A Graphical User Interface for Robo-AO-2
Victoria Catlett¹,², Paul Barnes², Christoph Baranec²
¹The University of Texas at Dallas, ²The University of Hawai`i at Mānoa Institute for Astronomy

Introduction
We created a Graphical User Interface (GUI) to assist with the optical alignment of the Shack-Hartmann wavefront sensors (WFS) in the Robo-AO 2 laser adaptive optics (AO) system.

AO systems correct images for atmospheric turbulence by measuring wavefront distortions in real time so that they may be corrected by a deformable mirror.

Robo-AO-2 will use a Shack-Hartmann WFS, which sends light from a guide star through a lenslet array and into a camera.

The GUI will help a user minimize optical errors inside the WFS so that the system can recover correct wavefront shapes during operation.

The WFS will use 5 different lenslet arrays: a 1x1, 2x2, 4x4, 8x8, and 16x16. The 16x16 mode is shown in the central image.

The GUI will help a user minimize optical aberrations on a planar wavefront. For each lenslet mode, we create reconstruction matrices which yield the amplitude of each Zernike polynomial present in the wavefront when multiplied by a vector of the centroid displacements. The amplitudes are displayed as a histogram in the GUI. This tells the user which errors are present in the system so that they can correct them. We can recover information about 5 modes for the 2x2 array and 20 modes for the 16x16 array.

We use a similar method to calculate the overall rotation of the full image, which is displayed above the error histogram on a dial and in text.

Centroiding
The GUI uses a center-of-mass algorithm to find the centroids of each image:

The centroid displacements of each image from the center yield the average gradient of the wavefront section that passed through the corresponding lenslet.

Effects of wavefront gradient on image placement

The images have an overall bias level, which pulls the calculated centroids to the centers of the sub-apertures. To account for this, the GUI can take a dark image with the camera, then subtract it from later images to remove the bias.

Project Stack
- Python: We created the GUI structure using the PyQt5 package.
- C++: We created C++ functions to interface with the camera, which the GUI calls using Python subprocesses.
- CSS: A CSS stylesheet dictates the GUI’s design and color scheme.
- X11 Forwarding: The GUI runs on a computer in Hilo, HI, and draws the window on a user’s local computer. This requires the user to have an X Server app on their computer.

Wavefront Fitting
We use Zernike polynomials to model the effects of optical aberrations on a planar wavefront. For each lenslet mode, we create reconstruction matrices which yield the amplitude of each Zernike polynomial present in the wavefront when multiplied by a vector of the centroid displacements. The amplitudes are displayed as a histogram in the GUI. This tells the user which errors are present in the system so that they can correct them. We can recover information about 5 modes for the 2x2 array and 20 modes for the 16x16 array.

We use a similar method to calculate the overall rotation of the full image, which is displayed above the error histogram on a dial and in text.

Victoria Catlett acknowledges support from the National Science Foundation (NSF) Research Experience for Undergraduate program at the Institute for Astronomy, University of Hawai`i at Mānoa, funded through NSF grant 16104574. The Robo-AO-2 system is supported by the National Science Foundation under Grant No. AST-1712014, the State of Hawai`i Capital Improvement Projects, and by a gift from the Lumb Family. VC would also like to thank the Institute for Astronomy for their kind hospitality and commitment to remote operation during the course of this project. VC thanks the Robo-AO team for their suggestions and critiques throughout the project, including Reed Riddle and Maissa Salama.