

DiscoverySpace: Suggesting Actions in Complex Software

C. Ailie Fraser^{1,2}, Mira Dontcheva², Holger Winnemöller², Sheryl Ehrlich², Scott Klemmer¹

¹Design Lab, UC San Diego
La Jolla, CA 92093
{cafraser, srk}@ucsd.edu

²Creative Technologies Lab, Adobe Systems
San Francisco, CA 94103,
{mirad, hwinnemo, sehrlich}@adobe.com

ABSTRACT

Complex software offers power for experts, yet overwhelms new users. Novices often do not know how to execute tasks, what they want to achieve, or even what is possible. To address this, we introduce the DiscoverySpace interface for executable action suggestions. DiscoverySpace is a prototype extension panel for Adobe Photoshop that suggests task-level action macros to apply to photographs based on visual features. DiscoverySpace harvests these one-click actions from the online Photoshop user community. A between-subjects study indicated that action suggestions may help novices maintain confidence, accomplish tasks, and discover features. This work demonstrates how interfaces can leverage user-generated content to help novices navigate complex software.

Author Keywords

complex software; recommendations; macros; learning; photo editing.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION

Software tends to accrue features and complexity over time [1,20,23,24]. This bloat creates a steep learning curve in four ways: First, novice users often have a different vocabulary than the application [1], making it difficult to locate desired features. Second, while online tutorials abound, they can be hard to follow and users must interpret and adapt them to suit their situation and goals [14]. Third, software often provides several strategies for accomplishing similar tasks. Some are faster, easier, or more effective than others, and it can be difficult to identify which these are. Finally, users are typically only aware of a small percentage of software features [24]; there may be potential results they could achieve but do not think to try.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DIS 2016, June 04-08, 2016, Brisbane, QLD, Australia
© 2016 ACM. ISBN 978-1-4503-4031-1/16/06 \$15.00
DOI: <http://dx.doi.org/10.1145/2901790.2901849>

Accomplishing one's goal with software generally requires composing multiple commands into a sequence. The burden is on the user to know which tools to combine and how to apply them. This paper aims to reduce the execution gap between users' high-level task goals and the low-level steps needed to achieve them, by suggesting action macros. These aggregated action suggestions move the user's focus from individual *operations* to human-understandable, goal-driven *activities* [21].

We investigate the efficacy of action suggestions through DiscoverySpace, a recommendation interface for action macros recorded by the user community. Action macros encapsulate a sequence of operations that is executed as a batch. The DiscoverySpace prototype is a Photoshop panel comprising action macros that can be applied in one click for quick and easy exploration. We hypothesize that action suggestions help users get started in complex applications by enabling them to easily explore creative possibilities and achieve quick results.

This paper contributes a prototype action suggestion system, DiscoverySpace, and the results of a preliminary experiment to examine its efficacy. This between-subjects study found that action suggestions may help prevent novices from losing confidence in their abilities, and help users to accomplish tasks and discover new features. This paper also proposes design guidelines for a suggestion interface based on these observations and results.

RELATED WORK: HELPING NOVICES USE SOFTWARE

The literature offers several strategies for helping novices effectively use software: some focus on high-level tasks, others on lower-level tool use. Figure 1 describes how our approach relates to this prior work.

Interactive Tutorials Provide Step-by-step Guidance

Online tutorials are a popular resource for users of complex software. However, they present difficulties such as switching back and forth between the browser and the application, and mapping screenshots or videos of the application to the user's own version [14,29]. Interactive tutorials address these challenges by guiding users step-by-step through hands-on example tasks inside the target application [14,17,29]. Such tutorials can even be generated automatically as a user demonstrates them [10]. However, there are still far more user goals than authored tutorials, and automatically making static tutorials interactive remains a chal-

leng [8,18]. In addition, users must translate between their own goals and the available tutorials.

Tutorials teach users to use applications step-by-step. Other systems focus on speeding up holistic tasks rather than teaching individual steps. For example, TappCloud introduced “semi-automated” tutorials that can be applied in one step, like macros, but also allow for the user to interact with each step if desired [18]. However, TappCloud users do not interact directly with the application’s tools, but instead with visual previews of different parameter settings. DiscoverySpace also focuses on smoothing the execution process rather than helping users learn the application.

Adaptively Disclosing Interface Functionality

Adaptive interfaces initially provide a simplified interface and progressively disclose additional features either automatically [9,23,26], or by allowing users to move between predefined interface stages [6,19,23,30]. Adaptive interfaces are most successful when they are controllable, predictable, efficient, and promote feature discovery [24,31]. However, achieving all of these is a well-recognized challenge [7,23]. Interfaces that disclose or move features automatically based on user behaviour (e.g., adaptive toolbars in Microsoft Word [9]) can improve efficiency, but often feel unpredictable, uncontrollable, and distracting [9,26,31].

Photoshop Elements is an example of an interface with predefined stages: quick, guided and expert. Users can choose a mode appropriate for their task and skill level, such as quick mode for easily accomplishing basic edits. However, because each mode is associated with a different level of complexity, the interface layout and grouping of tools differs between the modes, making it difficult to transition from one to another.



Figure 1: Two dimensions along which interventions to help novices can vary. X-axis: the goal of the user (whether they want to get things done quickly, or learn the application and its tools). Y-axis: the approach to assisting the user (providing high-level tasks or low-level tools).

Adaptive interfaces have mainly focused on task completion efficiency as the performance metric [9,19]. However, for open-ended creative tasks, speed is often less important than discoverability and final quality. Progressive disclosure impedes discoverability by hiding interface elements [7]. DiscoverySpace addresses discoverability by suggesting actions users may not have known to be possible.

Command Suggestions and Previews

CommunityCommands introduced the idea that recommending commands to users can improve an application’s discoverability [20,22]. It uses collaborative filtering to suggest AutoCAD commands that are unknown to the current user but frequently used by others in tandem with the current user’s frequent commands. Though command recommendations improve discoverability of new features [20], suggesting tool-level commands still requires the user to compose and apply them. Inspired by this, DiscoverySpace uses suggestions to promote exploration and discovery, but in the form of task-level actions rather than tool-level commands.

Previewing results helps users predict what a command will do without having to execute it first. Side Views demonstrated that previews are especially valuable when shown as multiples with different parameter settings [32]. Building on this, suggesting and previewing alternative courses of action provides users with ideas they may not have considered, speeds the iterative design loop, and shows the contextual effects of changes. Inspired by DesignScape’s suggestion interface for graphic design layouts [25], DiscoverySpace provides both minor (refinement) and major (radical) suggestions.

Popular recommendation algorithms, like collaborative and content-based filtering [27], rely on users’ behaviour and preferences as input. For suggestions within a user interface, the user often begins with a document, 3D model, or image. Suggestions should be content-dependent, since different documents will benefit from different types of operations. For example, DesignScape generates layout suggestions by altering existing elements in the document [25]. Since visual editing operations can have vastly different outcomes on different images [2], it is important to suggest effects that make sense for the content in question. Our algorithm therefore takes into account the content of the image to produce relevant suggestions.

Natural Language Search

To help users find functionality, applications like CommandSpace integrate search directly into the interface [1]. Natural language search reduces the gap between user language and application language [1]. Inspired by this work, DiscoverySpace includes search functionality in addition to automatic suggestions. DiscoverySpace also provides simple faceted browsing, the ability to refine a collection based on metadata characteristics [15]. Facets are especially beneficial for “exploratory searchers” who tend to have only a partial idea of what they are looking for [12].

Faceted search and browsing were designed for information retrieval tasks; DiscoverySpace extends this underlying idea to executable *actions*.

WHAT KIND OF HELP DO NOVICES NEED?

To unearth novice-expert differences in creative software usage strategies, we conducted nine 30-minute sessions in which three expert photo editors helped seven novices edit their photos using Adobe Photoshop and Lightroom. Each session included one novice and one expert (Figure 2). The novice described their goals while the expert controlled the program and communicated with the novice to help achieve them. One-on-one tutoring like these pair discussions is highly effective for teaching [3], and consequently a valuable model for what software could achieve. We observed these interactions to see what novices asked, how experts translated these requests into image editing operations, and whether there were novice-expert language differences or communication challenges. Our intuition is that good software should enable novices to solve challenges like these for themselves. After all the sessions, we reviewed our notes and looked for recurring behaviour patterns. The following four insights were most prominent:

1. Novices often did not know what they wanted to do with a photo, and could not picture how it might be improved. In these cases, the experts would provide their own suggestions to get things started.
2. Novices had very high-level goals (e.g., “I want to make this person stand out more”), and often did not know what tools or techniques were needed to accomplish them. Experts were able to translate these goals into concrete operations.
3. A common way to show a novice what an effect will do was to execute it on the photo and compare the result with the original photo, sometimes exaggerating the effect to illustrate the difference before dialing it back down. Novices were sometimes hesitant about an expert’s suggestion, but after seeing its effect on the

image would become more excited about it.

4. Novices appreciated both suggestions that were relevant to their goals, *and* suggestions for different effects they would not have otherwise thought of.

DESIGN GOALS

Presenting suggestions relevant to the user’s goal can help them accomplish it, as the experts did. Presenting suggestions the user might not have otherwise thought of can help them to discover what is possible and achieve creative results. Combining these insights with prior recommendation research (e.g. [12,13,15,20]), we developed five main **design goals** for suggestion interfaces:

1. *Help users get started*: Make suggestions available as soon as the user begins a task. A frequent observation throughout our formative study and prototyping process was that users often did not know where to start in Photoshop.
2. *Use human language*: Allow users to search using goal terminology, and describe suggestions in a language novices can understand by including a descriptive non-technical title for each. This reflects the growing popularity of natural language or “semantic” search [13].
3. *Show previews* of what a suggestion does, and allow users to easily compare before and after, for example as in Side Views [32].
4. *Offer faceted browsing* in addition to search to help users explore. This is a well-documented concept in the information retrieval literature (e.g., [12,15,33]), and we believe it to be applicable to software applications as well.
5. *Suggestions should be relevant* to the user’s current task, but should also alert the user to new or unknown possibilities. CommunityCommands users were found to prefer “contextual” suggestions related to their short-term history [20], and users of complex software often desire the ability to discover new features [24].

We iterated on the DiscoverySpace design based on feedback from users with varying levels of Photoshop proficiency. Because DiscoverySpace harvested actions from multiple online sources, they had widely varying amounts of user interaction: some automatically applied effects, while others paused at key points with dialogs for user interaction. We removed such pauses and dialogs when present because they tended to lack sufficient accompanying explanation, and therefore confused some non-expert pilot testers. Users often desired basic adjustments such as brightness/contrast, exposure, and saturation. The actions in our collection generally comprised several steps rather than just one, and so did not include many of these basic edits. We manually created 11 actions for these basic adjustments, which would simply initialize the values to auto, and leave the sliders visible for the user to adjust as they wish.



Figure 2: A novice-expert session. The expert (right) compares the edited photo with the original to illustrate what the edits did.

DISCOVERYSPACE: ACTION SUGGESTIONS

This section describes the DiscoverySpace interface (Figure 3) and its keyword-based suggestion algorithm.

User Experience

When a user first opens a photo, DiscoverySpace prompts them to enter a goal and select features their image contains from a list (e.g. “contains people”, “outdoor”). Next, the DiscoverySpace home page appears, containing a search bar, category buttons, and suggested effects (Figure 3). Each suggestion is displayed by showing the effect applied to an example image with appropriate content. For example, the image for a “skin smoothing” effect is of a close-up face. The user can mouse over the image to compare *before* and *after*. Clicking a suggestion applies that action to the photo. As given, the user has no control over the settings for each step of the action; they can see the sequence of steps that was done after applying it by opening the History panel (Figure 4), but cannot step through it or adjust the parameters of individual operations. These actions are therefore intended to allow users to quickly achieve an effect without needing to manually complete all of the intermediate steps. The user can easily undo or redo the effect once it has been applied. If the user scrolls down on the main page, more suggestions appear. To browse suggested effects in a specific category, the user can click a category button. To search the corpus of effects, the user can enter a query in the search bar. The



Figure 3: DiscoverySpace as a panel in Photoshop. Users can apply a suggested action by clicking on its image.

user can also browse effects that are similar to a selected effect by clicking the “*Similar*” button on the effect.

Implementation

DiscoverySpace is a Photoshop extension panel, written in HTML, Javascript, and Adobe ExtendScript. It retrieves a manually-curated corpus of 115 actions stored on an Amazon S3 server, providing the flexibility to update the actions to reflect popular trends. We manually defined the action categories by reviewing and clustering free actions found online. We created the corpus by downloading 2-3 actions from each subcategory. Paid actions could be added in the future by allowing users the option to pay for effects from within the panel. DiscoverySpace automatically augments search queries with synonyms to maximize the number of relevant results.

Keyword-based Recommendations

DiscoverySpace recommendations take the user’s photo as input (Figure 5). Each action has been assigned a number of descriptive keywords. When opening a new photo, users must select features that describe the image; image analysis could automate this step in the future. We map the selected features to potentially relevant keywords, and assign each action a weight based on the number of matching keywords.

To select and present suggestions, our algorithm randomly samples from the collection of actions based on these weights: actions with a larger weight have a higher probability of being selected. The randomness encourages discovery by presenting new suggestions when the user refreshes the page. Each action has also been manually classified as either *refinement* or *radical*. An action is deemed *radical* if it produces a drastic effect, does not necessarily

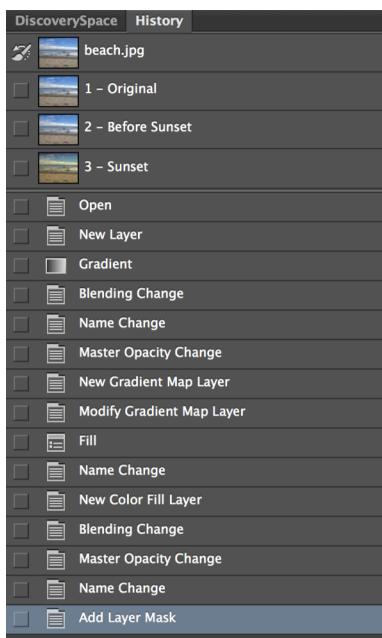


Figure 4: In the History panel, users can review the operations an action has performed.

maintain realism, and/or takes multiple steps to accomplish. Examples include painting effects, HDR, replicating old cameras, and drastic color effects. An action is deemed a *refinement* if it involves a small adjustment that is meant to improve the photo while maintaining realism, or could be accomplished in one step. Examples include brightening, softening skin, sharpening, and slight color casts. We prefer that both types of effects are suggested, to promote variety in the options. To ensure this, we run the prioritization algorithm separately on the two collections of actions, and then, for every four actions that are suggested, we sample two from the refinement pool and two from the radical pool. This is because on a 27-inch desktop display, DiscoverySpace in its default position shows four suggestions at a time.

STUDY

To investigate the efficacy of action suggestions, we conducted a between-subjects experiment in which participants edited their own photos in Photoshop. In the experimental

condition, subjects had access to DiscoverySpace; in the control condition they did not. We hypothesized that participants exposed to the suggestions in DiscoverySpace would find the editing process easier, feel more creative, and become more confident using Photoshop.

Method

To sign up, participants completed a brief initial survey about their Photoshop experience. Participants were then scheduled for a 30-minute session and assigned to one of two conditions: participants in the DiscoverySpace condition used Photoshop with the DiscoverySpace panel open, and participants in the Control condition used Photoshop with a blank panel that instructed them to save their photo when finished (Figure 7). The study held constant the length of the session (30 minutes), the number of photos edited (two), the questions participants answered when opening and closing a photo, and the availability of a web

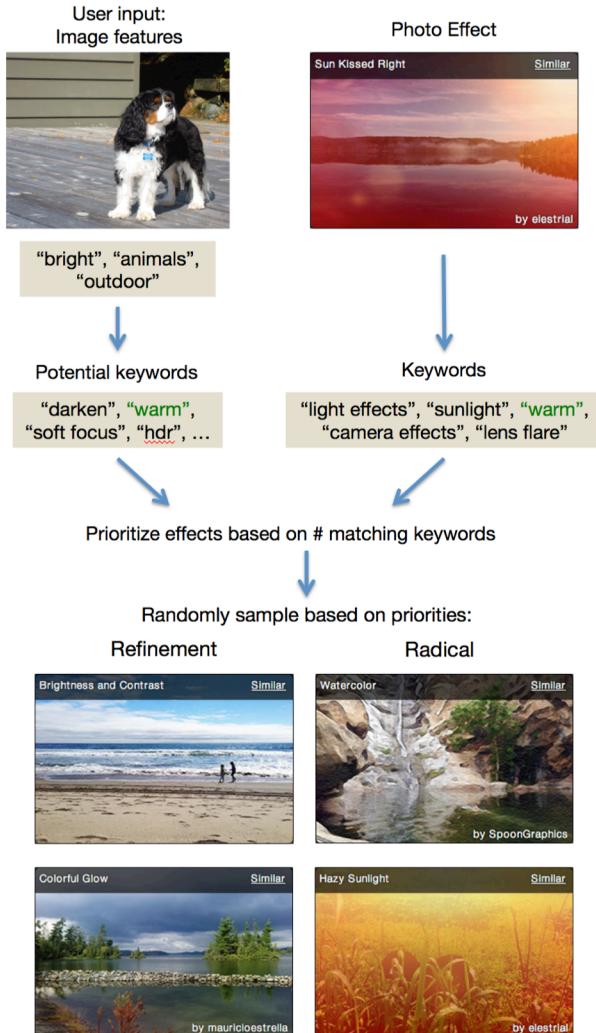


Figure 5: An overview of our keyword-based recommendation algorithm. User-selected image features are mapped to keywords and matched against the keywords for each action in the corpus.

browser (yes).

Each session comprised the following steps: consent form, background questions, edit first photo, edit second photo, short interview, and final online survey. The background questions allowed participants to elaborate on their initial survey responses regarding their prior experience with Photoshop and photo editing. Participants were instructed to bring one photo containing at least one person, and one photo without people, to ensure some consistency across participants' photos. The editing order of the two photos was randomly assigned with balancing in each condition. We chose to allow participants to bring their own photos, rather than provide sample photos, so that participants would be more likely to come up with their own goals for their photos, and would be motivated to do a good job.

In both conditions, upon opening a photo, the study panel prompted the user to describe their goal for the photo and select its image features. The panel also prompted the user to answer a few questions about their editing experience every time a photo was closed (Figure 6). The panel was in a prominent location and the participants were made aware of it by the experimenter. The task was open-ended: participants could define their goal however they wanted and were simply instructed to work until they were satisfied (or until time ran out). We used an open-ended task as opposed to a more directed one because participants would be more likely to perform as they would in a real-world setting when working on personally meaningful tasks. In the DiscoverySpace condition, participants were given a quick overview

The screenshot shows the DiscoverySpace software interface with a survey overlay. The survey includes questions such as 'Did you achieve your goal with the photo you just closed?' with options Yes, No, Partly, and Will come back to it later. It also includes rating scales for 'How easy or difficult was your editing experience?' (1=Very easy to 5=Very difficult), 'How creative do you think your edits were?' (1=Not creative to 5=Very creative), 'How confident are you that you did a good job editing this photo?' (1=Not confident to 5=Very confident), and 'Did you use any other resources to help you edit this photo (e.g. web searches, help menu, tutorials)?' with options Yes and No. A 'Submit' button is at the bottom.

Figure 6: Questions answered by participants after editing each photo. The measure for "How confident are you that you did a good job editing this photo?" is referred to as "confidence about performance" in the Results.

of the DiscoverySpace interface and were shown what each button in the panel did, so that time would not be wasted figuring out how to use the interface. The History panel was located behind DiscoverySpace, and was only visible when clicked on. Participants in the DiscoverySpace condition were shown the History panel only if they asked explicitly how to go back in time by more than one action.

In both conditions, a web browser was open on the Google home page just behind Photoshop; participants were told they could use it at any time. This was included to reflect the ready availability of web resources in creative workflows. A short interview and final survey took place in the last five minutes of the session. It comprised follow-up questions on the participant's overall editing experience and their perceived difficulties.

Participants

28 students were recruited from four undergraduate classes at a Southern California university (one photography class, three design classes). None of these classes provided formal training in Photoshop, though many participants had experience with other photo editing software and/or basic photography principles. Stratified randomization was used to balance gender and Photoshop expertise across conditions. Expertise with Photoshop was measured as follows: in the initial survey, participants were asked to rate their skill level with Photoshop from 1 to 5, and their amount of experience with Photoshop from 1 to 5. The answers to these two questions were added together to produce a score between 2 and 10, and participants were grouped according to their score into Beginner (2-4), Intermediate (5-7) and

	Female	Male	Total
Beginner	6	5	11
Intermediate	3	9	12
Expert	1	4	5
Total	10	18	28

Table 1: Participants' gender and Photoshop expertise.

Expert (8-10). See Table 1 for the distribution of participants' expertise and gender. 15 participants were assigned to the DiscoverySpace condition, and 13 to the Control condition (the uneven balance was due to the stratified randomization).

Measures

Both surveys asked participants to rate their Photoshop confidence by answering two questions: "How confident are you with your ability to edit photos in Photoshop? (1-5)" and "How confident are you that you can achieve your goals in Photoshop? (1-5)". We measured change in confidence as the difference between the sum of both responses (between 2 and 10) in the final survey and the initial survey. For each of the three Likert-scale questions answered after every photo (Figure 6), we added the responses from both photos to obtain one score between 2 and 10 for each participant ($N = 28$). For the questions with yes/no answers, we considered each photo separately ($N = 56$ as each participant edited two photos). We also recorded the number of Photoshop tools used (not including those from DiscoverySpace) and all clicks and pages loaded in DiscoverySpace.

Results

For the numerical measures, we used analysis of variance to analyze Condition, Expertise, and their interaction. For yes/no measures, we used logistic regression with the same variables.

Participant Goals

As expected with such an open-ended task, there was a large variety of participant goals. The most frequent goals focused on color/brightness (e.g., "correct exposure and color balance" or

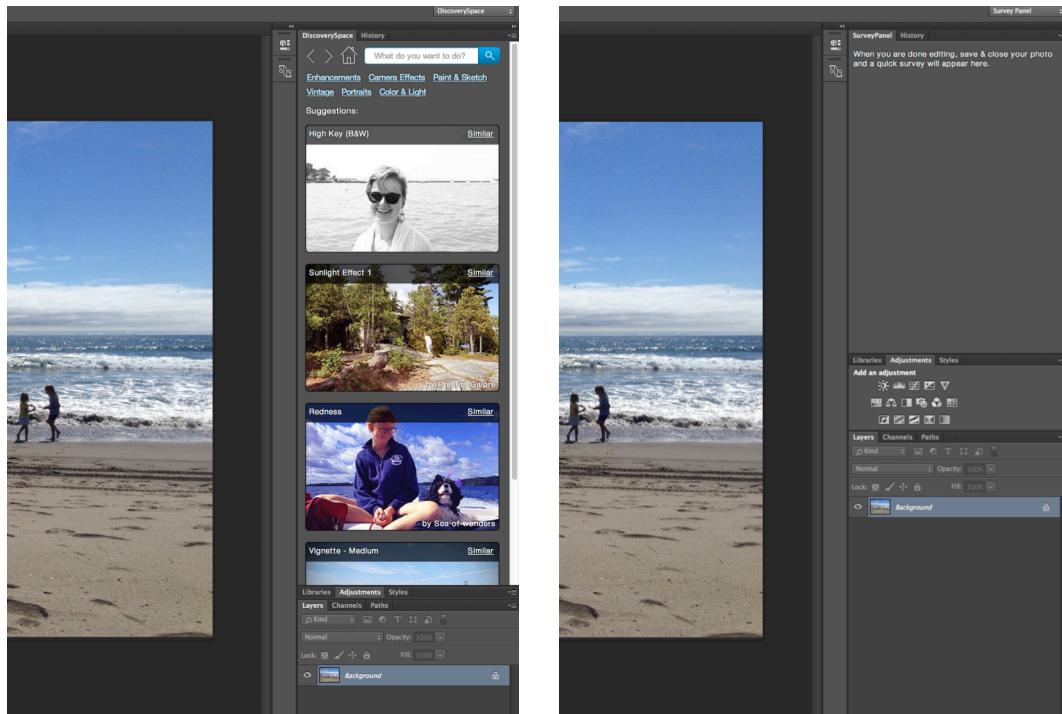


Figure 7: Photoshop as it appeared in the DiscoverySpace condition (left) and Control condition (right).

“experiment with color”), or a particular emotional effect (e.g., “make it surreal looking” or “give a lonely, solemn feel”). As expected, participants more often (72% of the time) used non-technical terms to describe their goals rather than photography-specific terms such as “saturation” and “sharpness”. This was most prominent among Beginners and Intermediates.

Quantitative Results

Beginner participants in particular were significantly more likely to lose confidence in the Control condition ($M = -2.0$) and gain confidence in the DiscoverySpace condition ($M = 0.5$), $t(22) = -2.16, p = 0.04$ (Figure 8). Across all Expertise levels, DiscoverySpace had only a marginal effect on participants’ confidence: the confidence of Control participants decreased ($M = -0.77$) while the confidence of DiscoverySpace participants increased ($M = 0.27$), but this was not significant, $F(1, 22) = 2.97, p = 0.10$. No effect was found for Expertise.

DiscoverySpace participants were significantly less likely to state that there was something they couldn’t figure out ($M = 50\%$) than Control participants ($M = 80\%$), $\chi^2(1) = 5.52, p = 0.02$.

All measures compared between conditions are summarized in Table 2. Responses to many of the post-editing questions varied widely. Within the Beginner group, DiscoverySpace participants rated difficulty as marginally but not signifi-

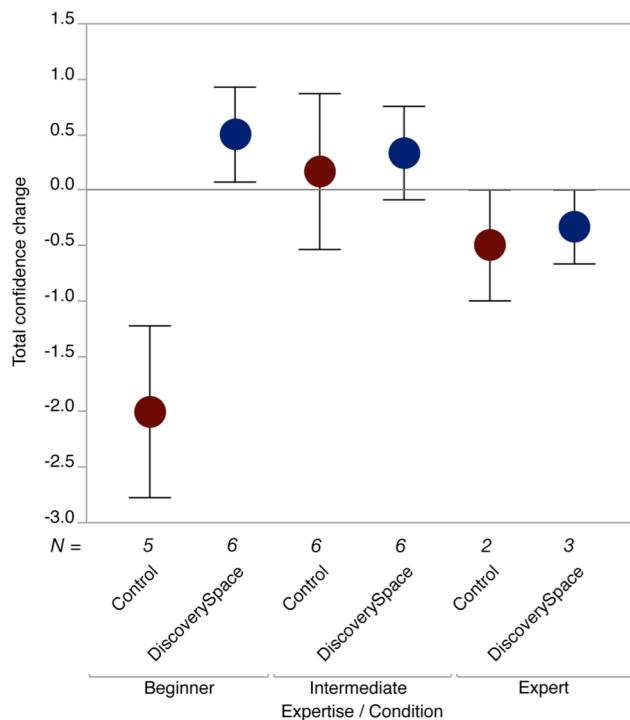


Figure 8: Confidence of Beginner Control participants decreased while confidence of Beginner DiscoverySpace participants increased. Error bars show 1 standard error from the mean. Note that due to small sample sizes in the sub-conditions, this is intended only as a descriptive illustration.

cantly lower ($M = 5.3$) than Control participants ($M = 6.8$), $t(22) = 1.62, p = 0.12$.

Across both conditions, Experts rated difficulty significantly lower ($M = 4.4$) when compared against Intermediates ($M = 5.8$) and Beginners ($M = 6.0$) together, $t(22) = -2.14, p = 0.04$. Experts also used significantly more tools ($M = 26$) than Intermediates ($M = 16$) and Beginners ($M = 16$), $t(22) = 2.37, p = 0.03$. In the DiscoverySpace condition, 51% of the actions used were radical and 49% were refinement, which suggests both types of actions were useful.

Overall, there was a strong negative correlation between difficulty and confidence about performance, $r(28) = -.70, p < .0001$. Interestingly, there was also a strong positive correlation between creativity and confidence about performance, $r(28) = .67, p < .0001$. This effect seems to hold most strongly for Beginners, as the correlation was weaker for Intermediates and Experts only, $r(17) = 0.48, p = 0.05$. Further exploration revealed that using the web was negatively correlated with achieving one’s goal, $\chi^2(1) = 8.09, p = .005$, and positively correlated with having trouble figuring something out, $\chi^2(1) = 6.93, p = 0.01$.

Preliminary analyses found no effects for type of class (photography vs. design), so this was excluded from further analyses. Gender was confounded with Expertise (Table 1) so Gender was also excluded from analyses; however within the Beginner group where Gender was roughly balanced, no Gender effects were found.

Qualitative Results

Participants were asked at the end of the session if they had discovered any new effects or tools. All DiscoverySpace participants replied yes, most of them adding that Discov-

	DiscoverySpace	Control	p
Total confidence change (-8 - 8)	0.27	-0.77	0.10
Couldn’t figure something out = yes	50%	80%	0.02
Achieved goal = yes	47%	31%	0.72
Used web browser	33%	54%	0.26
Difficulty (2-10)	5.3	6.0	0.37
Creativity (2-10)	4.5	4.1	0.86
Confidence about performance (2-10)	5.9	5.2	0.68
# Photoshop tools used	15	20	0.38

Table 2: Summary of mean values for measures compared between conditions. DiscoverySpace participants had marginally higher confidence change (this effect was significant within the Beginner group), and stated they could not figure something out less often than Control participants.

erySpace had exposed them to effects they had not seen before. 9 of 13 Control participants also responded yes, that they had discovered a new tool, mainly by clicking around or from an online tutorial. Two added that while they had discovered a new tool, they were unsure how to use it.

DiscoverySpace participants were asked two additional questions about the kinds of tasks for which they might use DiscoverySpace, and about the capabilities they wish it had. The most popular response to the task question was “for making quick and fun edits”, such as preparing a photo to post on social media or editing personal photos for fun, as opposed to aiming for professional-looking or detailed edits. The capabilities most participants wished DiscoverySpace had were more control over the result of an effect, such as sliders to dial it down or to adjust its parameters; and the ability to apply the effect to a selected part of the image only. Other requests included an explanation as to how the effect was achieved so that the user could easily repeat it and experiment with it, and better interaction with the document history. Though users could go back and forward through the applied operations using the History panel, it is separate from DiscoverySpace, and does not allow editing of the previous steps.

DiscoverySpace participants were also given the opportunity to share their general thoughts about the panel. Most stated either that they found it helpful, or that it had the potential to be helpful if some of the aforementioned improvements were made. Novice participants in particular seemed on the whole excited by DiscoverySpace as it meant they were able to achieve effects they otherwise would not have thought possible.

DISCUSSION

Despite the relatively small sample sizes when participants are broken up into sub-conditions, our results suggest that action suggestions can be beneficial for novice users of complex software. Further research should explore these possible effects in more detail.

It may not be that surprising that the confidence levels of participants in the Control condition dropped after using Photoshop. Novice participants were quickly frustrated with the time it took to find functionality and get accustomed to the interface, and more experienced participants struggled to locate forgotten tools and commands. We believe that DiscoverySpace, while it may not currently support all possible goals, provided a friendlier starting point that showed or reminded participants what types of edits are possible. Again, it is not that surprising that this effect was more pronounced for Beginners, as they are the most likely to misjudge their abilities, having never or rarely used Photoshop.

We assume that participants who answered “no” to the question, “Was there anything you wished you could do to this photo but couldn’t figure out how?” were able to accomplish the tasks they set out to do. Participants who

answered “yes” to the question were prompted to describe what they could not figure out. Most responses named a task the participant had been trying to accomplish (e.g., removing redness from a person’s face), but either gave up on or ran out of time. It is likely that DiscoverySpace provided quicker ways to accomplish many of these tasks, and thus participants in the DiscoverySpace condition were more likely to complete them in the time allotted than those in the Control condition.

As the study involved using the full Photoshop interface, expertise had a great impact on user performance. Observations during the sessions confirmed that Expert users were comfortable with Photoshop’s interface, whereas Beginners tended to find it overwhelming because of the large number of menus and buttons. Ratings of creativity and confidence about performance varied widely, likely because the task was open-ended, and these measures depended on each individual’s goal.

The correlation between confidence in performance and creativity indicates that participants in both conditions did tend to include creativity as a criterion for doing a good job, supporting the motivation for an interface that encourages creative exploration. Participants who used the web achieved their goals less often, likely because these are users who had more trouble with their tasks and turned to the web for help. However, using the web did not in fact seem to help them. The following section provides some suggestions as to why that may be.

Common Difficulties

The same researcher conducted all 28 sessions, and observed participants’ recurring errors and difficulties. The most prominent observations, outlined here, support our motivation for action suggestions, and provide directions for future work.

1. Failing to find the best way to accomplish a task

Several participants used Google to search for help and often followed the top result without realizing that there was an easier, more efficient, or more effective way to accomplish the task. For example, the automatically generated Google summary listed at the top for “how to crop photoshop” suggests using the Rectangular Marquee tool, rather than the more relevant Crop tool. In addition, many tutorials included screenshots from older versions of Photoshop, making it difficult for participants to find the tools mentioned. Version-specific help is a challenge for all software. Systems for augmenting search queries with the user’s context could improve this (e.g. [4]). For example, the top search result shown when “cc 2015” is appended to the above Google search is recent and demonstrates the Crop tool. Such functionality could be helpful for DiscoverySpace’s search feature when a larger corpus of actions or tutorials is added.

2. Not noticing when commands cause state changes or side effects

A common issue participants ran into was creating new layers without realizing, and then having the wrong layer selected when trying to apply another effect. For example, performing an adjustment from the Adjustments panel creates a new Adjustment layer, but does not explicitly notify the user of this. Most actions in DiscoverySpace also create at least one layer. In this study, many participants in both conditions struggled with having the wrong layer selected when trying to apply other edits, which meant those edits did not work as expected. This is an instance of a more general problem in complex software: commands often have side effects that the user is not aware of, which may cause confusion later on. This suggests that programs should 1) make the user aware of what they are doing, and 2) encourage better development of a correct mental model of the program, so that such side effects are not so surprising when they occur. Support for better interactive history may be an important improvement for DiscoverySpace as it would allow participants to work through and explore the steps that have been completed.

3. Difficulty finding out what tools do

For novices, complex software has poor information scent [28]: we observed users sequentially browsing menus, toolbars and panels to find something specific or see what is possible. Even with exhaustive search, Beginners seemed to have difficulty understanding what the buttons and menu items did, given only their names and brief tooltip descriptions. When participants did find the desired tool, they often had trouble figuring out how to use it. Better in-app descriptions and assistance are needed. For example, ToolClips augments tool-tips to include media content and detailed information [11]. This is an advanced form of “speaking navigation”: verbose navigational links that improve information scent. While the current goal of DiscoverySpace was to move away from individual tools toward better task-level support, one way to improve information scent for tools could be to display example tasks that a tool is often used for when mousing over it.

Study Limitations

Having an open-ended task facilitated more realistic usage than a pre-defined task, and having participants work on their own personal photos likely increased their motivation to do a good job. However, participants’ widely varying goals resulted in widely varying results regarding their editing experience and likelihood of achieving their goals. Future studies might consider more directed tasks to reduce the variability in participants’ goals.

Using self-reported expertise ratings has the potential for inaccurate assignments based on differences in how participants perceive their own skill. Our observations found their self-report sufficiently accurate. However, future studies might consider a more objective expertise measure, such as a pre-test. The pool of “novices” is larger than one might think: many Photoshop novices were experienced with

photography principles and/or simpler photo editing software. They had domain knowledge, just not expertise on the specific task at hand. Especially with feature-rich software like Photoshop, nearly everyone is a novice in some regard.

Participants’ widely varying expertise levels yielded small sample sizes for measuring interaction effects. However, for this preliminary study, this variety unearthed valuable differences in participant behaviour.

Finally, many participants ran out of time and stated this as the main reason they could not achieve their goal; future studies should allow participants to work for longer. A longitudinal study would allow for more realistic usage data, as participants could use the application on their own time and more goals could be collected per person.

FUTURE WORK

Participants’ experiences with DiscoverySpace suggest several ways to improve the robustness and usability of action suggestions in complex software.

Improving DiscoverySpace for Photo Editing

An important limitation of the current DiscoverySpace prototype is that most actions in its corpus produce *global* edits, not *local* ones. A few actions intended for faces (e.g. skin smoothing) allow the user to brush on the effect, but the majority apply to the entire photo. This is representative of the selection of free Photoshop actions available online; most are global edits because these are much easier to create than ones that allow for user input such as selection. However, users often desire local editing capabilities, and so providing actions that allow for this would be valuable. One way of providing support for user input would be to leverage the interactive steps provided by TappCloud [18].

Based on participants’ responses to the question of what tasks they would use DiscoverySpace for, it seems to be effective for quick exploration. In order to help users make more professional-looking edits, DiscoverySpace could take better advantage of the properties of the user’s input photo and apply effects in a content-adaptive way [2] or suggest automatic fixes based on properties of the photo such as its histogram (e.g. as in [5]).

Other immediate improvements to DiscoverySpace for photo editing will include automatic image analysis to replace manual user descriptions of photos, and replacing the default preview images of effects with previews of the user’s actual photo.

General Improvements to Action Suggestions

Many programs have the capability to record and save actions (e.g., Adobe Photoshop, Illustrator, and Acrobat), also referred to as presets (e.g., Adobe Lightroom), action macros (e.g., Autodesk’s AutoCAD), and macros (e.g., Microsoft Excel and Word). Even more programs allow for task automation via scripting (e.g., Adobe InDesign, Mac OS). DiscoverySpace and the following proposed im-

provements can apply not only to Photoshop but to any complex software that provides some way of combining operations into actions.

End-user Control

The most frequent piece of feedback received throughout the study and prototyping process was the desire for more control over the effects. One of the main advantages complex programs have over simple ones (e.g., Instagram) is the ability to have fine-grained control over operations, and so providing this control to users would be a valuable component of an interface like DiscoverySpace. This could be accomplished by examining current usage data of the software in question to determine what parameters users most often edit for common operation sequences, and making only those parameters available when an action is executed. Alternatively, DiscoverySpace could make the action history more interactive by allowing users to select, edit, and delete any step that was performed. To provide an intuitive way of setting parameters, visual previews like the variations in Side Views [32] and TappCloud [18] could be incorporated.

Recommendation Algorithm

Given the diverse goals that users have in complex applications, a technique like collaborative filtering could improve recommendations by personalizing the suggestions to the user and their goals. Li et al. recommend an “item-based” collaborative filtering algorithm based on their work with CommunityCommands [20]. This algorithm bases recommendations on the similarity between commands, rather than between users. They also found that users preferred recommendations based on their behaviour within the current session, rather than long-term. The goals users enter into DiscoverySpace should also be included as input to the suggestion algorithm, so that suggestions are not only relevant to the user’s photo, but also to the goal they have in mind. This natural language description could also be used to annotate the actions taken by users. With this extra information, we could build a better model of the relationships between goals, the language used to express them, and the tools or actions used. Such descriptions would otherwise be difficult to collate, as users would be required to do extra work to describe the actions they perform.

Building a Scalable Corpus of Actions

User-generated actions are posted across multiple sites and carry differing amounts of metadata and detail, and differing distribution licenses. Curating our corpus of actions required careful attention to these differences, and we had to manually define keywords and names for most actions. Scaling up such a corpus or building it automatically would require accounting for these issues. This could be done by creating a centralized repository inside the software itself or as part of DiscoverySpace, wherein users can create and share actions, and are required to provide keywords and agree to a distribution license. An existing example of this is AdaptableGIMP, an interface for the photo editor GIMP that provides user-created actions referred to as “task sets”

[16]. Users can create and share task sets from within the interface, and search the corpus for ones to apply.

Action suggestions need not be user-created. They could instead be generated by automatically recording and segmenting user behaviour along with the type of document being worked on, or by mining and analyzing the large collection of online tutorials that are available for most applications. Work has been done on generating interactive tutorials from static online ones [8,18], and incorporating such work into a search and suggestion interface like DiscoverySpace is a promising direction for improving both the initial usability and long-term learnability of complex software applications.

CONCLUSION

This paper introduced DiscoverySpace, a task-level action suggestion interface, and found that easy-to-execute task-level action suggestions may help prevent novices from losing confidence with complex software. DiscoverySpace allows users to more easily discover what is possible in an application by abstracting away individual commands in favour of understandable and goal-driven actions. By suggesting actions relevant to the user’s input document, DiscoverySpace can provide useful assistance toward accomplishing the user’s task. The experiment described in this paper confirmed the need for interventions like this in modern software, and provided results suggestive of the fact that action suggestions are beneficial to users.

ACKNOWLEDGMENTS

We thank Nancy Reid & Catherine Hicks for their help with statistical analyses. This work was supported in part by a Powell Fellowship from UC San Diego.

REFERENCES

1. Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: Modeling the Relationships between Tasks, Descriptions and Features. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST ’14*, ACM Press, 167–176. <http://doi.org/10.1145/2642918.2647395>
2. Floraine Berthouzoz, Wilmot Li, Mira Dontcheva, and Maneesh Agrawala. 2011. A Framework for Content-Adaptive Photo Manipulation Macros: Application to Face, Landscape, and Global Manipulations. *ACM Transactions on Graphics* 30, 5: 1–14. <http://doi.org/10.1145/2019627.2019639>
3. Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13, 6: 4–16. <http://doi.org/10.3102/0013189X013006004>
4. Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-centric programming: Integrating Web Search into the Development Environment. *Proceedings of the 28th*

- international conference on Human factors in computing systems - CHI '10*, ACM Press, 513. <http://doi.org/10.1145/1753326.1753402>
5. Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Fredo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. *CVPR 2011*, IEEE, 97–104. <http://doi.org/10.1109/CVPR.2011.5995413>
 6. John M. Carroll and Caroline Carrithers. 1984. Training wheels in a user interface. *Communications of the ACM* 27, 8: 800–806. <http://doi.org/10.1145/358198.358218>
 7. Leah Findlater and Joanna McGrenere. 2010. Beyond performance: Feature awareness in personalized interfaces. *International Journal of Human-Computer Studies* 68, 3: 121–137. <http://doi.org/10.1016/j.ijhcs.2009.10.002>
 8. Adam Fourney and Michael Terry. 2014. Mining Online Software Tutorials: Challenges and Open Problems. *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14*, ACM Press, 653–664. <http://doi.org/10.1145/2559206.2578862>
 9. Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the design space for adaptive graphical user interfaces. *Proceedings of the working conference on Advanced visual interfaces - AVI '06*, ACM Press, 201. <http://doi.org/10.1145/1133265.1133306>
 10. Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. *ACM SIGGRAPH 2009 papers on - SIGGRAPH '09*, ACM Press, 1. <http://doi.org/10.1145/1576246.1531372>
 11. Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, ACM Press, 1515. <http://doi.org/10.1145/1753326.1753552>
 12. Marti A. Hearst. 2006. Design recommendations for hierarchical faceted search interfaces. *Proc. SIGIR 2006, Workshop on Faceted Search*, 26 – 30.
 13. Marti A. Hearst. 2009. Chapter 12: Emerging Trends in Search Interfaces. In *Search User Interfaces*. Cambridge University Press, Cambridge, 297–324. <http://doi.org/10.1017/CBO9781139644082>
 14. Caitlin Kelleher and Randy Pausch. 2005. Stencils-Based Tutorials: Design and Evaluation. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, ACM Press, 541. <http://doi.org/10.1145/1054972.1055047>
 15. Jonathan Koren, Yi Zhang, and Xue Liu. 2008. Personalized interactive faceted search. *Proceeding of the 17th international conference on World Wide Web - WWW '08*, ACM Press, 477. <http://doi.org/10.1145/1367497.1367562>
 16. Benjamin Lafreniere, Andrea Bunt, Matthew Lount, Filip Krynicki, and Michael A. Terry. 2011. AdaptableGIMP: Designing a Socially-Adaptable Interface. *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology - UIST '11 Adjunct*, ACM Press, 89. <http://doi.org/10.1145/2046396.2046437>
 17. Benjamin Lafreniere, Andrea Bunt, and Michael Terry. 2014. Task-centric interfaces for feature-rich software. *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures the Future of Design - OzCHI '14*, ACM Press, 49–58. <http://doi.org/10.1145/2686612.2686620>
 18. Gierad Laput, Eytan Adar, Mira Dontcheva, and Wilmot Li. 2012. Tutorial-based interfaces for cloud-enabled applications. *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, ACM Press, 113. <http://doi.org/10.1145/2380116.2380132>
 19. Rock Leung, Leah Findlater, Joanna McGrenere, Peter Graf, and Justine Yang. 2010. Multi-Layered Interfaces to Improve Older Adults' Initial Learnability of Mobile Applications. *ACM Transactions on Accessible Computing* 3, 1: 1–30. <http://doi.org/10.1145/1838562.1838563>
 20. Wei Li, Justin Matejka, Tovi Grossman, Joseph A. Konstan, and George Fitzmaurice. 2011. Design and evaluation of a command recommendation system for software applications. *ACM Transactions on Computer-Human Interaction* 18, 2: 1–35. <http://doi.org/10.1145/1970378.1970380>
 21. Yang Li and James A. Landay. 2008. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, ACM Press, 1303. <http://doi.org/10.1145/1357054.1357259>
 22. Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. 2009. CommunityCommands: Command Recommendations for Software Applications. *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, ACM Press, 193. <http://doi.org/10.1145/1622176.1622214>

23. Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. 2002. An evaluation of a multiple interface design solution for bloated software. *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*, ACM Press, 164. <http://doi.org/10.1145/503376.503406>
24. Joanna McGrenere and Gale Moore. 2000. Are We All in the Same “Bloat”? *Proceedings of the Graphics Interface 2000 Conference, May 15-17*, 187–196. Retrieved from <http://graphicsinterface.org/wp-content/uploads/gi2000-25.pdf>
25. Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with Interactive Layout Suggestions. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, ACM Press, 1221–1224. <http://doi.org/10.1145/2702123.2702149>
26. Tim F. Paymans, Jasper Lindenberg, and Mark Neerincx. 2004. Usability Trade-offs for Adaptive User Interfaces: Ease of Use and Learnability. *Proceedings of the 9th international conference on Intelligent user interface - IUI '04*, ACM Press, 301. <http://doi.org/10.1145/964442.964512>
27. Michael J. Pazzani and Daniel Billsus. 2007. *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer Berlin Heidelberg, Berlin, Heidelberg. <http://doi.org/10.1007/978-3-540-72079-9>
28. Peter Pirolli. 2003. Chapter 7: Exploring and Finding Information. In *HCI Models, Theories, and Frameworks*, John M. Carroll (ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, 157–191.
29. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, et al. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, ACM Press, 135. <http://doi.org/10.1145/2047196.2047213>
30. Ben Shneiderman. 2000. Universal usability. *Communications of the ACM* 43, 5: 84–91. <http://doi.org/10.1145/332833.332843>
31. Ben Shneiderman. 2002. Promoting universal usability with multi-layer interface design. *ACM SIGCAPH Computers and the Physically Handicapped*, 73-74: 1–8. <http://doi.org/10.1145/960201.957206>
32. Michael Terry and Elizabeth D. Mynatt. 2002. Side Views: Persistent, On-Demand Previews for Open-Ended Tasks. *Proceedings of the 15th annual ACM symposium on User interface software and technology - UIST '02*, ACM Press, 71. <http://doi.org/10.1145/571985.571996>
33. Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted metadata for image search and browsing. *Proceedings of the conference on Human factors in computing systems - CHI '03*, ACM Press, 401. <http://doi.org/10.1145/642611.642681>