**Cleanser: An Adaptive, Minimalist Image Editing Software**

**ABSTRACT**

Currently, there are a litany of existing applications which facilitate users to edit images, such as Adobe Photoshop and Capture One. These applications provide dozens upon dozens of tools to users, all of which provide useful functionalities such as removing blemishes from faces or sharpening color contrasts. However, in interviews with photographers (the users most affected by these applications), I have found that despite this large base of functionalities, users still experience many frustrations due to the volume of these tools. Based on their feedback, I present here a new application which I contend will reduce these frustrations by only presenting a few tools at a time to the user.

**INTRODUCTION**

The motivation for this work is based on the frustrations of several interviews with professional photographers, all of whom are instructors who manage independent practices as well. The major complaint which all of these participants noted relates to the way that image editing tools are organized. Frequently, the right tools are buried deep within the popup menus, and can be hard to find, particularly for beginners. This creates a steep learning curve which can be difficult to overcome.

This problem is not new. Researchers have noted how existing interfaces overload the screen with tools and resources, forcing users to adapt to this overload with strategies such as moving menus to separate screens [6]. The photographers interviewed also noted another major problem which occurs when different editions of the same software change the location of certain tools. This change in location further increases the amount of time it takes to properly learn the software. Furthermore, it also creates long term frustration because expert users immediately become novices in a short period of time.

**PRIOR EFFORTS**

**IMPLEMENTATION**

**EXTENSIBILITY**

In order to provide the most flexibility for both users and software engineers, I implemented with extensibility in mind. Developers can extend this software in three critical areas: screens, windows, and tools.

*Screens.* Screens are simply the use screens which users see.

*Windows.* Windows allow developers to enrich their modes with interactive windows which can collect data from the users.

*Tools.* Tools are perhaps the most important extensible feature.

**REFERENCES**

[1] Banovic, Nikola, et al. "Triggering triggers and burying barriers to customizing software." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012.

[2] Guiard, Yves, Halla B. Olafsdottir, and Simon T. Perrault. "Fitt's law as an explicit time/error trade-off." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011.

[3] Laput, Gierad P., et al. "Pixeltone: A multimodal interface for image editing." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013.

[4] Fitts, Paul M., and James R. Peterson. "Information capacity of discrete motor responses." *Journal of experimental psychology* 67.2 (1964): 103.

[5] Allen, J. E., Curry I. Guinn, and E. Horvtz. "Mixed-initiative interaction." *IEEE Intelligent Systems and their Applications*14.5 (1999): 14-23.

[6] Stuerzlinger, Wolfgang, et al. "User interface façades: towards fully adaptable user interfaces." *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 2006.