



Figure 1: DiscoverySpace is an extension panel for Adobe Photoshop. Our prototype suggests high-level macros created by the user community for photo effects, to help the user quickly improve their photo and explore the creative space.

DiscoverySpace: Crowdsourced Suggestions Onboard Novices in Complex Software

C. Ailie Fraser

The Design Lab
UC San Diego
La Jolla, CA 92092, USA
cafraiser@cs.ucsd.edu

Mira Dontcheva

Adobe Research
San Francisco, CA 94103, USA
mirad@adobe.com

Holger Winnemöller

Adobe Research
Seattle, WA 98103, USA
hwinnemo@adobe.com

Scott Klemmer

The Design Lab
UC San Diego
La Jolla, CA 92092, USA
srk@ucsd.edu

Abstract

Complex software, while powerful for experts, can overwhelm new users. Novices often do not know how to execute tasks, what they want to achieve, or even what is possible. We aim to address these problems by leveraging the large expert user base such programs tend to have. We present DiscoverySpace, a prototype extension panel for Adobe Photoshop that suggests macros for effects to apply to the user's photo, based on features of the photo. These macros are Photoshop actions that have been created and shared online by the user community, and can be applied in one click. Our work demonstrates how high-level macro suggestions can help users get started in a complex application. Preliminary feedback from a pilot study indicates that these suggestions may be most useful for users with exploratory goals, such as "make this photo more fun", rather than users with very specific goals.

Author Keywords

complex software; novice onboarding; recommendations; crowdsourcing; photo editing.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI):
Miscellaneous.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).

CSCW '16 Companion, February 27 - March 02, 2016, San Francisco, CA, USA

ACM 978-1-4503-3950-6/16/02.

<http://dx.doi.org/10.1145/2818052.2874317>

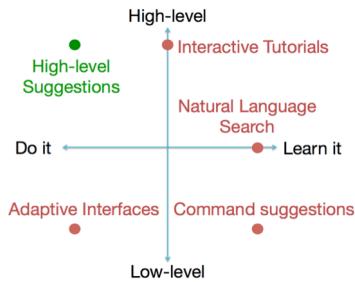


Figure 2: Two dimensions along which interventions to help novices can vary.
 x-axis: the goal of the user (whether they want to get things done quickly, or learn the application and its tools).
 y-axis: the approach to assisting the user (providing high-level tasks or low-level tools).

Our work provides high-level suggestions for getting things done, an area that has not been fully explored in previous work on complex software applications.

Helping Novices Use Software Applications

Software tends to accrue features and complexity over time [1, 4]. This creates a steep learning curve for several reasons: First, novice users often have a different vocabulary than the application [1], making it difficult to locate desired features. Second, while there is an abundance of online tutorials, they may not always directly map to the user's task or software version. Third, there are often several ways to accomplish similar tasks, making it difficult to identify the best way for a given situation. Finally, users may not even know what they *can* do let alone what help to search for.

Natural language search can help users locate functionality by mapping user language to program language [1]. Another strategy to onboard novices is to initially provide a simplified interface and progressively disclose additional features. Progressive disclosure can happen automatically, or by allowing users to move between predefined interface stages [5]. For example, Photoshop Elements provides three interface modes: quick, guided and expert. However, these modes look completely different and the transition between them is abrupt, so the first two modes do not prepare the user for the full expert mode. Another challenge with adaptively omitting and showing interface elements is that it is difficult for users to know what is available [2].

Interactive tutorials guide users through high-level tasks. While recent work has improved authoring tools [3], there are still far more user goals than authored tutorials, so users must often adapt the given instructions to their own situation, a complication that can reduce their motivation.

There is promise in leveraging the crowd's usage, as shown by CommunityCommands, a system that recommends new commands to AutoCAD users [4]. Their recommendation

algorithm uses collaborative filtering to suggest commands that are unknown by the current user but used by other similar users. However, suggesting low-level commands still requires the user to compose and apply them in order to achieve their goal, and thus provides little help toward achieving quick results.

We introduce DiscoverySpace, which integrates macros (recorded actions) created by the user community directly into complex software. We predict that this will help users to quickly achieve results they are happy with, and discover a wider range of creative possibilities, thus increasing motivation to continue engaging with the application. Our demo shows DiscoverySpace integrated into Adobe Photoshop (Figure 1). It suggests potential photo edit macros that have been gathered from the web.

DiscoverySpace: High-level Suggestions

DiscoverySpace provides high-level macro suggestions for helping users quickly accomplish tasks, an idea not fully explored in existing research on complex applications (Figure 2). These macros (called Photoshop actions) have been authored by other users and shared online, and can be applied in one click. We hypothesize that high-level suggestions will increase confidence and motivation in new users by allowing them to achieve results early on.

Design Guidelines

To understand what kinds of help maximally benefit novices, we convened and observed nine 30-minute sessions in which experts helped novices edit their photos. We gathered the following key insights from these sessions:

1. Novices often did not know what they wanted to do with a photo. In these cases, the experts would provide their own suggestions to get things started.

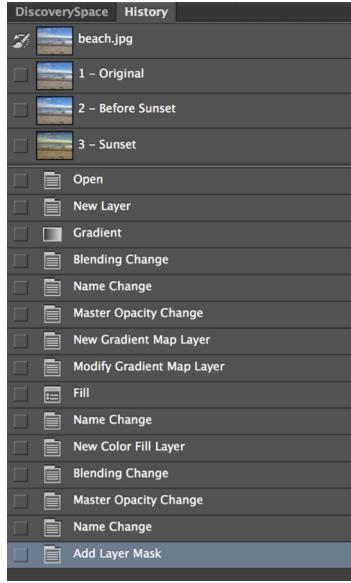


Figure 3: Top: Users can see the steps an action has done in the History panel. More than half the actions in our corpus consist of at least 10 steps, and thus would take much longer to execute manually. Bottom: Once applied, an effect can easily be undone and redone to compare before and after.

2. Novices had very high-level goals, 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.

Based on these findings as well as existing research on search interfaces and recommendation systems [e.g. 4, 6], we developed five design guidelines for suggestion interfaces:

1. **Help users get started:** Make high-level suggestions available as soon as the user begins a task.
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.
3. **Show previews** of what a suggestion does, and allow users to easily compare before and after.
4. **Allow faceted browsing** in addition to search to help users explore [6].
5. **Provide relevant suggestions** to the user's current task [4].

Implementation

DiscoverySpace is an HTML extension for Adobe Photoshop. It retrieves our manually curated corpus of 115 actions from an Amazon S3 server where they are stored, providing the flexibility to update the collection of actions to reflect popular trends. Each suggestion is displayed as a preview of the effect applied to an example image. Users can mouse over the image to compare before and after. Clicking a suggestion applies the action to the photo. Users can see the steps that were executed in the History panel

(Figure 3). Aside from browsing suggestions, users can also explore by category or search the corpus. Search queries are augmented with synonyms to maximize the number of relevant results.

Photo-driven Recommendations

DiscoverySpace recommendations take the user's photo as input, and the algorithm works as follows: Each action is assigned a number of descriptive keywords. When the user opens a new photo, they must select features that describe the image (e.g. "outdoor", "contains faces"). The selected features are then mapped to potentially relevant keywords, and actions are prioritized by the number of matching keywords. Since this algorithm does not rely on other users, it could be improved by using a technique like collaborative filtering. We intend to explore these possibilities after gathering data from our own system on how users are using the suggestions.

Early Findings: Best for Exploration

We conducted a pilot study ($N = 11$) in which a participant edited his or her own photos for 30 minutes, using a think-aloud protocol. In the first five sessions, participants were exposed to DiscoverySpace for the entire session. To enable comparison in the following six sessions, participants were not exposed to DiscoverySpace until they had finished editing their first photo. Participant skill level ranged from beginners who had never used Photoshop to expert Photoshop users.

For each photo, in a pre-survey participants were asked to state their goal, and in a post-survey they were asked to state whether they felt they had accomplished it. Suggestions were generally more useful for open-ended exploratory goals than for concrete tasks. One reason is that suggestions let users quickly try multiple alternatives. By

contrast, a specific task may not have a matching suggestion in the prototype's corpus. This may be because either our corpus is not large enough or the task requires additional user input (such as selection) and thus cannot be completed by a one-click action. Supporting more specific tasks would require a larger corpus of macros, as well as more ways for users to provide input to them. Many participants stated that they felt their edits were more creative when using DiscoverySpace, though they would have liked to have more control over the effects.

Challenges and Future Directions

User-generated macros are posted across multiple sites and have differing amounts of metadata and detail, varying behaviours when executed, and differing distribution licenses. Scaling up the corpus of suggestions automatically would require accounting for these issues. This could be done by creating a centralized repository inside the software itself, and by inferring metadata to fill in gaps or crowdsourcing this process.

Our future work is focused on richer automatic understanding and better user control. First, we intend to use automatic image analysis to replace manual user descriptions of their photos. Second, we intend to replace the preview images of effects with previews of the user's current image. Third, DiscoverySpace currently asks users to describe their goal in natural language. Future work would interpret this goal and include it as input to the suggestion algorithm. Finally, we would like to improve DiscoverySpace's ability to expose parameters and sliders for end-user control. This can be done by examining current usage data to determine what parameters are most often edited, and allowing more interaction with macros than a single click. More broadly, the DiscoverySpace concept applies to any application that allows its users to record and

save macro operations. The lessons learned here can benefit any such domain, not only in helping novices benefit from expert users, but in allowing users of all skill levels to learn and benefit from each other.

Acknowledgements

We thank Sheryl Ehrlich, Erik Natzke, Michael Rubin and Robert Bailey for their valuable feedback.

References

1. Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: modeling the relationships between tasks, descriptions and features. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). ACM, New York, NY, USA, 167-176.
2. Leah Findlater and Joanna McGrenere. 2010. Beyond performance: Feature awareness in personalized interfaces. *Int. J. Hum.-Comput. Stud.* 68, 3 (March 2010), 121-137.
3. Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. In *ACM SIGGRAPH 2009 papers* (SIGGRAPH '09). ACM, New York, NY, USA, Article 66, 9 pages.
4. 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 Trans. Comput.-Hum. Interact.* 18, 2, Article 6 (July 2011), 35 pages.
5. Ben Shneiderman. 2000. Universal usability. *Commun. ACM* 43, 5 (May 2000), 84-91.
6. Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03). ACM, New York, NY, USA, 401-408.