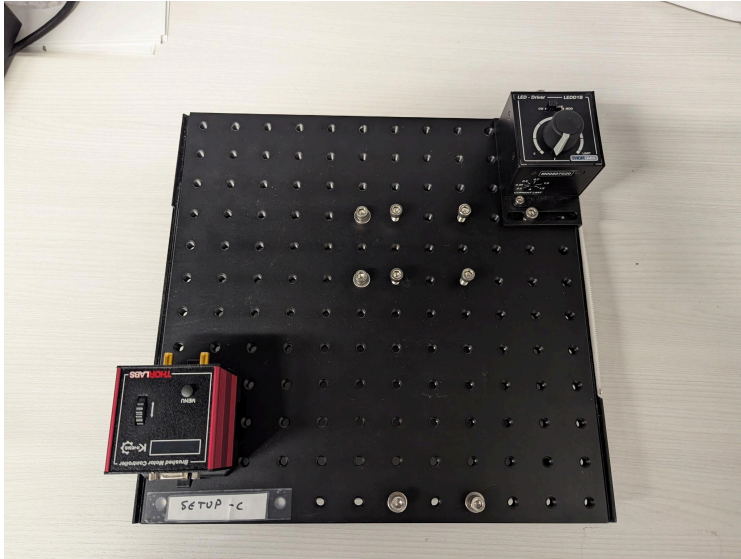


15-463, 15-663, 15-862 final project

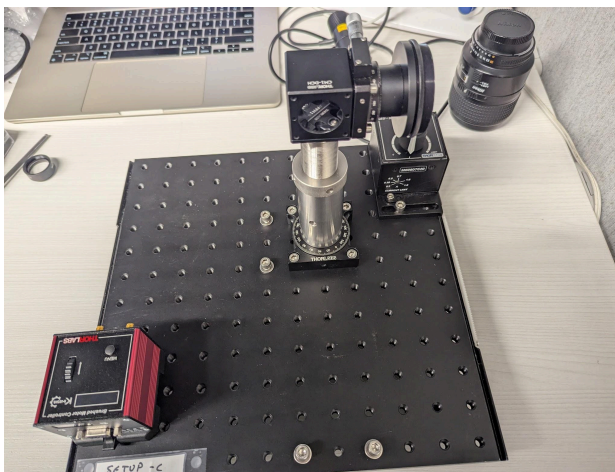
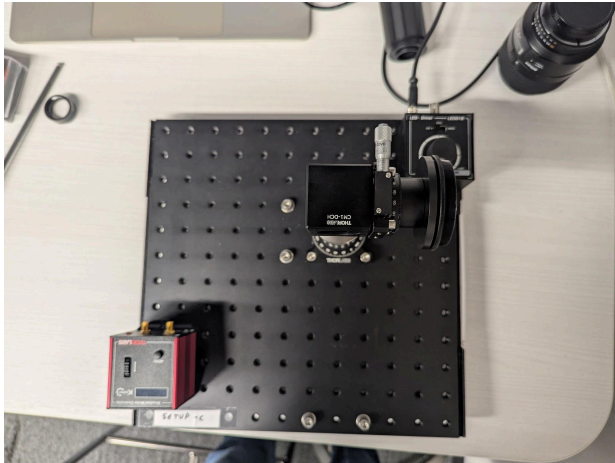
Interferometric Imaging

Part 1: Construction

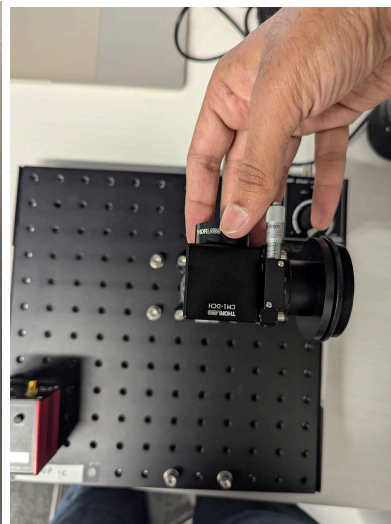
1. Your breadboard should look something like this:



2. Mount the main post with the beamsplitter (BS) using the cap screws at the designated location on the breadboard. Make sure it is oriented correctly



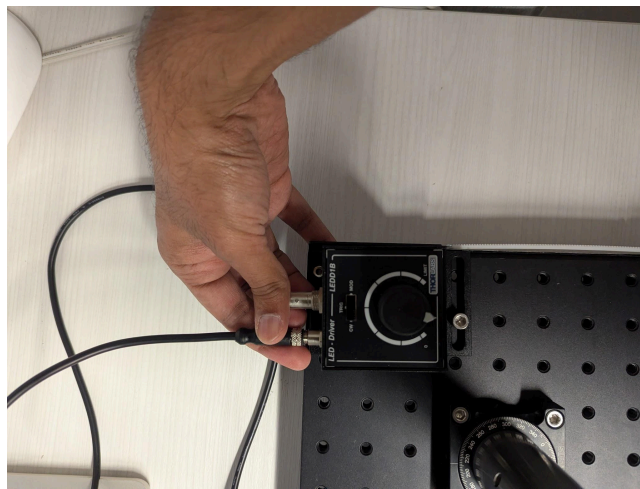
3. Light source arm:
 - a. Screw on the 100mm Thorlabs lens.



- b. Screw on the LED with the telescoping tube onto the threading of the lens



- c. Connect the LED to the LED driver, and try powering it up with the power adapter.



4. Camera arm:

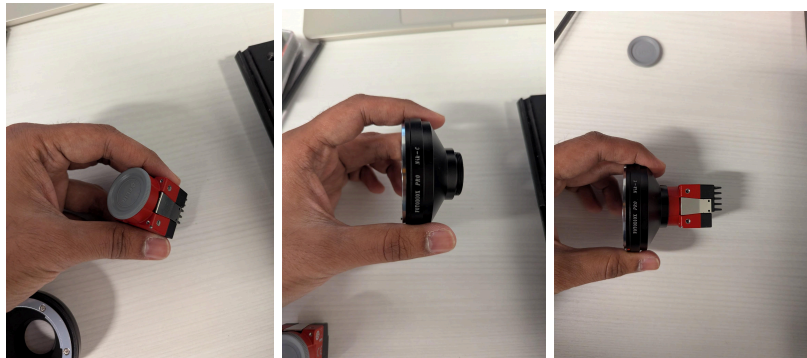
- a. Screw the 105mm (or 200mm) Nikon lens on the threading at the right end of the BS cube



- b. You should have a C-mount camera that looks like this:



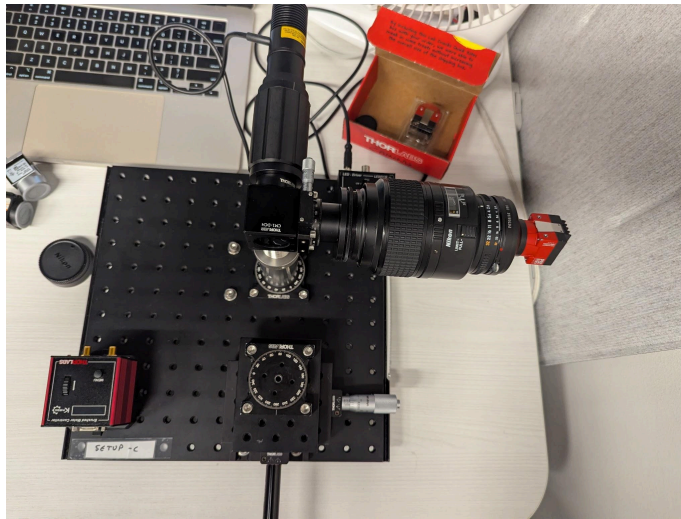
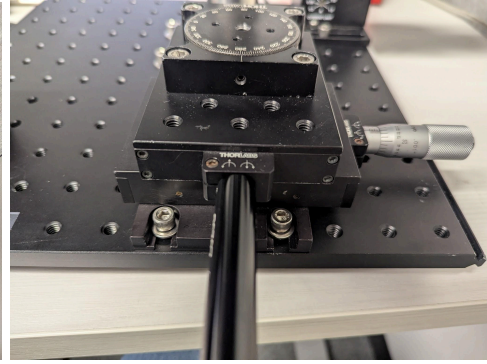
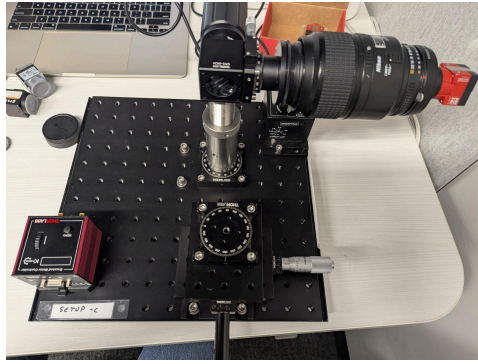
- c. Screw the C-mount camera onto the F-C mount, and *then* attach it to the back of the lens. (careful not to expose the sensor.)



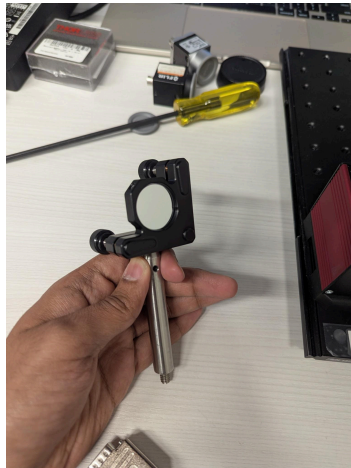


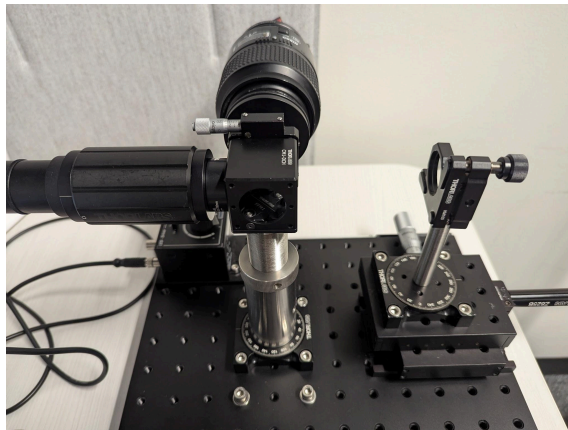
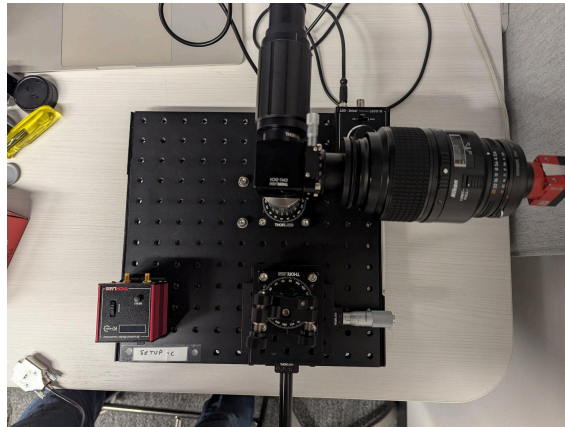
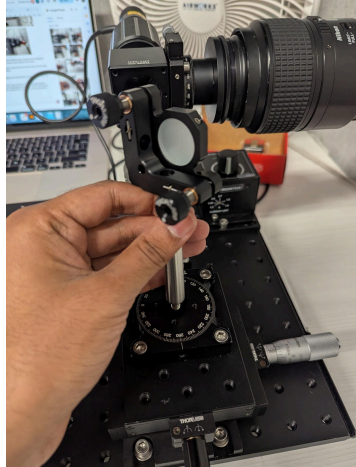
5. Reference arm:

- a. Place the XY stage (with motorized Z825B actuator inserted) at the designated location on the breadboard and secure using washers and cap screws.

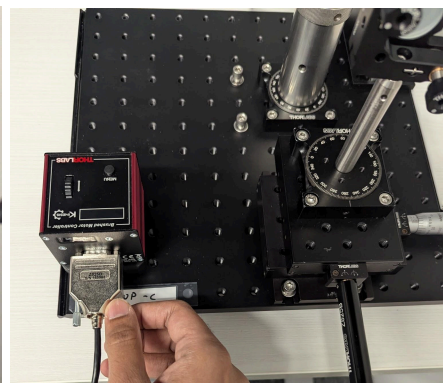
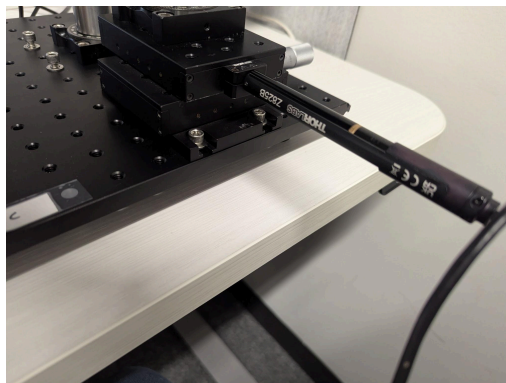


- b. **Careful** not to touch or damage the mirror, screw in the round mirror mount onto the stage using the set screw at the bottom

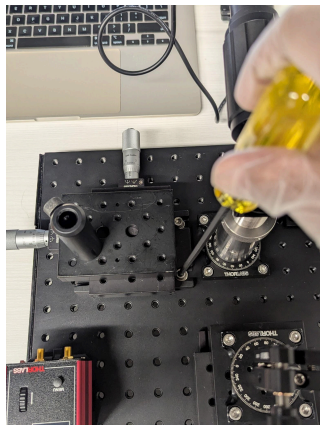
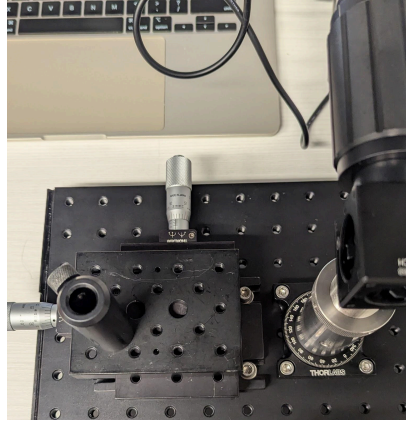




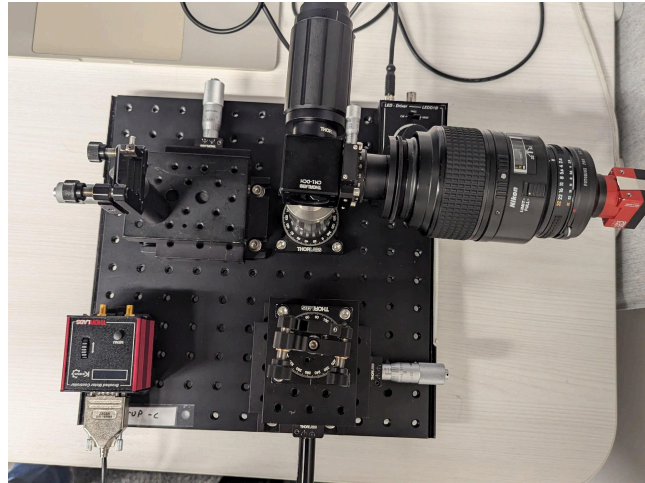
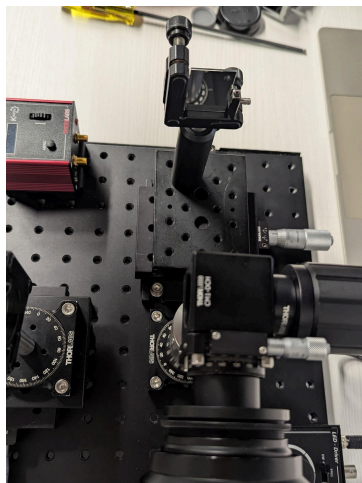
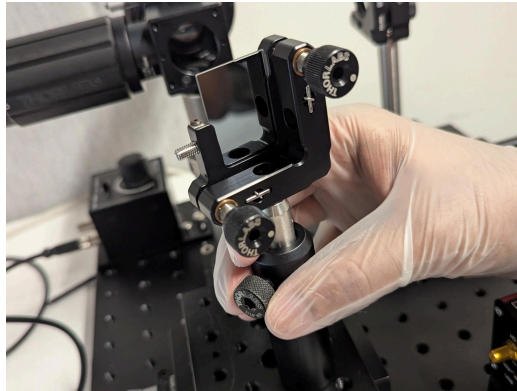
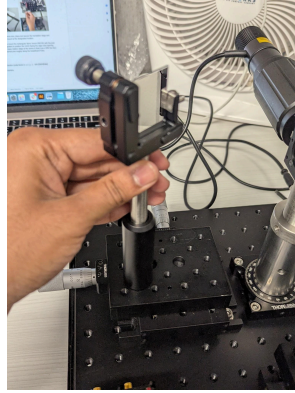
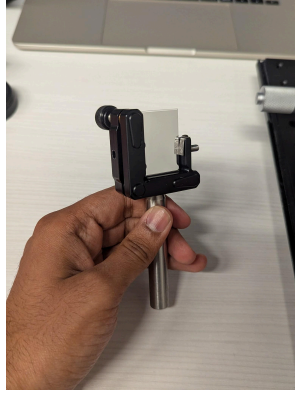
- c. Connect the end of the Z825B motor to the KDC101 Kcube motor controller (red cube)



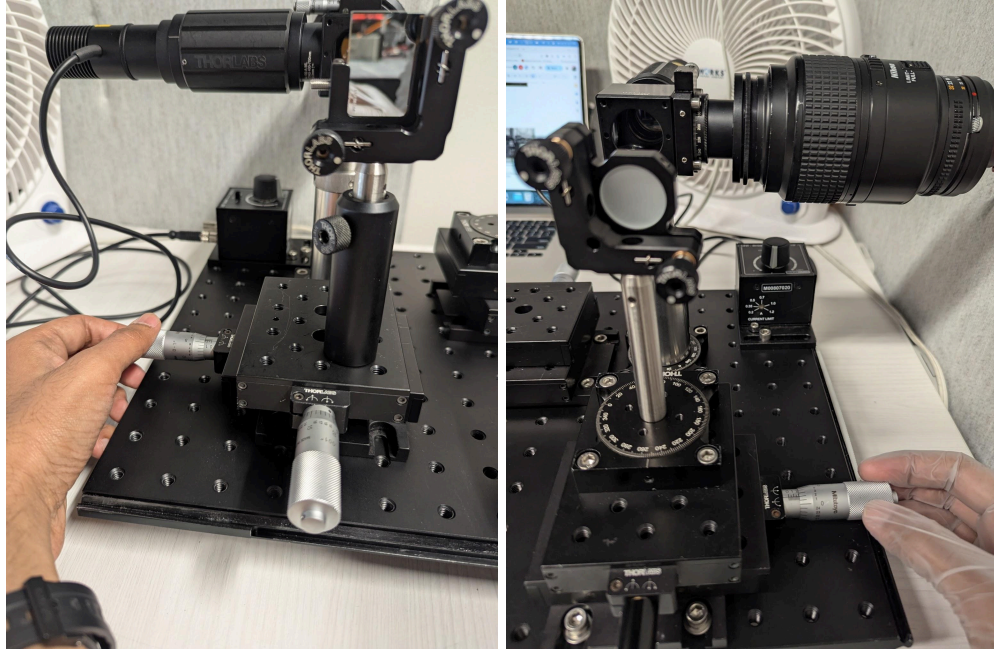
6. Object arm:
 - a. Similar to the reference arm, place and secure the XY translation stage for the object side (shown below) at the designated location



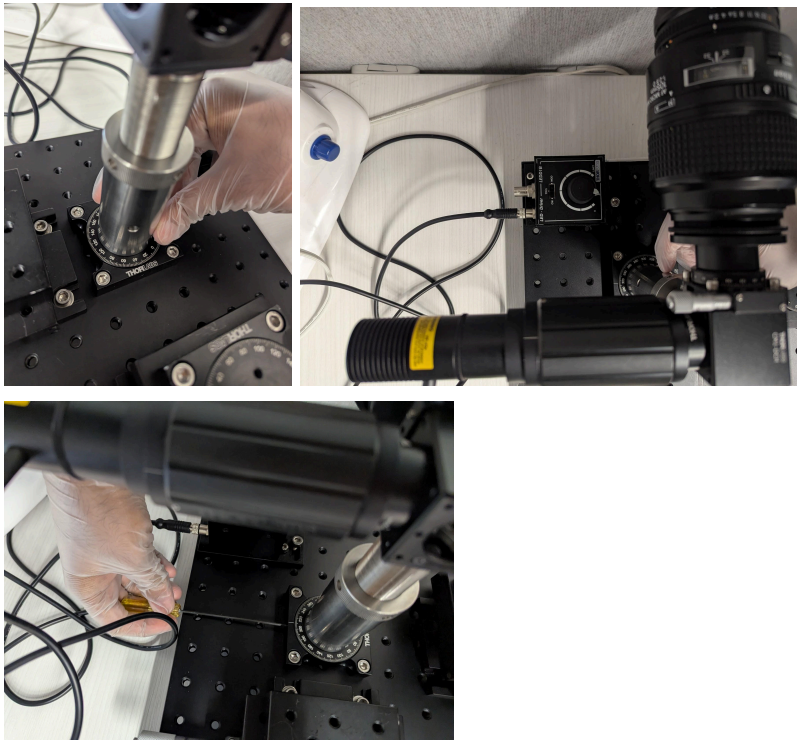
- b. Drop in the post holding the rectangular mirror, careful not to touch or damage the mirror.



- c. Use the *lateral* micro-adjusters to position the mirrors to be facing directly opposite the cage-cube opening.



7. Fix the orientation of the base rotation stage at the centre is fixed (ask for a 5/64 hex key) such that the LED and camera are roughly along the breadboard holes.



Part 2: Alignment

Note: You can execute each section (code block) in `setup.m` with **[Ctrl+Enter]**

1. Getting light into the camera:
 - a. Ensure the lens aperture is fully open.



- b. Connect to the camera feed using the camera-dependent launch function and begin camera preview.

```
%% connect to camera
[vid,src] = flirOpen(1);
% [vid,src] = alviumOpen(1);
src.ExposureTime = 2e4;

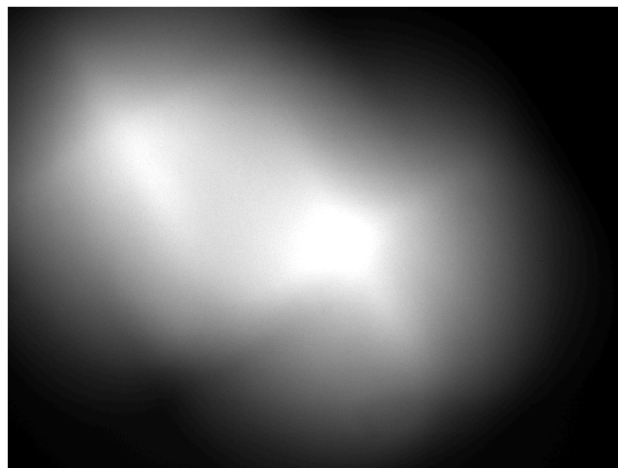
%% open camera preview
[imObj] = previewinteractive(vid,false);
```

- c. Adjust your camera to be roughly upright by rotating it about the rotation mount with graduations.



- d. Switch on the LED, and try to coarsely align the orientation of the mirrors to get maximum light into the camera by:
- coarsely adjusting the mirror post mounts.
 - using the mirror adjuster knobs

You should see two 'blobs' or (likely blurred) images of the LED that move as you adjust the mirrors.



Note: You may need to lower the power of the LED and reduce your exposure. You can adjust exposure using `src.ExposureTime = ..` at the MATLAB command line (in microseconds).

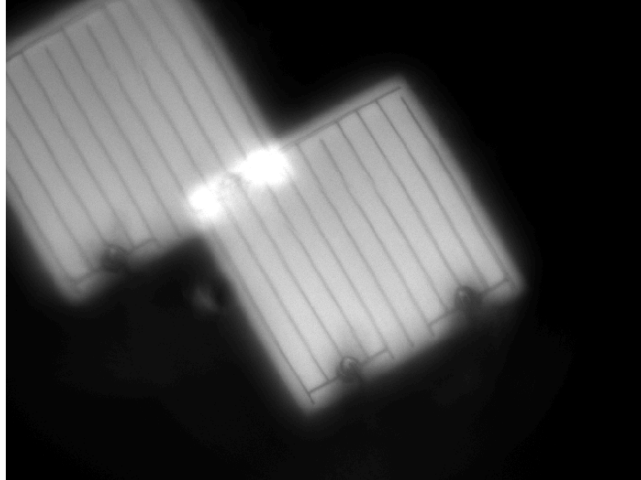
2. Collimating the LED:

- a. Set the lens to infinity focusing.



- b. Adjust the telescoping tube until the images of the active element of the LED (this may look different for each LED) look as sharp as possible.





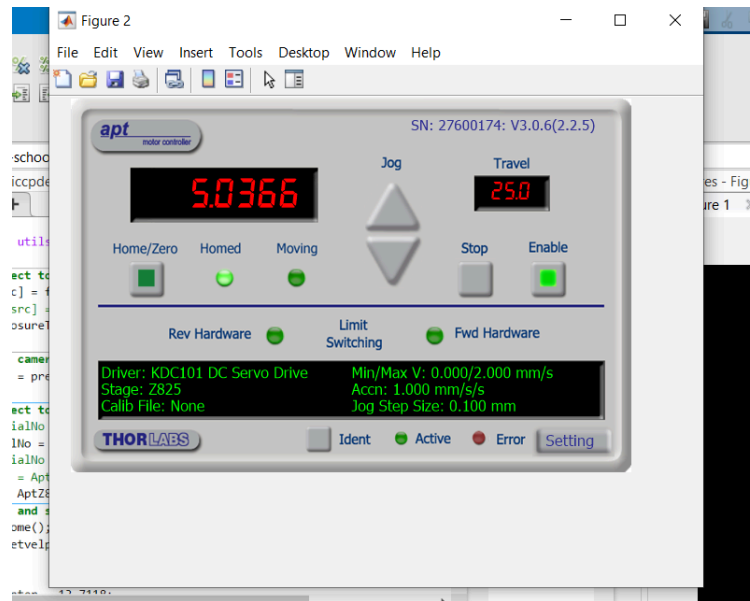
You may choose to block one of the arms using a piece of card to make alignment easier.

3. Aligning the mirrors:
 - a. Use the adjusters on the mirror mounts to position both images of the LED roughly at the center of the field of view of the camera.
4. Controlling the reference mirror motor:
 - a. Connect the USB cable to the laptop and turn on the KDC101 'Kinesis' motor controller using the red switch.

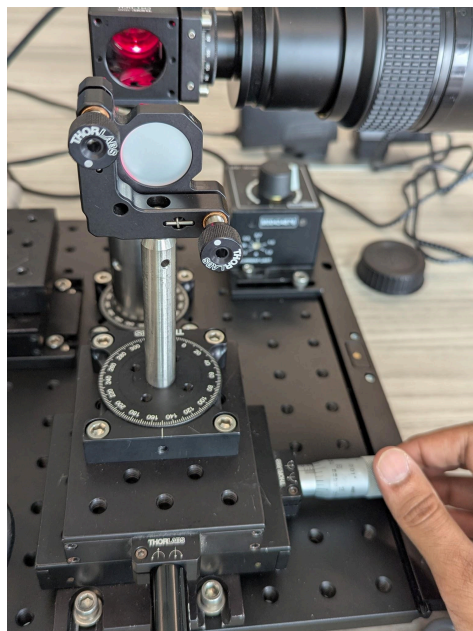


Connect to the motor using `AptZ825B(serialNo)` . A pop-up Figure window (screenshot below) should show up. Make sure the serial number is the one on

the motor controller (S/N:...)

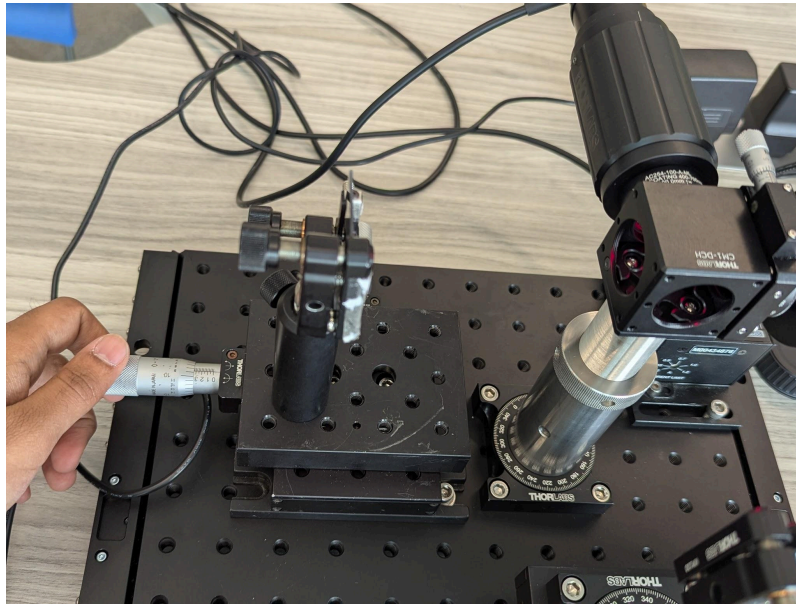


- b. Home the motor using `motor.home()`
 - c. Set the reference motorized translation stage at about **5mm**, using `motor.goto(5)`. Leave it in this position for the next few steps.
5. Coarse pathlength matching
- a. Block the object side using a piece of card. Change the lens focusing such that the reference mirror comes into focus (about 1:1); Tip: You can laterally translate the mirror using the micro-adjusters (image shown below) until you see the edge of the mirror, so it is easy to see if it comes into focus. You may also use dust/scratches present on the mirror.

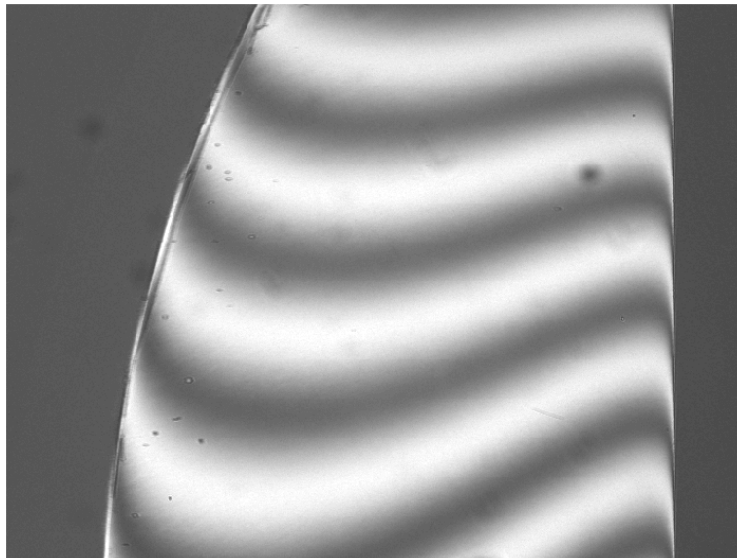


You will need to increase the exposure time and/or LED power.

- b. Measure the distance from the BS to the ref mirror using a ruler or tape measure. Adjust the object side mirror's translation along the optical axis until it is roughly the same.

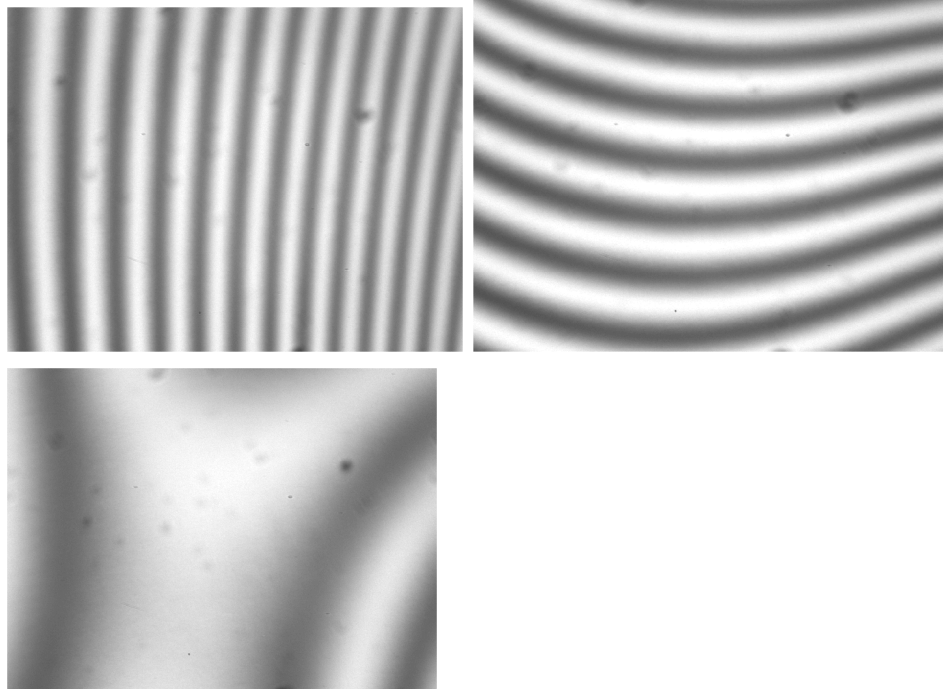


- c. With the reference mirror in focus, keep adjusting the object arm's distance until it also comes into near-perfect focus.
6. Fine pathlength matching
- a. If you're lucky, you may already see bright **interference fringes** as shown below. You can laterally translate the mirrors back so that you see fringes in the entire fov of the camera. Because of imperfections of translation orientation, you may need to readjust distances along the axes slightly.



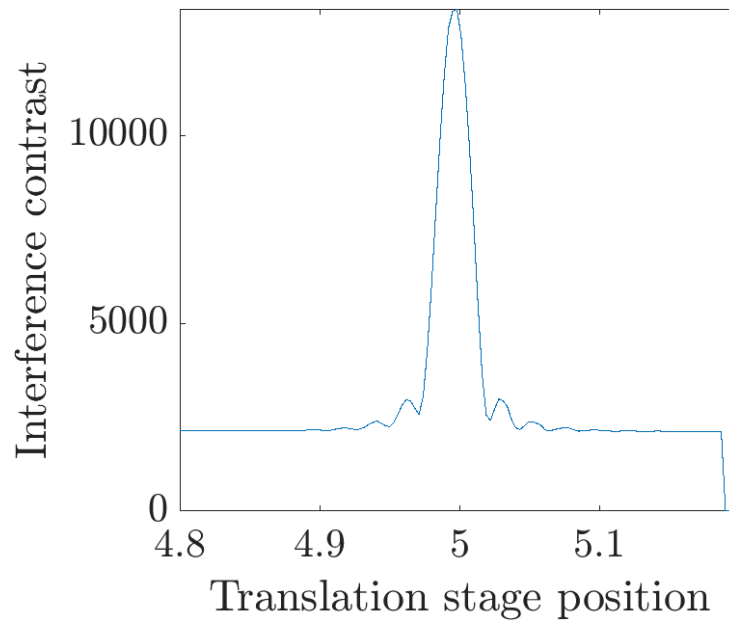
If you don't see fringes yet, do the steps b and c below:

- b. Make sure that both mirrors still have the right orientation by switching back to infinity-focusing and matching both images of the LED correctly. Making this nearly perfect will make fine path length matching easier. (Focus back onto the mirrors once you're done checking)
- c. Use the manual micro-adjuster on the object arm and very slowly adjust it in half-turns and keep looking at the screen. At some point you should start to see interference fringes
 - i. If this is unsuccessful, you can also use `align_pathlength()` over a wide range (say 2 mm, in steps of about 20 micron). which performs a linear scan of the motorized reference arm in small steps around the current position, and automatically determines the best position.
- d. You can use the mirror-orientation adjusters to change the interference fringe pattern.



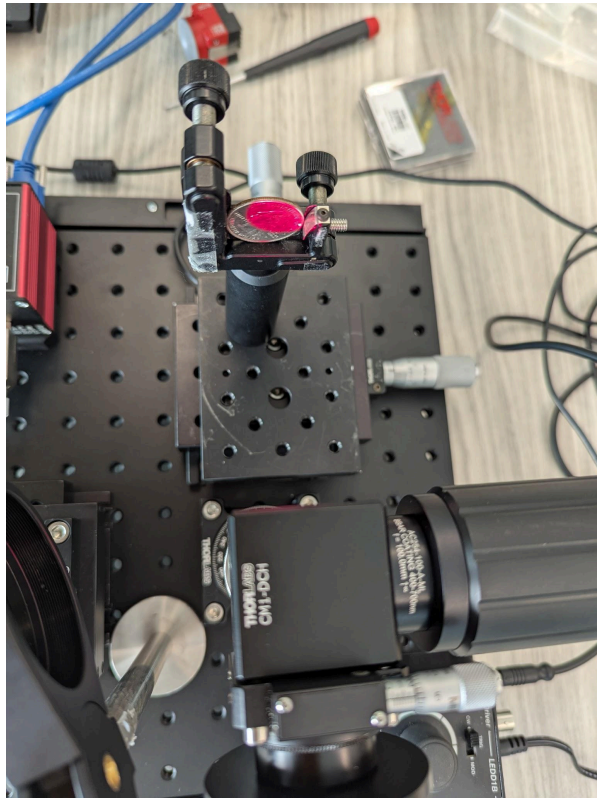
7. Measuring temporal coherence length:
- a. Use `align_pathlength()` to do a finer scan (steps of 3-5 micron over a range of about 200-300 microns) to refine this position such that you get maximum contrast. This should get you a rough plot of contrast, and allow us to determine temporal coherence length of the spectrally filtered light reaching the

camera. This should be our *depth resolution* for OCT.



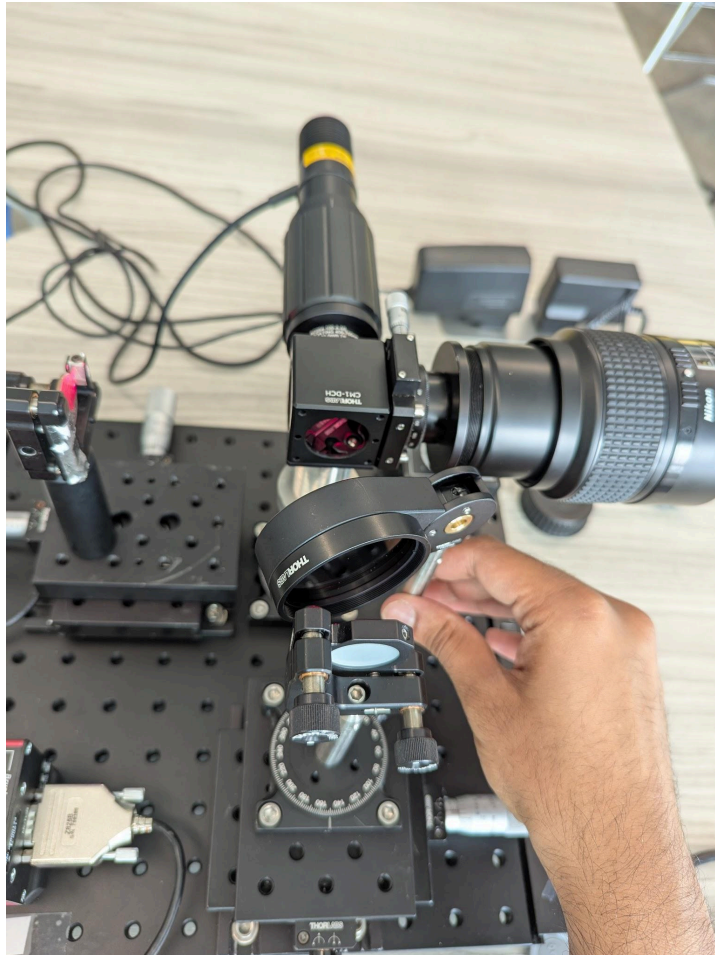
8. Scanning a coin

- a. When you're ready, ask for help to remove the rectangular mirror from the mirror mount. You can then replace it with a coin (quarter/euro/..).



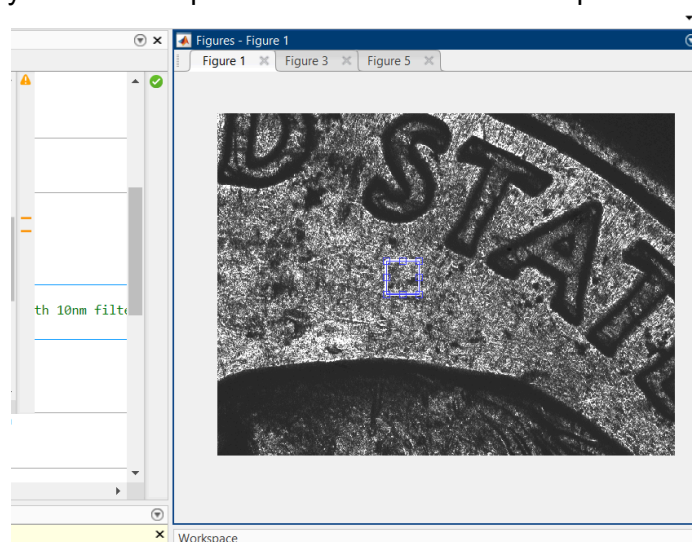
- b. Insert the 0.6 OD ND filter (placed at a slight angle to avoid interreflections) in the light path to roughly match intensities from both arms. (You will need to readjust

your exposure time.)

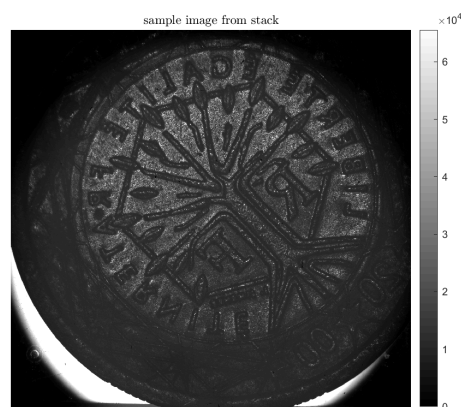


- c. (optional) You may readjust your focus depending on the object surface position. You may also (optionally) reduce your aperture size to increase your depth of field.
- d. Crop a small region of the FOV, and perform a scan using `align_pathlength()` once again, over a length comparable to the error due to the difference in thickness of the coin and the mirror (1-2mm). This should get

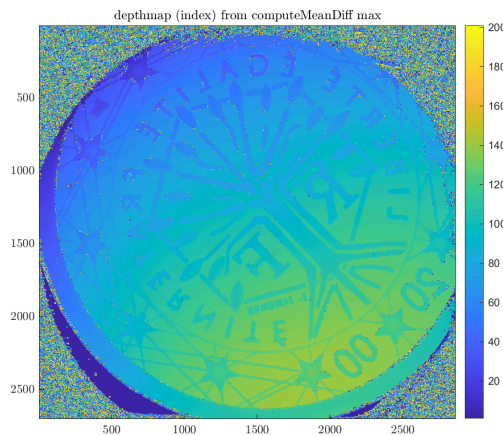
you a 'center' position around which we will perform a scan.



- e. Now perform a scan over a length of 1mm. Use the code block provided in the setup script. Use the temporal coherence length determined in step 6 to determine the step size. The data captured may take a while to save.



- f. Call `computeMeanDiff()` to recover a depth map from the scanned data.



9. Scanning other objects:
- Replace the coin with another object (or coin) of your choice and try doing the same!

(additional information)

Software Installations:

- **MATLAB (R2021b)**
 - Image Acquisition Toolbox
 - GenICam interface [add on](#).
- Thorlabs [APT](#) (see 'Archive' Tab)
- Camera software (useful to have viewer to debug any issues, and install any drivers to detect the camera correctly):
 - AVT camera: [Vimba X SDK](#)
 - FLIR cameras: [Spinnaker SDK](#)
 - Basler camera: [pylon](#) (required)
- (Optionally) Thorlabs [Kinesis](#)