Redevelopment and Evaluation of LabVIEW Programming for Adaptive Optics Imaging Methods

Joseph J. Jasper,1 Ruth E. Woehlke,2 Joseph Carroll,3,4 Kim Pierson4

1Department of Physics and Astronomy, University of Wisconsin-Eau Claire, Eau Claire, WI
2Department of Biomedical Engineering, Medical College of Wisconsin & Marquette University, Milwaukee, WI
3Department of Ophthalmology and Visual Sciences, Medical College of Wisconsin, Milwaukee, WI

ABSTRACT

Within the medical community, adaptive optics scanning light ophthalmoscopes (AOSLOs) represent a powerful clinical and experimental tool for non-invasive assessment of the living retina. However, current devices are large, unwieldy, and in limited quantity. To meet the need for a convenient mobile device capable of precision imaging, a handheld AOSLO is being built based on a design published by Duke University in 2018.1,4 The current LabVIEW operating software requires redevelopment to meet industry standards of quality and increase the speed of image acquisition, which is fundamental to advancing the potential of mobile AOSLO imaging for future applications.

The current program uses both LabVIEW and MATLAB to record, display, and process retinal images and their resulting data. This code was originally written using multiple files, coding structures, and lacked speed and efficiency, necessitating extensive redevelopment before implementation. However, some coding improvements have been completed, including the reconstruction of case structures, redefinition of variables, optimization of pathways, and creation of documentation. This code improvement project intends to optimize program speed and efficiency, update the coding structure to reflect industry standards, and produce superior levels of data and image quality when the program is operated.

OBJECTIVES

1. Align the existing program with LabVIEW industry standards of quality
2. Create documentation and reform the program into a project
3. Rewire the existing code, minimize the use of local variables and property nodes, and activate hidden functions

INTRODUCTION

Adaptive optics (AO) enables the acquisition of retinal imaging through non-invasive scanning light ophthalmoscopy (SLO) using near infrared wavelengths.2 Utilizing wavelengths that can be safely and easily tolerated by patients, this imaging method provides a non-harmful way to image a patient’s eye (Figure 1), providing cell specific approaches to monitoring diseases and therapeutic interventions.2

Figure 1: Foveal AOSLO images of a control subject (left) and subject with red green–color deficiency (right) (MCW AOIP images, Open Source)

AOSLO imaging systems are composed of five subunits: (1) illumination arms, (2) optical scanning engines, (3) wavefront sensors, (4) wavefront corrector, and (5) light detection.8 All necessary components for imaging, these five subunits makes current models extremely large and cumbersome (Figure 2) necessitating the need for a mobile, handheld unit capable of equal quality retinal imaging (1.3 Figure 3).

Possessing numerous positive attributes, the capabilities of a handheld AOALSO (HAOSLO) holds the potential to make AOSLO technology accessible to non-mobile patients as well as increase the number of AOSLO clinical imaging centers. The HAOSLO also possesses a unique challenge, as the small size creates a necessity for a robust computer program capable of performing the multiple mechanical functions that are removed from the system by decreasing its size. This challenge was met by integrating AOSLO image acquisition into a LabVIEW program.

METHODS AND MATERIALS

An object-oriented programming language heavily used in the engineering and science industry, LabVIEW allows users to visualize individual and system aspects of the application in question. For use in this project, LabVIEW is being used to gather, analyze, and interpret physical data resulting from Vision Development and Acquisition software for image creation and processing. These programs were used in conjunction with MATLAB software for polynomial generation as well as ALPAO software development kits for Zernike polynomial generation.

RESULTS AND DISCUSSION

The original program from Duke University was received with 10 software errors; these errors were the result of a non-industry standard method of programming, resulting in a lack of efficiency in organization and documentation.

In order to create a LabVIEW program possessing industry standard quality, the user interface of the program, called the front panel, was heavily modified. Controls and indicators were organized depending on their uses, with multiple tabs for different processes the user may want to accomplish (Figure 4).

The LabVIEW code files developed by Duke University were received in multiple folders. In order to facilitate sharing capabilities, all files were converted into a LabVIEW project, this being a standard method of LabVIEW program organization.

In terms of organization, many loop terminals within the code have discrepancies due to the occurrence of wire inputs and outputs. This was addressed by finding the corresponding input and output wires for each terminal and manually connecting them to their respective terminals. Originally written in an inconsistent style which relied heavily on the use of large coding areas, this length also utilizing non-industry standard format, using extensive wires and inefficient use of coding space. This was altered to reflect industry standards by compacting both the size of the entire program and the connections between nodes within the program. After observing the operation of the code, it was found that aspects of code were inefficient or being used in place of more efficient methods. In the original version of code, a flat sequence, a form of data flow control, was implemented but not controlling data flow. This, along with other case structures, were removed due to their ineffectiveness. The most important efficiency change was the reduction of local variables; local variables act in a similar manner to front panel controls, but many local variables can reference one control. Local variables are a powerful tool in LabVIEW, but decrease the speed of program when used en masse.1 To rectify this issue and maximize efficiency, twelve local variables were replaced with their corresponding, previously unwired controls.

CONCLUSION

In order to continue code improvements, it is our goal to transform the remainder of the code into a producer-consumer design pattern as there is currently no distinguishable, existing design pattern within the code. Implementing this producer-consumer design pattern would increase code efficiency, simplify interpretation, and aid in future program adjustments. This project continues to be in developmental phases and is anticipated to be applied to mechanical elements at the Medical College of Wisconsin in the summer of 2021.

REFERENCES


ACKNOWLEDGEMENTS

• Kristen Hogan – PhD Student at Duke University
• Dr. Sina Farsiu – Professor of Biomedical Engineering and Ophthalmology at Duke University