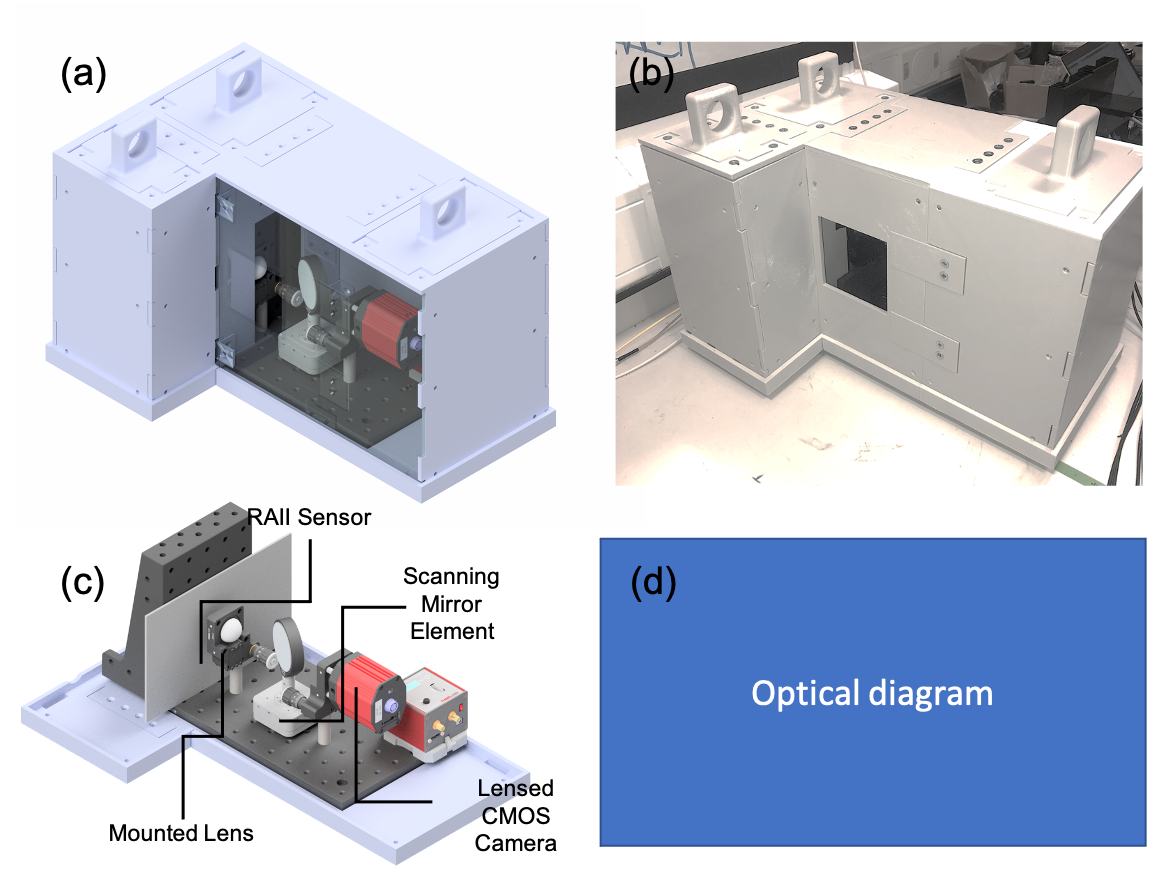
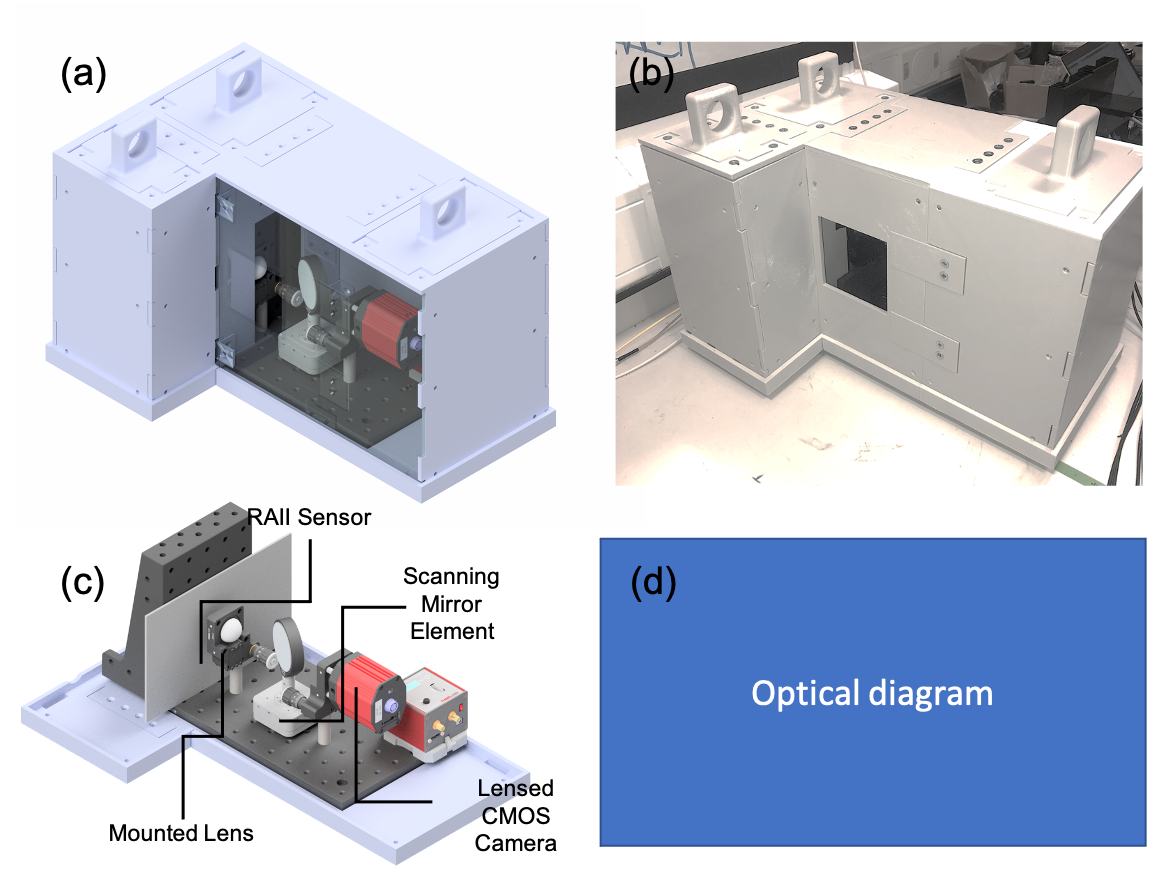
**User Manual: RAII Imaging System**

08-08-21

**System Design**

The RAII based imaging system consists of a series of optical elements used to sweep the FOV of the RAII line sensor in a second axis, recovering a 2D image consisting of a series of stitched together sequentially acquired image ‘slices’.



Key system components are highlighted above, along with a CAD rendering of the 3D printed plastic housing.

The RAII line sensor performance and design is detailed [elsewhere.](https://ieeexplore.ieee.org/document/8637804)

Information on the scanning [rotation mount](https://www.thorlabs.com/thorproduct.cfm?partnumber=DDR25/M), [mirror](https://www.thorlabs.com/thorproduct.cfm?partnumber=PF20-03-G01), [lensing](https://www.thorlabs.com/thorproduct.cfm?partnumber=LA1951-B), [filter](https://www.laser2000.co.uk/product/ff01-780§12-25) and CMOS camera are also available online.

**System Operation**

Data acquisition, mirror movement (motor movement) and histogram creation are performed in an automatic manner, running a single MATLAB function located within the file TCSPC\_Iterative. (see attached). Several steps are required before this can be carried out however to link both the motor and RAII sensor hardware to MATLAB, allowing results to be obtained thereafter. I will step out the process of turning the system on, before then outlining how imaging is performed using the system.

1. Initial connection: Supplementary MATLAB scripts will be used to connect the motor and RAII sensor. These being ‘motor.m’ (see inside for list of functions) and ‘SensorStart.m’.

**Motor:** Switch on power to motor before connecting motor USB 3 to computer (important). After a few seconds connect the USB to the computer. Launch ThorLabs software package ‘Kinesis’. Connect to the motor using kinesis, enable the motor, and home the motor. (both of these are buttons on their GUI interface) Following this, disconnect from the motor on kinesis (do not disconnect the USB). Switch to MATLAB (ensure motor.m is readable from your current folder) run the following lines of code manually line by line.

m1 = motor; a = motor.listdevices;

connect(m1,a{1}); home(m1);

disconnect(m1);

Now the motor should be connected correctly, and this element of the setup is ready for use.

**RAII Sensor:** Ensure SensorStart.m is on the path. From the console run SensorStart; Wait a few moments until the compiler finishes spitting out the voltage ramp values, then this element is ready to use. (note: SensorStop is used to turn off the sensor).

1. Data Acquisition: For scanned TCSPC measurements, this is controlled exclusively through the TCSPC\_Iterative.m script. A single function is used to control the sensor voltage values during acquisition, acquisition parameters (number of frames, exposure time per frame) and the movement parameters for the motor, these being the step size (in degrees) and the number of steps, describing the total movement of the sensor and the number of lines of TCSPC acquired. This function has the form TCSPC\_Iterative(Bias Voltage, Quench, Number of frames, exposure time, number of steps, step size).

Using this function allows the user to create a 3D matrix containing histograms for each ‘pixel’ in the scan, the script spits this out as a single ScanResults.mat file to a predefined location on the hard drive.

1. Automated Scanning Mechanism: Many separate scans can be acquired automatically by using a predefined number of scans (the data acquisition elements of the script are looped over this number of iterations). This can be changed via the ‘NumberOfScans’ flag.