule and the lack of insight into community members' needs and desires. *Cobi* presents an alternative approach to conference scheduling that engages the entire community in the planning process. Cobi comprises (a) communitysourcing applications that collect preferences, constraints, and affinity data from community members, and (b) a visual scheduling interface that combines communitysourced data and constraint-solving to enable organizers to make informed improvements to the schedule. This paper describes Cobi's scheduling tool and reports on a live deployment for planning CHI 2013, where organizers considered input from 645 authors and resolved 168 scheduling conflicts. Results show the value of integrating community input with an intelligent user interface to solve complex planning tasks.

Author Keywords

Cobi; conference scheduling; mixed-initiative; constraint solving; crowdsourcing; community; communitysourcing

ACM Classification Keywords

H5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces - Graphical user interfaces

INTRODUCTION

Creating a compelling schedule for a large conference is a difficult task. Hundreds of accepted submissions must be scheduled into sessions across multiple days and rooms, while accounting for the multi-faceted preferences and constraints of organizers, authors, and attendees. Organizers aim to create thematic sessions, avoid scheduling related papers or the same presenters in opposing sessions, and generally make the program interesting for attendees with different interests.

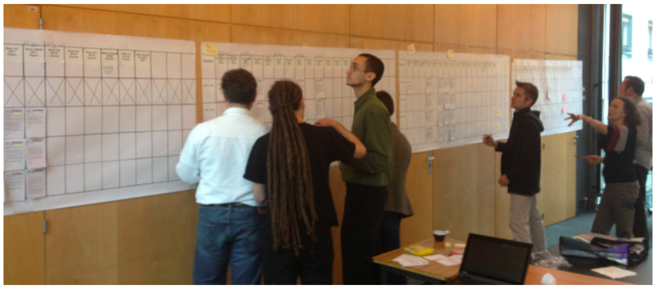


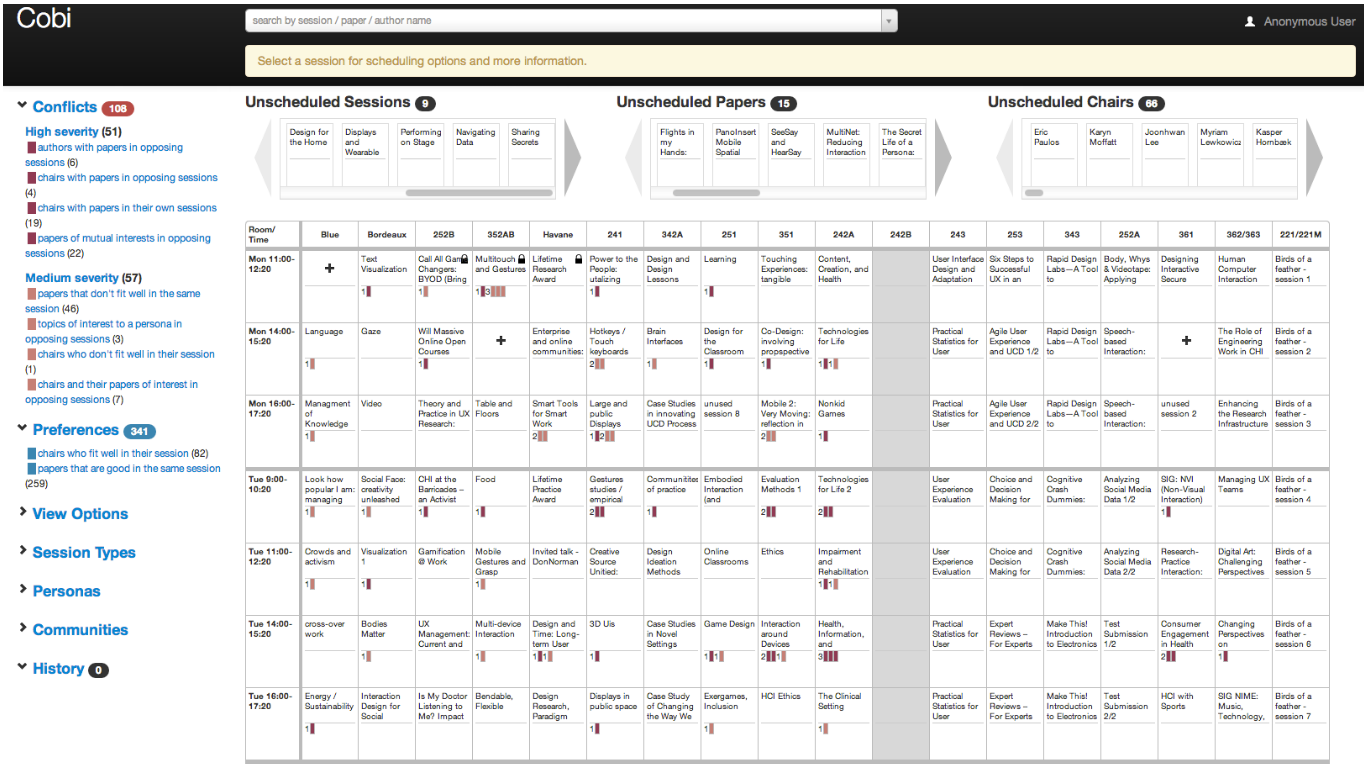
Figure 1. A small group of organizers and associate chairs create a preliminary CHI program on paper.

To better understand this challenge, we observed the schedule creation process for CHI, the largest human-computer interaction conference: CHI 2013 received over 2260 submissions and accepted more than 500 to be scheduled in 16 simultaneous sessions spanning four days. Scheduling CHI involves two stages. Once papers are accepted, a small group of associate chairs help the conference organizers to roughly create categories and suggest sessions. Over the next two days, the organizers and a few assistants build a rough preliminary schedule (see Figure 1). The process is paper-based, collaborative, and time-consuming; its output is highly dependent upon the specific knowledge of the individuals in the room. In stage two, organizers refine the rough schedule to create the final program. They attempt to resolve conflicts, handle stray papers, respond to last minute changes, and generally look for ways to improve the program. The organizers use a script to check that no presenter is scheduled to be in two places at once, but otherwise, all changes are made manually. Interviews with past organizers revealed that the process was extremely time-consuming, and that resolving conflicts was "painstaking" due to schedule complexity and the lack of feedback on whether changes resolved existing conflicts or created new ones.

Despite organizers' best intentions and efforts, previous CHI programs often contained incoherent sessions, similarly-

UIST'13, October 8–11, 2013, St. Andrews, United Kingdom.

ACM 978-1-4503-2268-3/13/10...$15.00.
http://dx.doi.org/10.1145/2501988.2502034



Cobi

search by session / paper / author name

Anonymous User

Select a session for scheduling options and more information.

Conflicts 108

High severity (51)
- ▮ authors with papers in opposing sessions (6)
- ▮ chairs with papers in opposing sessions (4)
- ▮ chairs with papers in their own sessions (19)
- ▮ papers of mutual interests in opposing sessions (22)

Medium severity (57)
- ▮ papers that don't fit well in the same session (46)
- ▮ topics of interest to a persona in opposing sessions (3)
- ▮ chairs who don't fit well in their session (1)
- ▮ chairs and their papers of interest in opposing sessions (7)

Preferences 341
- ▮ chairs who fit well in their session (82)
- ▮ papers that are good in the same session (259)

View Options

Session Types

Personas

Communities

History 0

Unscheduled Sessions 9

Design for the Home | Displays and Wearable | Performing on Stage | Navigating Data | Sharing Secrets

Unscheduled Papers 15

Flights in my Hands: | PanoInsert Mobile Spatial | SeeSay and HearSay | MultiNet: Reducing Interaction | The Secret Life of a Persona:

Unscheduled Chairs 66

Eric Paulos | Karyn Moffatt | Joonhwan Lee | Myriam Lewkowicz | Kasper Hornbæk

Room/ Time	Blue	Bordeaux	252B	352AB	Havane	241	342A	251	351	242A	242B	243	253	343	252A	361	362/363	221/221M
Mon 11:00-12:20	+	Text Visualization 1▮	Call All Game 🔒 Changers: BYOD (Bring 1▮	Multitouch 🔒 and Gestures 1▮3▮▮▮	Lifetime 🔒 Research Award	Power to the People: utalizing 1▮	Design and Design Lessons	Learning 1▮	Touching Experiences: tangible	Content, Creation, and Health		User Interface Design and Adaptation	Six Steps to Successful UX in an	Rapid Design Labs—A Tool to	Body, Whys & Videotape: Applying	Designing Interactive Secure	Human Computer Interaction	Birds of a feather - session 1
Mon 14:00-15:20	Language 1▮	Gaze	Will Massive Online Open Courses 1▮	+	Enterprise and online communities:	Hotkeys / Touch keyboards 2▮▮	Brain Interfaces 1▮	Design for the Classroom 1▮	Co-Design: involving propspective 1▮	Technologies for Life 1▮1▮		Practical Statistics for User	Agile User Experience and UCD 1/2	Rapid Design Labs—A Tool to	Speech-based Interaction:	+	The Role of Engineering Work in CHI	Birds of a feather - session 2
Mon 16:00-17:20	Managment of Knowledge 1▮	Video	Theory and Practice in UX Research:	Table and Floors	Smart Tools for Smart Work 2▮▮	Large and public Displays 1▮2▮▮	Case Studies in innovating UCD Process	unused session 8	Mobile 2: Very Moving: reflection in 2▮▮	Nonkid Games 1▮		Practical Statistics for User	Agile User Experience and UCD 2/2	Rapid Design Labs—A Tool to	Speech-based Interaction:	unused session 2	Enhancing the Research Infrastructure	Birds of a feather - session 3
Tue 9:00-10:20	Look how popular I am: managing 1▮	Social Face: creativity unleashed 1▮	CHI at the Barricades – an Activist 1▮	Food 1▮	Lifetime Practice Award	Gestures studies / empirical 2▮▮	Communitites of practice 1▮	Embodied Interaction (and	Evaluation Methods 1 2▮▮	Technologies for Life 2 2▮▮		User Experience Evaluation	Choice and Decision Making for	Cognitive Crash Dummies:	Analyzing Social Media Data 1/2	SIG: NVI (Non-Visual Interaction) 1▮	Managing UX Teams	Birds of a feather - session 4
Tue 11:00-12:20	Crowds and activism 1▮	Visualization 1 1▮	Gamification @ Work	Mobile Gestures and Grasp 1▮	Invited talk - DonNorman	Creative Source Unitied:	Design Ideation Methods	Online Classrooms	Ethics	Impairment and Rehabilitation 1▮1▮		User Experience Evaluation	Choice and Decision Making for	Cognitive Crash Dummies:	Analyzing Social Media Data 2/2	Research-Practice Interaction:	Digital Art: Challenging Perspectives	Birds of a feather - session 5
Tue 14:00-15:20	cross-over work	Bodies Matter 1▮	UX Management: Current and	Multi-device Interaction 1▮	Design and Time: Long-term User 1▮1▮	3D Uis 1▮	Case Studies in Novel Settings	Game Design 1▮1▮	Interaction around Devices 2▮▮1▮	Health, Information, and 3▮▮▮		Practical Statistics for User	Expert Reviews – For Experts	Make This! Introduction to Electronics	Test Submission 1/2	Consumer Engagement in Health 2▮▮	Changing Perspectives on 1▮	Birds of a feather - session 6
Tue 16:00-17:20	Energy / Sustainability 1▮	Interaction Design for Social	Is My Doctor Listening to Me? Impact	Bendable, Flexible	Design Research, Paradigm	Displays in public space 1▮	Case Study of Changing the Way We	Exergames, Inclusion 1▮	HCI Ethics	The Clinical Setting 1▮		Practical Statistics for User	Expert Reviews – For Experts	Make This! Introduction to Electronics	Test Submission 2/2	HCI with Sports	SIG NIME: Music, Technology,	Birds of a feather - session 7

Figure 2. Cobi's scheduling tool consists of the top panel (top), the sidebar (left), and the unscheduled panel and main schedule table (right).

themed sessions that run in parallel, and author-specific conflicts. Several aspects of the process contribute to these problems. First, due to the organic nature of how organizers make connections between papers in stage one, many sessions have odd papers mixed in. Second, because the process does not capture affinities between papers in different sessions, it is difficult for organizers to make scheduling changes that lead to more cohesive sessions. Third, organizers are often unaware of the preferences of authors and attendees. This can lead to sessions of interest being scheduled at the same time. Finally, the lack of tools for managing constraints and the sheer size of the schedule make it difficult for organizers to make informed decisions when finalizing the schedule.

Cobi addresses these challenges by drawing on the people and expertise within the community, and embedding intelligence for resolving conflicts into a scheduling interface. The Cobi system consists of a collection of communitysourcing applications that elicit preferences, constraints, and affinity data from program committee members and authors, and an intelligent scheduling tool that provides organizers with helpful context and suggestions for improving the schedule. By engaging the community in the planning process, Cobi exposes the preferences and constraints of its members to the organizers and makes the planning process more transparent.

Cobi's scheduling tool (Figure 2) integrates community preferences and constraints with constraint-solving intelligence into a new kind of *community-informed mixed-initiative system*. The interface helps organizers visually spot problems and resolve them in the schedule. It highlights general, high-level conflicts such as scheduling a presenter in two opposing sessions. It also exposes more detailed, communitysourced preferences such as scheduling together papers that authors feel fit well in a session with their paper. When manipulating the schedule (e.g., assigning, moving, and swapping sessions, papers, and session chairs), the interface uses a constraint solver to help organizers make informed decisions by recommending edits that best improve the schedule and visualizing the consequences of potential edits. Organizers drive the system by applying their personal knowledge, choosing which problems to focus on, and making final decisions.

We deployed the Cobi system for planning CHI 2013. We recruited associate chairs to group sets of related papers, and authors to identify papers of interest and those that fit well in a session with their own paper. The process collected 1722 paper affinities from 64 associate chairs and 8651 preferences and constraints from 645 authors (covering 87% of accepted submissions). In addition, we asked candidate session chairs to submit their representative papers to determine their fit with sessions. The organizers used Cobi's scheduling tool to improve the preliminary schedule and assign session chairs. The tool helped the organizers resolve 168 conflicts as they created the final schedule. They found that the scheduling tool greatly simplified conflict resolution, while allowing them to combine their own knowledge with the machine intelligence and the community's input.

The paper proceeds as follows. We first discuss related work in communitysourcing and conference scheduling. We then share findings from a preliminary study and identify key design goals. We present Cobi's scheduling tool, focusing on the integration of community data, machine intelligence, and end-user interface. We report on our deployment at CHI 2013 and discuss the key lessons learned. The paper concludes with notes on future research directions.

RELATED WORK

Our work seeks to tailor tasks to the inherent incentives, interests, and expertise of diverse groups within a community. We draw from the broad literature on encouraging community contributions, both in online [10] and physical spaces [5], and on tasks ranging from collecting scientific data [4] to co-designing public transportation services [16]. In our work, community members contribute to solving a specific problem whose solution affects themselves and the community at large. With Cobi, we are exploring incentives, methods, and interfaces for collecting and incorporating multidimensional preferences and constraints from large numbers of individuals within a community into a single, cohesive outcome.

Previous research introduced mixed-initiative solutions to complex tasks such as aircraft scheduling [3] and manufacturing task scheduling [7] by modeling expert knowledge. To this line of research, Cobi contributes a community-driven approach, which raises the unique design challenges of incentivizing community members to express preferences, and mediating, encoding, visualizing, and acting on noisy and diverse community input. Interactive machine learning (IML) approaches such as CueT [1] and E-mazing [15] use a combination of human labor and machine learning. However, they take opposite approaches to the problem. IML supervises an algorithm with human input, while Cobi helps conference organizers make informed decisions with computations shown as visual feedback.

Automated scheduling is a well-studied problem in both computer science and operations research. Specific to conference scheduling, Sampson et al. [13] introduced formulations for maximizing the number of talks of interest attendees can attend. For the related problem of course scheduling, Murray et al. [11] introduced formulations for minimizing student and instructor conflicts subject to scheduling constraints. While automated scheduling is appropriate when the parameters and constraints of the optimization problem are well-specified, our interviews with past CHI organizers show that they attempt to tackle soft constraints and other tacit considerations. With Cobi's mixed-initiative, interactive optimization [6, 14], the machine plays a supporting role, providing intelligence and feedback for detecting problems and resolving conflicts. Organizers can thus better interpret and act on the community's input, which can be overwhelmingly rich, subjective, and incomplete at the same time.

Jacob et al. [8] developed a tangible interface for manipulating a conference schedule and checking constraints on a physical grid layout. Cobi extends this approach by accounting for input from the larger conference community.

There are several commercial systems for conference and course scheduling. One example is Confex's scheduling tool (`confex.com`), which detects and highlights hard conflicts in the schedule (e.g., scheduling a presenter in simultaneous sessions) but makes no suggestions about how to resolve them. Another example is UniTime (`unitime.org`), which first computes an optimal course schedule to minimize conflicts and then allows a user to make fine-grained adjustments while seeing the effect on conflicts. Our work presents an alternative approach in which the user is in control at all times while the system detects conflicts and provides suggestions for resolving them.

PRELIMINARY STUDY AND DESIGN GOALS

Our research team conducted hour-long semi-structured interviews and exchanged emails with five past and current CHI organizers, two of whom are co-authors on this paper. Discussions centered around the planning process at CHI and focused in particular on existing challenges and potential solutions. Conversations revealed three high-level goals that drove the design of the Cobi system and its scheduling tool:

Understanding paper affinities. Organizers stressed that "papers fit into sessions in complex ways" and that "getting a session together that makes sense is hard." While the in-person meeting created sessions that are mostly cohesive, organizers still needed to break open some sessions. Organizers noted that this is a "major pain point" and that it is "very hairy to break up a session" because swapping a paper with another paper requires each paper to fit well in the other's session. In order to capture paper affinities across sessions, organizers noted that you would need contributors "knowledgeable enough in the field to know that papers should or shouldn't be in the same session." Cobi draws on input from paper authors, who we hypothesize would know what other papers fit well in a session with their own.

Detecting conflicts automatically and providing feedback for resolving them. Organizers found it particularly difficult to know the consequences of moving a paper or session in the schedule, which requires reasoning about the conflicts that would be created in addition to those that would be resolved. One organizer noted that she "would (painstakingly) solve those [conflicts and] re-run [a constraint-checking script], usually showing that the problems I had solved had generated other author conflicts." In order to avoid thrashing and frustration, Cobi recommends moves and swaps that resolve the most conflicts and allows organizers to preview the effect of possible edits on conflicts.

Keeping the human in control. While an automated constraint solver can be used to resolve known conflicts, previous organizers felt that taking a purely automated approach would be impractical and would fail to capture the "many semantic constraints that are hard to express using machine understandable ways." The organizers also stressed the importance of being able to make sense of the schedule so that they can apply their knowledge and weigh the various demands of the community while scheduling.

COBI'S SCHEDULING TOOL

Once the program committee determines the accepted papers, Cobi's communitysourcing applications collect preferences, constraints, and affinity data from community members. This

Example Constraints & Preferences	*Possible Source*	*Rationale*
Papers that don't fit well together shouldn't be in the same session	Authors	Authors know what papers are related to theirs and care about which end up in a session with their own.
Papers of mutual interest shouldn't be in opposing sessions	Attendees	Knowing what attendees want to see can avoid scheduling talks of interest at the same time.
Chairs' area of research should match the topic of their session	Chairs' papers	We can collect papers from potential session chairs and check if they are related to the papers in a session.

Table 1. Examples of preferences and constraints encoded in Cobi that can be collected from community members.

input is then encoded and presented in Cobi's scheduling tool to help users (conference organizers) resolve conflicts and improve the schedule.

Encoding Preferences and Constraints

Cobi supports preferences and constraints over attributes at three entity levels: sessions, papers, and chairs. For example, a constraint may specify sessions that should not be concurrent (e.g., "sessions of interest to the ICT4D community should not oppose one another"), and a preference may state that a chair is a good fit for a session (e.g., "James Sysmaster is a good fit for the systems session"). At a high level, the goal is to create a schedule that violates few constraints and meets many preferences.

In early prototype testing, we found that most constraints and preferences of interest can be stated as conditions on a single entity or a pair of entities (e.g., "George Latewaker prefers to chair sessions in the afternoon" or "sessions on crowdsourcing and social computing should not oppose one another"). Further simplifying matters, paired-entity constraints of interest tend to describe relations over entities when they are in the same time or room, suggesting that we need only check conflicts between entities in such cases. Currently Cobi supports encoding paired entity constraints of the form "x and y should [not] be in the same session" and "w and z should [not] oppose one another," where x and y can be papers and chairs and w and z can be sessions, papers, and chairs.

Some constraints (and likewise preferences) may be *system-defined*, which refers to high-level, overarching constraints that can be stated using data from a submission management system (e.g., "a presenter should not be scheduled in opposing sessions"). Other constraints may be *community-defined*, which refers to more specific and perhaps more subjective wishes stated by community members (e.g., "Sessions A and B should be scheduled apart because Mary Liker is interested in papers in both sessions"). Table 1 provides examples of community constraints and preferences encoded in the current Cobi prototype and potential sources of community input that can be used to instantiate them.

While encoding system-defined constraints is straightforward, encoding community-provided constraints requires taking into account the potential sparsity, diversity, and subjectivity of the collected data. We may collect thousands of preferences and constraints, some of which are in direct conflict with others (e.g., an author may feel that his paper fits well in a session with another paper whose author disagrees). To account for such issues, an *input-mediation layer* aggregates responses before adding a constraint or preference in Cobi. For instance, we add a preference for two papers to be in the same session, only if the majority of people providing data about both papers agreed that they are related. We restrict two papers from being in opposing sessions only when many people express interest in seeing both papers.

In addition to managing the complexity of subjective data, the input-mediation layer can help focus the user's attention on salient constraints that matter to many community members. It can also be used to capture variance in the data and note an absence of data, so that the user can know when not to rely excessively on community input. The goal is to capture the community input at a level where the user can best act upon it: too little mediation makes it difficult to understand the community's wishes, and too much may end up hiding some of the useful information contained in the data.

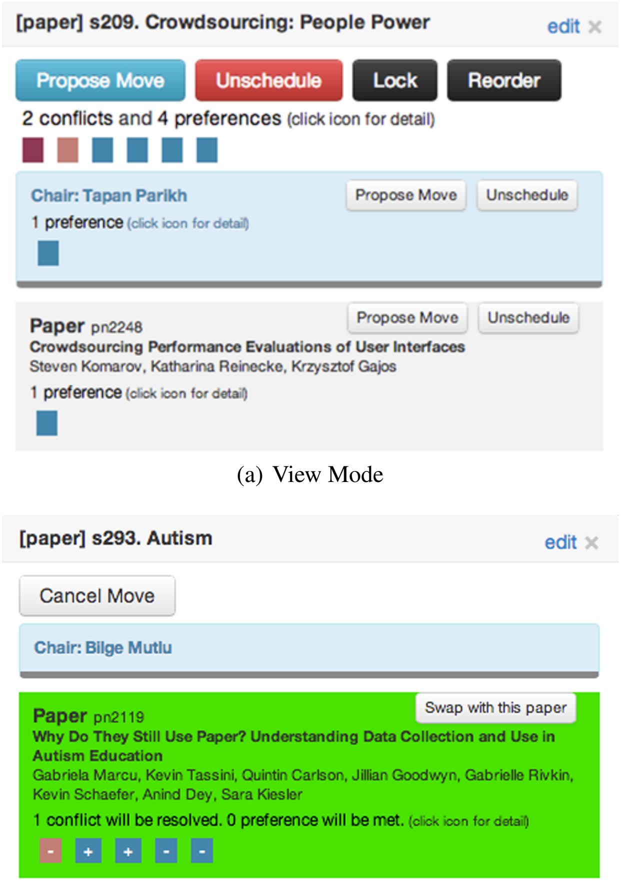
Scheduling Interface

Cobi's scheduling tool (Figure 2) enables manipulating, scanning, and reviewing the schedule with support for conflict resolution and multi-faceted views. The interface keeps the user in control and provides advice on entity moves and swaps. It consists of three components: the top panel, the sidebar, and the unscheduled panel and main schedule table.

The top panel (Figure 2, top) allows the user to search for entities and displays information about the current operation. The sidebar (Figure 2, left) contains view modes and faceted browsing options to help the user analyze the current schedule. *Conflicts* and *Preferences* display conflicts and satisfied preferences in the current schedule and their counts. They are separated by type and grouped based on their severity. Counts update immediately following any change to the schedule, and provide immediate feedback on the effects of the user's actions on conflicts.

View options display different aspects of the schedule and help the user spot issues requiring attention. For example, the default *Conflict* view shows icons for conflicts that involve entities within a session. (Figure 2, right). Clicking on the *Duration* view option displays the length of each session, and allows the user to quickly identify ones with too many or too few papers. *Personas* and *Communities* allow the user to skim the schedule for various interest-based subgroups and check if the schedule is well-distributed across subgroups. The *History* option keeps track of all the scheduling operations and who performed them.

The unscheduled panel displays unscheduled sessions, papers, and session chairs. In addition to holding the entities to be scheduled, in initial testing we found that users needed



[paper] s209. Crowdsourcing: People Power

2 conflicts and 4 preferences (click icon for detail)

Chair: Tapan Parikh

1 preference (click icon for detail)

Paper pn2248

Crowdsourcing Performance Evaluations of User Interfaces

Steven Komarov, Katharina Reinecke, Krzysztof Gajos

1 preference (click icon for detail)

(a) View Mode

[paper] s293. Autism

Chair: Bilge Mutlu

Paper pn2119

Why Do They Still Use Paper? Understanding Data Collection and Use in Autism Education

Gabriela Marcu, Kevin Tassini, Quintin Carlson, Jillian Goodwyn, Gabrielle Rivkin, Kevin Schaefer, Anind Dey, Sara Kiesler

1 conflict will be resolved. 0 preference will be met. (click icon for detail)

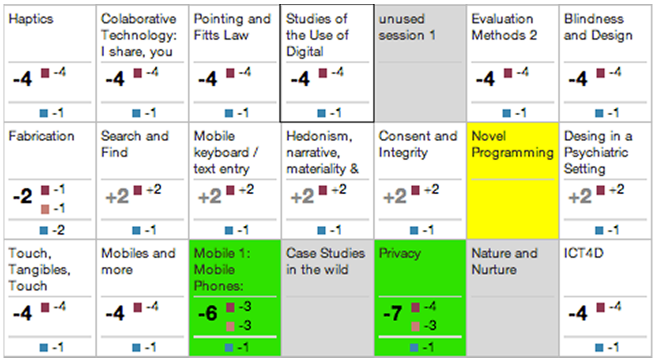
(b) Move Mode

Figure 3. Inner-session view in view mode and move mode. A recommended paper move is highlighted in green.

a scratch space to construct a session without having to worry about where it is placed in the schedule. The unscheduled panel serves as this scratch space.

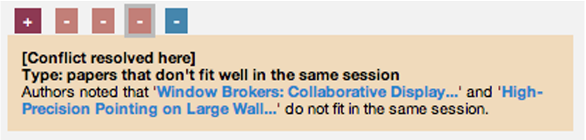
The main schedule table displays the entire schedule in a time table (Figure 2, right). Each cell displays a session name and additional information based on the view option (e.g., *Conflicts*). Clicking on a session displays details of the session (Figure 3(a)), which includes conflict information and details on its papers and chair. Here the user is provided with options for scheduling entities, unscheduling entities, swapping entities, reordering papers, editing titles, and locking sessions.

When working to resolve conflicts related to an entity, the user can click *Propose Move* on a session, paper, or session chair and enter *move mode*, which displays previews of the consequences of moving to an empty slot or swapping with a candidate entity (Figure 4). Each target cell displays the net change in the number of conflicts for swapping with entities in the cell, and cells highlighted in green represent recommended moves that would lead to the largest reduction in the number of conflicts. The user can scan different options, and click on cells to examine in detail the consequences of potential moves (Figure 3(b)).



Haptics -4 ■ -4 ■ -1	Colaborative Technology: I share, you -4 ■ -4 ■ -1	Pointing and Fitts Law -4 ■ -4 ■ -1	Studies of the Use of Digital -4 ■ -4 ■ -1	unused session 1	Evaluation Methods 2 -4 ■ -4 ■ -1	Blindness and Design -4 ■ -4 ■ -1
Fabrication -2 ■ -1 ■ -1 ■ -2	Search and Find +2 ■ +2 ■ -1	Mobile keyboard / text entry +2 ■ +2 ■ -1	Hedonism, narrative, materiality & +2 ■ +2 ■ -1	Consent and Integrity +2 ■ +2 ■ -1	Novel Programming	Desing in a Psychiatric Setting +2 ■ +2 ■ -1
Touch, Tangibles, Touch -4 ■ -4 ■ -1	Mobiles and more -4 ■ -4 ■ -1	Mobile 1: Mobile Phones: -6 ■ -3 ■ -3 ■ -1	Case Studies in the wild	Privacy -7 ■ -4 ■ -3 ■ -1	Nature and Nurture	ICT4D -4 ■ -4 ■ -1

Figure 4. Move preview displays the change in the number of conflicts and preferences should the user swap the source session (shown in yellow) with each of the candidate target sessions. The system recommends sessions that minimize conflicts by highlighting them in green.



[Conflict resolved here]
Type: papers that don't fit well in the same session
Authors noted that 'Window Brokers: Collaborative Display...' and 'High-Precision Pointing on Large Wall...' do not fit in the same session.

Figure 5. In move mode, conflict details preview the consequence of making a swap. By clicking on individual icons and following navigation links, the user can understand the specific conflicts and preferences that would be added or removed by making this swap.

For each empty target or target entity, Cobi displays an icon for each conflict that would be added or removed at the source and target by making the move. Clicking on the icon shows detailed information about the selected conflict or preference (Figure 5). Cobi adds links to all entities involved in the conflict or preference, so that clicking on the link highlights the selected entity. Since conflicts created or removed may also involve entities other than those being moved (e.g., a conflicting paper in an opposing session), this helps the user carefully understand which conflicts and preferences will be violated or met for which entities. Upon making a decision, the system displays the change briefly to allow the user to make sense of their decision, and returns to the view mode with updated conflict counts in the sidebar.

By using a combination of overview and detailed views, Cobi's scheduling tool aims to support quick scans as well as detailed investigations. It leaves the user in control of selecting which entities to work on and in what order, while making potential problems in the schedule evident via the conflict view and other view options. When making schedule changes, the system helps the user narrow down the set of candidates to consider, and understand visually the consequences of all possible moves on conflicts in the schedule. Since communitysourced data may be noisy or incomplete and the user may weigh various factors beyond the encoded preferences and constraints, Cobi leaves it to the user to make final scheduling decisions by applying their knowledge and making sense of recommendations from the tool.

Implementation

Cobi associates with each type of preference or constraint a lookup table containing pairs of entities that would be in conflict if particular conditions are met. Since Cobi only encodes constraints within a timeslot, conflict checking and resolution can be performed by simply taking pairs of entities in the same timeslot in the schedule and using the lookup tables to determine if they are in conflict. When computing the consequences of moves and swaps, the system uses the same lookup tables to determine which conflicts involving the source and target entities would be added or removed.

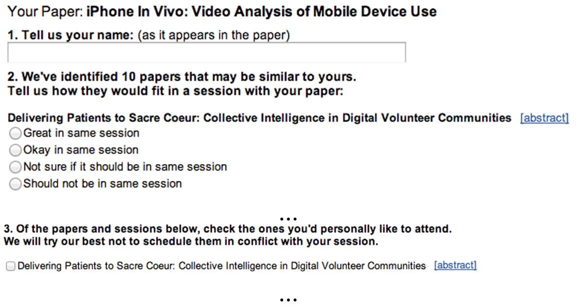
To help the user make sense of potential conflicts in the schedule, each constraint is associated with a *template message* that is instantiated with the entities in conflict when a conflict is detected. For example, a template message may state that "authors noted that x and y do not fit in the same session" and be instantiated with papers x and y that are in the same session and in the corresponding lookup table.

The scheduling tool allows multiple users to collaborate synchronously or asynchronously. The system keeps consistent transaction records on the database and pushes changes to users as they interact with the system. In cases of simultaneous, conflicting edits, the interface displays a message to the user with the failed operation, performs a local rollback, and updates with changes other users have made. Cobi also provides a polling API for other systems or interfaces to access its schedule state, which is useful when alternative visualizations or output devices are available (as in our deployment). The frontend web interface is built with HTML5, CSS3, and Javascript using the jQuery and Bootstrap toolkits.

DEPLOYMENT AND EVALUATION

We deployed the Cobi system for scheduling CHI 2013. Prior to deployment, a small group of organizers and associate chairs met in early December to produce a preliminary schedule by clustering accepted papers and making initial sessions. We deployed Cobi's communitysourcing applications between January 6 and February 12, 2013 to collect preferences, constraints, and affinity data from associate chairs, authors, and session chairs. This data was then encoded into Cobi's scheduling interface. Organizers used Cobi over a period of 42 days from February 10 to March 23, 2013. They took into account the community input, resolved session, paper, and session chair conflicts, and generally worked to improve the initial schedule.

To better understand the scheduling experience with Cobi, we collected quantitative and qualitative data from the organizers. We logged all operations the organizers executed using Cobi. A log entry includes the user responsible for the action, the action type, affected sessions or papers, and a snapshot of conflict counts as a result of the action. At the end of the deployment, we reflected on the process with each of the three CHI 2013 organizers individually (two are co-authors of this paper). Each session lasted 60-120 minutes, and was audio-recorded for later analysis. The organizers talked about high-level goals in scheduling, walked through the scheduling process, and described specific subtasks they were involved in. During the discussion they also tested the latest version of the Cobi scheduling tool that encoded the authorsourcing data, and provided feedback on the experience and usability.



Your Paper: **iPhone In Vivo: Video Analysis of Mobile Device Use**

1. Tell us your name: (as it appears in the paper)

2. We've identified 10 papers that may be similar to yours.
Tell us how they would fit in a session with your paper:

Delivering Patients to Sacre Coeur: Collective Intelligence in Digital Volunteer Communities [abstract]

○ Great in same session
○ Okay in same session
○ Not sure if it should be in same session
○ Should not be in same session

...

3. Of the papers and sessions below, check the ones you'd personally like to attend.
We will try our best not to schedule them in conflict with your session.

□ Delivering Patients to Sacre Coeur: Collective Intelligence in Digital Volunteer Communities [abstract]

...

Figure 6. Authors were presented with a custom list of 20 papers and asked to judge which are related to their paper or of interest to them.

System-defined Preferences and Constraints

For sessions and papers, we used data from the submission management system to encode two constraints that sought to avoid scheduling paper authors in opposing sessions and sessions of interest to a persona in opposing sessions. The former constraint seeks to ensure that no presenter has to be in two places at once and that all authors can see their papers presented. The latter constraint seeks to keep sessions on a particular area of interest apart in the schedule.

For session chairs, we used data from the submission management system to encode constraints stating that chairs should not have papers in opposing sessions (for the same reason as authors), and that they should not chair sessions in which they have a paper (to avoid perceived conflict of interest).

Collecting and Encoding Community Input

We deployed three communitysourced initiatives that collected input from associate chairs, authors, and session chairs. To better understand paper affinities, we first recruited associate chairs to cluster papers in their area of expertise. The process collected 1722 paper affinities from 64 associate chairs (ACs). We then invited authors of accepted papers to identify papers that would fit well in a session with their own and that they are interested in seeing at CHI (Figure 6). To produce a small list of papers for authors to judge, we seeded suggestions based on affinities identified by ACs and by running TF-IDF (Term Frequency - Inverse Document Frequency) [9] comparisons on paper titles and abstracts. The process collected 8651 preferences and constraints from 645 authors, which covered 87% of accepted submissions. The high response rate suggests that authors were inherently interested in seeing their paper in a session with related papers and thus willing to contribute. For more information on the committee and authorsourcing stages, as well as empirical comparison of affinity creation methods, see André et al. [2].

Taking the collected authorsourcing data, Cobi's input mediation layer filtered and aggregated preferences and conflicts so that only those submitted by multiple authors were encoded. For papers, this led to encoding 923 constraints of the form "papers x and y are of mutual interests and shouldn't be scheduled in opposing sessions," 651 constraints of the form "papers x and y do not fit well in the same session," and 805 preferences of the form "papers x and y are good in the same session." For authors who also served as session chairs, we

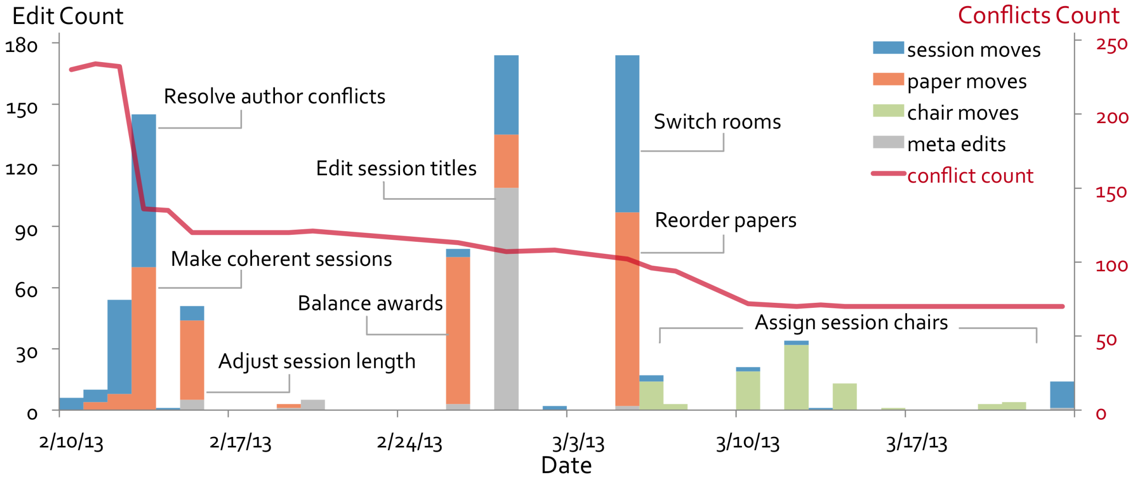


Figure 7. Representative subtasks during the CHI 2013 scheduling process are shown with associated operation types from the interface log. For example, the "Resolve author conflicts" subtask is associated with a cluster of session swap operations. Session, paper, and chair edits indicate scheduling operations involving those entities. Meta edits indicate non-scheduling operations, such as editing session titles and locking or unlocking sessions.

also added 243 constraints of the form "chair x is interested in a paper y in an opposing session." Due to time constraints in the deployment schedule, only the chair-related constraints were visible to organizers in the version of Cobi's scheduling tool they were using. For resolving other community-defined author conflicts, the organizers relied on visualizing the authorsourced data externally without the preview and recommendation support from Cobi.

In addition to the authorsourcing data, we collected representative paper samples from 165 potential session chairs, which we used to compute affinity measures on how well they may fit as chairs for a session. Using TF-IDF similarity between each session chair's papers and each session's papers, we produced affinity scores between chairs and sessions. Cobi encoded the affinity information as a preference for assigning a chair to a session when they are among the top 5 by affinity score, and a constraint when they are out of the top 25. To compute an initial assignment, we solved a linear optimization program to compute an assignment of session chairs that maximizes the sum of affinity scores.

Scheduling Process

Cobi was used in a number of scheduling meetings that involved the organizers and other collaborators. The version of Cobi that organizers used supported all session, paper, and session chair-related scheduling operations. It provided previews and recommendations for system-defined constraints, but as mentioned before, did not incorporate the authorsourcing data. For some of the meetings, Cobi was used in conjunction with a large wall display that visualized community data along with detailed session information and supported multiple users simultaneously exploring the schedule. The wall display did not include intelligence for resolving conflicts; the organizers relied on Cobi for conflict resolution and making actual changes to the schedule.

During the 42-day deployment, the three CHI 2013 organizers made 815 scheduling operations using Cobi's scheduling interface. We reconstructed the scheduling process by connecting organizers' description of the process with the interface usage log. Figure 7 shows the variety of subtasks that organizers faced during the scheduling process. By decomposing the scheduling problem into subtasks, organizers could focus on a particular aspect of the schedule at a given time.

The scheduling process proceeded in three high-level phases. In phase one (February 10 to February 17), organizers took the preliminary schedule from the technical program meeting and worked to resolve conflicts from violated system- and community-defined constraints. Organizers first moved papers so as to construct more coherent sessions based on the collected feedback from authors on which papers fit well in a session with theirs. Organizers then resolved all author conflicts by swapping sessions. While this eliminated nearly all of the existing system-defined conflicts, many sessions contained too few or too many papers. Organizers then moved papers to ensure that all sessions were at or under 80 minutes.

At the end of phase one, organizers had a mostly complete program. In phase two (February 17 to February 28), they worked to enhance themes and fine-tune the schedule to address special requirements. On the room level, they placed related sessions in the same or nearby rooms and sessions with awards in larger rooms. On the paper and session level, they distributed awards across sessions and reordered papers within a session so that they are presented in a logical progression. These subtasks generally did not involve conflict resolution, but the organizers used Cobi's conflict preview to ensure that changes would not introduce new conflicts. The organizers also made 122 session title edits to better capture and promote sessions and the papers within.

Once the organizers finalized the program, they worked in the final phase (March 1 to March 23) on assigning session chairs. Organizers first moved chairs out of sessions in which they had papers and corrected assignments where the chair was a poor fit. They then announced the initial assignments

Constraint Type	Related Entity	Data Source	Severity	Encoded	Initial	Final	Change
author with papers in opposing sessions	session (Figure 8)	system-generated	high	-	31	0	-31
topics of interest to a persona in opposing sessions	session (Figure 8)	system-generated	medium	-	6	4	-2
papers of mutual interests in opposing sessions	paper (Figure 9)	authorsourcing	high	923	40	19	-21
papers that do not fit well in the same session	paper (Figure 9)	authorsourcing	medium	651	129	42	-87
chair's paper in own session	chair (Figure 10)	system-generated	high	-	21	0	-21
chair's paper in opposing sessions	chair (Figure 10)	system-generated	high	-	6	0	-6
chair interested in opposing sessions	chair (Figure 10)	authorsourcing	medium	243	5	4	-1
chair in a session with a bad fit	chair (Figure 10)	chairs	medium	-	0	1	1
Total violated					**238**	**70**	**-168**

Preference Type		Data Source	Severity	Encoded	Initial	Final	Change
papers good in the same session	paper	authorsourcing	N/A	805	268	272	4
chair fits well in the session	chair	chairs	N/A	-	90	78	-12
Total satisfied					**358**	**350**	**-8**

Table 2. For all constraint or preference types in our deployment, the table shows the related entity, the data source for encoding, the severity level displayed in the tool, the total number of encoded items if authorsourced, the violation or satisfaction count from the preliminary schedule, the count in the finalized schedule, and the change in the count (highlighted if improved). There remains no conflicts for 3 of the 4 high severity types (highlighted). Soft constraints (medium severity) have more violations, which shows that scheduling involves multiple factors in addition to conflict resolution.

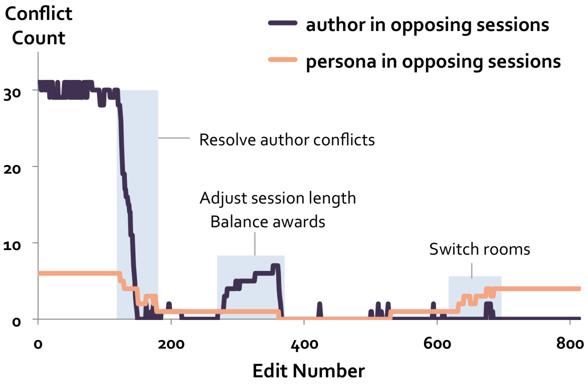


Figure 8. Change in system-defined conflicts over time. The organizers resolved 30 author conflicts during initial session making. When they were adjusting session lengths and balancing awards, the conflict count temporarily increased, but they resolved all of them shortly after. The organizers introduced three persona conflicts while they were switching rooms, but they chose not to resolve these conflicts due to other factors.

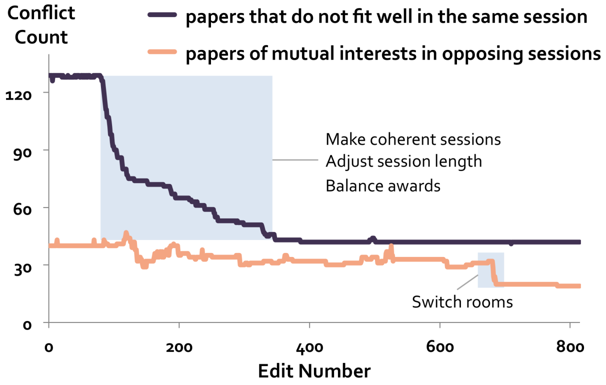


Figure 9. Change in community-defined conflicts over time. Organizers first attempted to make more coherent sessions with papers that did not fit well in the same session. After switching rooms, they resolved 11 conflicts to avoid having simultaneous sessions with similar topics.

to the session chairs, who provided additional feedback on fit and conflicts that the chairs resolved subsequently.

Conflict Resolution

The preliminary schedule from the technical program meeting included 238 conflicts. Organizers resolved 168 of them during the scheduling process. Table 2 summarizes changes in conflict counts for each constraint and preference type.

Schedule creation (phase one and two)

Organizers resolved all but four conflicts from system-defined constraints (Figure 8). Shortly after making more cohesive sessions (but in the same meeting), the organizers eliminated all 30 author conflicts in 29 minutes. This ensures that no presenter has papers in parallel sessions and that co-authors can attend all sessions that contain their papers. Persona conflicts were mostly absent because the in-person meeting to generate the initial schedule already took personas into account by scheduling sessions of interest to the same persona apart.

Organizers also resolved many of the community-defined conflicts (Figure 9). 21 of the 40 “papers of mutual interest in opposing sessions” conflicts in the initial schedule were resolved (53%), and 87 of the 129 “papers that do not fit well in the same session” conflicts were resolved (67%). Note that organizers made these changes using community-provided data and Cobi’s paper-level operations, but without the previews and recommendations.

Session chair assignment (phase three)

During phase three (Figure 10), organizers resolved all 27 conflicts based on system-defined chair constraints. 6 conflicts involved chairs with papers in opposing sessions and 21 conflicts involved chairs with papers in their own session.

Reflection

Discussions with organizers revealed a number of key points on how Cobi supports the conference scheduling process and helps to resolve conflicts:

Simplifying conflict resolution with preview and feedback

The organizers commented that Cobi “trivialized conflict resolution” and was “a major stress reducer.” Organizers noted

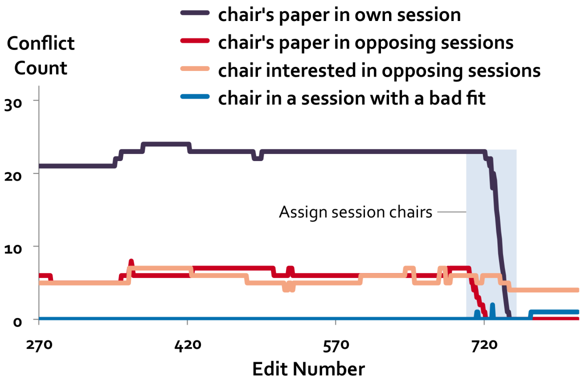


Figure 10. Change in session chair-related conflicts over time. Session chair assignment took place near the end of the scheduling process. The organizers resolved a majority of conflicts from the initial assignment by considering the individual constraints they collected from chairs.

that Cobi's visual previews aided in understanding the consequence of making a move, and immediate feedback accelerated conflict resolution. One organizer described the experience of resolving system-defined conflicts as follows: "It went really really fast, because we see a session that has conflicts, and the suggestions Cobi was giving were really good, and then we swapped things. It was almost a no brainer."

Enabling mixed-initiative problem solving

Cobi allowed organizers to use constraint-solving intelligence alongside their own knowledge of the schedule and of tacit constraints to improve the program. One organizer commented that "I was by and large driven by what Cobi was suggesting. As you make progress you can then progressively integrate other criteria that are not explicit in the system." In swapping sessions to resolve author conflicts, another organizer noted that he used Cobi's conflict previews to find good sessions to swap with that were in the same room, so as to resolve conflicts while maintaining the themes that were loosely assigned to rooms during the in-person meeting. Cobi also allowed organizers to be aware of causing potential conflicts even when they weren't working to resolve them. "We had Cobi up all the time to make sure that when we had a solution that we thought worked from the affinity point of view, that it didn't introduce new conflicts."

An important aspect of any mixed-initiative system is the user's trust in system recommendations. Organizers noted that their trust in Cobi grew over time. "For those of us who returned to the problem on multiple occasions, with a diverse set of short-term colleagues with varying expertise who came in to help, I think our appreciation of Cobi actually grew. We were looking at C&B [contribution & benefit] statements, as well as abstracts, and double-checking with our visitors, and Cobi kept coming up with great suggestions. So it stopped being based on following Cobi because it dealt with the areas that we knew, and became more fundamental, because it held up under scrutiny from a variety of visitors and when we delved into the details of various papers."

Intelligence powered by community input

Organizers commented on the value of having authorsourcing data during the scheduling process. In previous years, "authors saw their paper move from a slot they liked to a time they didn't and talked to the TP chairs." But this year, "we had virtually none of this. Authors were asked for input, most gave it, we tried hard to accommodate them, and almost nobody complained."

Organizers noted that authorsourcing data was particularly useful for understanding the affinity among papers beyond the initial sessions created at the TP meeting. "It helped with the more subtle issue of what happens when a paper moves out of a 'happy session'. Pairs of papers with a strong affinity are not in conflict if in the same session, and become conflicted if they move out of that session, but stay in the same time slot. This got lost over and over again in previous years and is the big win this year."

The organizers used a wall display to visualize authorsourcing data. Since the version of Cobi that organizers used at the time did not incorporate this data, organizers had to manually identify paper swaps that lead to more cohesive sessions. Later, when asked to test a version of Cobi's scheduling tool that encoded authorsourced preferences and constraints, organizers noted that having Cobi able to propose and resolve community-defined conflicts would have been valuable: "That is precisely what I was doing by hand with the authorsourcing data. And [the blue icon showing community preference] is absolutely useful." "I think if we had this earlier when we were doing the first round of improving the sessions, we would certainly have used it."

Weighing priorities and deliberately leaving conflicts

If Cobi makes it easy to resolve conflicts, why does the final schedule still include 70 conflicts? When asked this question, an organizer replied that for many of the remaining conflicts "we felt that they were minor or we decided that they weren't really conflicts." Since there are multiple factors that affect scheduling decisions, satisfying all constraints is sometimes not possible. An organizer commented that "it's also a question of opinion in some cases." In these cases the organizers made the final decision, which corresponds to Cobi's design goal to support decision-making while keeping the user in control at all times.

Mediating and visualizing community input

While the authorsourcing data provided a rich perspective on the community's preferences and constraints, the data was also sparse and noisy. In resolving authorsourced conflicts, organizers strived to understand the complexity of community input. Organizers used their wall display to visualize the raw community data, and attempted to account for the variance, quality, and weight of the data when making decisions. While organizers appreciated Cobi's aggregating authorsourcing data to remove noise and highlight salient conflicts, they also noted that visualizing the raw data was helpful and gave them more flexibility in using community input. Based on this feedback, we plan to develop other methods for mediating and visualizing community input, that can reflect its variance and quality while also maintaining simplicity.

Visualization improvements

Organizers noted a few areas for potential improvement in Cobi's visualization. A major issue raised was that Cobi can only display the details of one session at a time. While scheduling, organizers found that they often wanted to compare multiple entities. One organizer noted that "having to sequentially navigate multiple items worsens the experience a bit." To address these comments, we are currently exploring alternative visualizations for displaying detailed information for multiple sessions and are considering applying focus plus context techniques such as those found in TableLens [12]. The challenge is in providing additional context without distorting the schedule table in ways that hinder sensemaking.

Another solution to the visualization challenge is to leverage more space when available. For example, the wall display the organizers used for planning CHI 2013 was large enough to visualize detailed submission information in the global view. It allowed a group of people to collaborate on scheduling subtasks, although it lacked conflict resolution capability. A possible extension for Cobi is to directly connect to a large display so as to facilitate collaboration and enable the micro-outsourcing of scheduling tasks.

CONCLUSION AND FUTURE WORK

The Cobi system integrates community process, constraint-solving intelligence, and end-user interface to help organizers plan large conference schedules. Cobi's scheduling tool encodes community input as preferences and constraints, and helps organizers resolve conflicts by providing previews and recommendations when editing the schedule. A live deployment of Cobi for planning CHI 2013 demonstrated the effectiveness of collecting preferences and constraints from community members, and of the scheduling tool for simplifying conflict resolution and supporting informed decision-making.

The challenges of conference scheduling—understanding paper affinities, knowing what people want, and managing the solution complexity—are shared by conferences beyond CHI. We believe the approach presented in this paper generalizes to scheduling other academic conferences, and also to other events such as trade shows, film festivals, or university courses. More generally, the success of Cobi's deployment posits community-informed, mixed-initiative interaction as a novel approach for solving optimization problems, for which the goal is to collect important data from the community, use computation to guide the solution, and allow users to apply their tacit knowledge.

In future work, we plan to explore ways to further engage the community in the scheduling process. We wish to provide a generalized method for community members to express arbitrary constraints and preferences (e.g., travel plans and special considerations). We are currently working on an interactive interface that allows community members to specify a broader range of preferences and constraints. The collected input can then be encoded like other community-defined constraints, so that conflicts can be easily resolved using Cobi.

Another direction for future work is to extend the role of the community beyond providing data. Can hundreds of people collaboratively make sessions and resolve conflicts? We imagine that tools can support new ways of communicating, collaborating, and incorporating different opinions. Pursuing this research direction can lead to a new family of community-supported mixed-initiative systems that better mediates and visualizes diverse input from the community.

ACKNOWLEDGMENTS

We thank Patrick Baudisch, Kia Höök, Ed Chi, Pedro Lopes, Nirmal Patel, and members of the CHI community. This work is supported in part by Quanta Computer as part of the Qmulus project, by the Ford-MIT Alliance, and by NSF under awards SOCS-1111124 and SOCS-1208382. Juho Kim is partly supported by the Samsung Fellowship.

REFERENCES

1. Amershi, S., Lee, B., Kapoor, A., Mahajan, R., and Christian, B. Cuet: human-guided fast and accurate network alarm triage. In *Proc. CHI'11* (2011), 157–166.
2. André, P., Zhang, H., Kim, J., Chilton, L. B., Dow, S. P., and Miller, R. C. Community clustering: Leveraging an academic crowd to form coherent conference sessions. In *Proc. HCOMP 2013, in press* (2013).
3. Butler, K. A., Zhang, J., Esposito, C., Bahrami, A., Hebron, R., and Kieras, D. Work-centered design: a case study of a mixed-initiative scheduler. In *Proc. CHI'07* (2007), 747–756.
4. Evans, C., Abrams, E., Reitsma, R., Roux, K., Salmonsen, L., and Marra, P. P. The neighborhood nestwatch program: Participant outcomes of a citizen-science ecological research project. *Conservation Biology 19*, 3 (2005), 589–594.
5. Heimerl, K., Gawalt, B., Chen, K., Parikh, T., and Hartmann, B. Communitysourcing: engaging local crowds to perform expert work via physical kiosks. In *Proc. CHI'12* (2012), 1539–1548.
6. Horvitz, E. Principles of mixed-initiative user interfaces. In *Proc. CHI'99* (1999), 159–166.
7. Hsu, W.-L., Prietula, M. J., Thompson, G. L., and Ow, P. S. A mixed-initiative scheduling workbench: integrating ai, or and hci. *Decis. Support Syst. 9*, 3 (Apr. 1993), 245–257.
8. Jacob, R. J. K., Ishii, H., Pangaro, G., and Patten, J. A tangible interface for organizing information using a grid. In *Proc. CHI'02* (2002), 339–346.
9. Jones, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation 28*, 1 (1972), 11–21.
10. Kraut, R., and Resnick, P. *Building Successful Online Communities: Evidence-Based Social Design*. Mit Press, 2011.
11. Murray, K., Müller, T., and Rudová, H. Modeling and solution of a complex university course timetabling problem. In *Proceedings of the 6th international conference on Practice and theory of automated timetabling VI* (2007), 189–210.
12. Rao, R., and Card, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proc. CHI'94* (1994), 318–322.
13. Sampson, S. E. Practical implications of preference-based conference scheduling. *Production and Operations Management 13*, 3 (2004), 205–215.
14. Scott, S. D., Lesh, N., and Klau, G. W. Investigating human-computer optimization. In *Proc. CHI'02* (2002), 155–162.
15. Stumpf, S., Sullivan, E., Fitzhenry, E., Oberst, I., Wong, W.-K., and Burnett, M. Integrating rich user feedback into intelligent user interfaces. In *Proc. IUI'08* (2008), 50–59.
16. Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., Thiruvengadam, N. R., Huang, Y., and Steinfeld, A. Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design. In *Proc. CHI'11* (2011), 1677–1686.