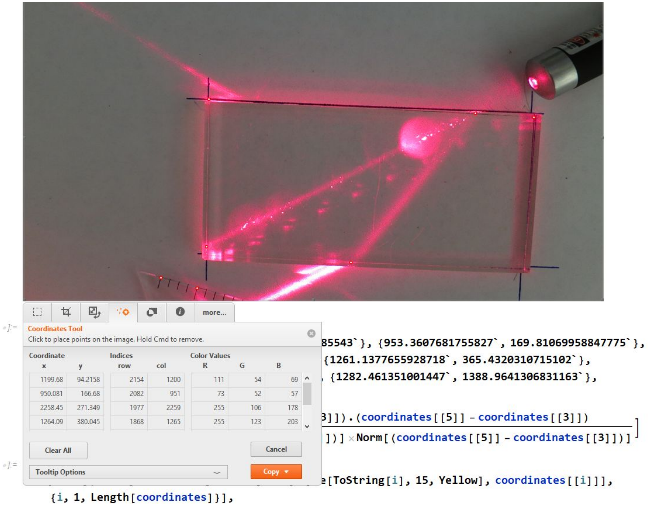
Aerogel Index of Refraction

Update 8/9/19

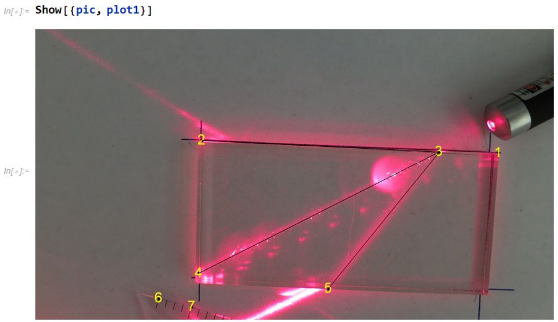
Image Analysis - Plastic Scintillator

I finished up a program started by Nick that calculates the index of refraction from a selection of points on a picture.

I have not thoroughly examined the results yet and might like to add an additional data analysis section to the program.



(1) (Above) Take coordinates in mathematica using image coordinates tool



(2) (Above) Coordinates and order highlighted

(3) (Below) End product of the code

- Allows you to input a set of coordinates and gives you information on angles and distances and a value for n

```
In[•]:= refractiveindex3[coordinates12]

Out[•]= α = 50.5189, β = 31.5234 γ = 18.9956 y =
        3.01003 x = 1.49933 conversion factor = 107.905 n = 1.17896
```

Initial Scintillator Image Analysis Results and Speculation

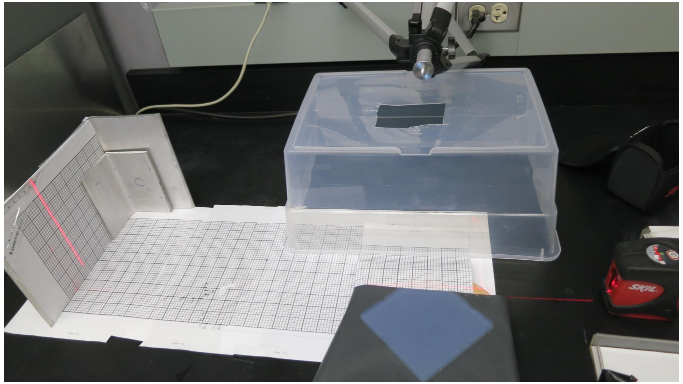
- Generally, the numbers I got for the index of refraction were around n = 1.0 or n = 1.2 - far from the expected value of 1.5
- I reverified all of the equations and processes in the program
- Possible sources of this large error:
    - The pictures may not be horizontal enough or directly over the scintillator
    - The program is not precise enough because the edges and corners of the tile and laser beam are too hard to correctly label manually and with just a few points
    - There is something else wrong with the code or equations that I did not catch
- I would like to go back an analyse the data more thoroughly and get a better error estimate - possible retake the overhead pictures with a better setup

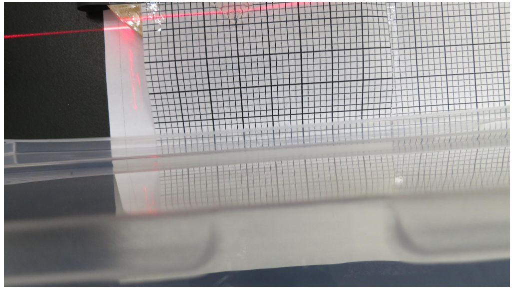
Aerogel Pictures

- I spent some time taking new pictures of the laser refracted through the aerogel tile to use for image analysis
- 2 new sets of pictures
- Used the 3rd tile (stored under the broken pieces) because it looked like it had decent edges

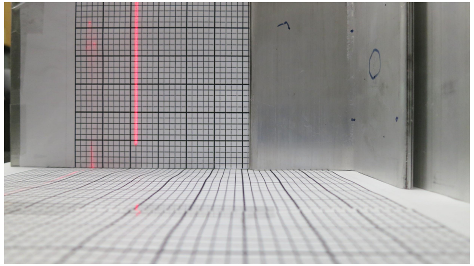
Aerogel Pictures set 1

- 3 pictures for each laser position - gradually moved the laser back into the tile up to about 1.5 inches in
    - (2) Overhead picture taken by attaching the camera to a horizontal tripod (1) and verifying that it was horizontal with a level
    - 2 pictures of the target taken at 2 parallel camera positions (3)
- The overhead picture will be used to verify alpha and beta for each set of measurements and corner of the tile



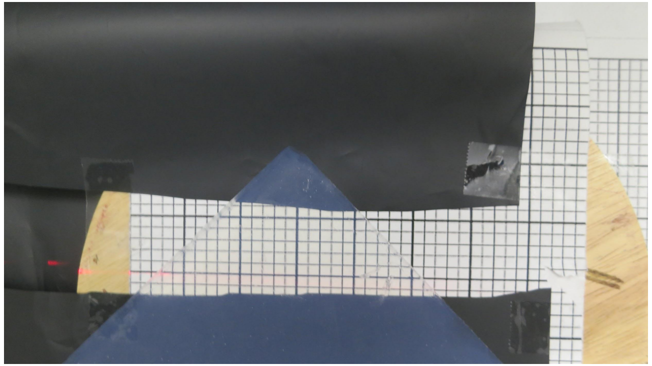


(1) Above, setup for these pictures
(2) Left, overhead picture example
(3) Below, target picture example

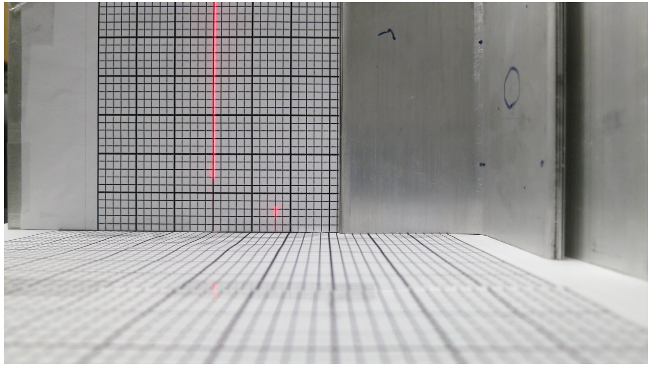


Aerogel Pictures set 2

- Took pictures of the laser refracted through the tile closer to the center of the tile in order to see if there were any variations in the refractive index
- Took pictures of the target, moving the laser's position from hitting the tip to the middle of the aerogel
- Setup: A strip of graph paper between two protective surfaces to verify tile/laser position
  - Took overhead shots for reference but did not make sure they were horizontal
- Took 2 pictures from different parallel camera positions of the target for each laser position

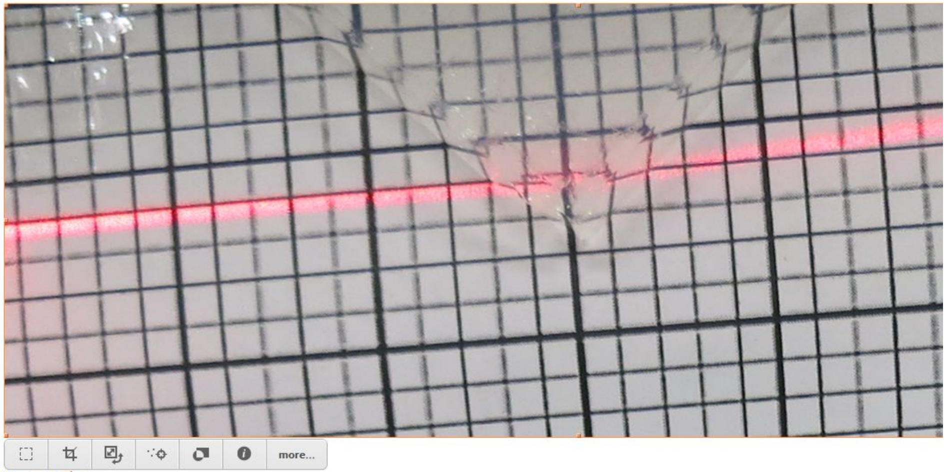


(1) Above, setup picture
(2) Below, target picture example



Aerogel Image Analysis

- I put together the first section of the aerogel image analysis program in Mathematica for the overheads from the1st set
- The program fits lines to a selection of 30 coordinates taken with the coordinates tool, outlining the edges of the tile and the incoming laser outside the tile
- It then takes the intersection points between those fitted lines and measures the angles to give values for alpha and beta



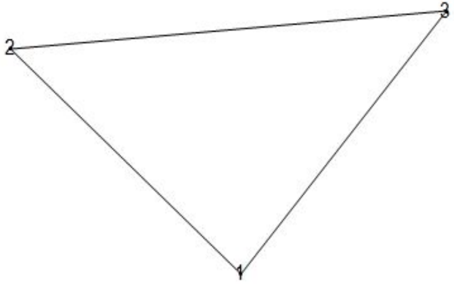
(Attempt 3)

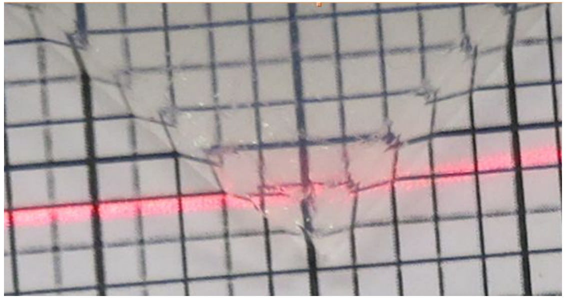
```
ocoordinates1 (*Overall set of coordinates → 1st 10 describing left line,
 then 10 describing right line and finally 10 describing the top edge of the laser beam*) =
 Reverse[{{602.1518397764323`, 330.1369352585002`}, {569.852507374631`4, 326.6762925011644`}, {496.0254618848006`, 322.0621021580504`},
   {432.5803446669772`, 316.29436422915694`}, {370.28877503493254`, 311.68017388604255`}, {321.839776432231`, 308.21953112870676`},
   {242.24499301350724`, 302.45179319981366`}, {175.3392330383481`, 296.6840552709207`}, {119.96894892097501`, 292.0698649278062`},
   {78.44123583294521`, 288.6092221704704`}, {831.7078093463749`, 343.9795062878435`}, {900.9206644930913`, 415.49945660611706`},
   {910.1490451793201`, 430.49557522123894`}, {943.6019251668997`, 469.71619313771157`}, {975.9012575687008`, 505.4761682968484`},
   {1013.9683278993947`, 551.6180717279926`}, {1000.1257568700513`, 534.3148579413136`}, {962.0586865393574`, 489.32650209594783`},
   {928.6058065517778`, 452.4129793510325`}, {879.0032603632977`, 395.8891476478808`}, {504.1002949852508`, 435.10976556435344`},
   {436.04098742431313`, 496.2477876106195`}, {446.4229156963205`, 489.32650209594783`}, {610.2266728768826`, 346.2866014594007`},
   {587.1557211613105`, 363.5898152460798`7}, {569.8525073746314`, 377.43238627542314`}, {546.7815556590592`, 398.196242819438`},
   {490.2577239559075`, 446.6452414221394`}, {515.6357708430369`, 425.88138487812455`}, {460.2654867256638`, 473.17683589504736`}}]
```

Aerogel Image Analysis

(1) (Top) - Shows the segments of the fitted lines between their intersection points drawn from the coordinates and the original image for comparison

(2) (Bottom) The code and final input and output





```
In[ ]:= AerogelAngles[ocoordinates_] := Block[{α, β, lcoordinates, rcoordinates, hcoordinates, lline, rline, hline, a, b, c, set1, set1f},
   lcoordinates = {ocoordinates[[1]], ocoordinates[[2]], ocoordinates[[3]], ocoordinates[[4]], ocoordinates[[5]], ocoordinates[[6]],
     ocoordinates[[7]], ocoordinates[[8]], ocoordinates[[9]], ocoordinates[[10]]};
   rcoordinates = {ocoordinates[[11]], ocoordinates[[12]], ocoordinates[[13]], ocoordinates[[14]], ocoordinates[[15]], ocoordinates[[16]],
     ocoordinates[[17]], ocoordinates[[18]], ocoordinates[[19]], ocoordinates[[20]]};
   hcoordinates  = {ocoordinates[[21]], ocoordinates[[22]], ocoordinates[[23]], ocoordinates[[24]], ocoordinates[[25]], ocoordinates[[26]],
     ocoordinates[[27]], ocoordinates[[28]], ocoordinates[[29]], ocoordinates[[30]]};
   rline = Fit[rcoordinates1, {1, x}, x];
   lline = Fit[lcoordinates1, {1, x}, x];
   hline = Fit[hcoordinates1, {1, x}, x];
   a = Solve[{y == lline && y == rline}, {x, y}];
   b = Solve[{y == hline && y == lline}, {x, y}];
   c = Solve[{y == hline && y == rline}, {x, y}];
   set1 = {{{x, y} /. a}, {{x, y} /. b}, {{x, y} /. c}};
   set1f = ArrayFlatten[set1];
   β = PlanarAngle[{set1f[[2]], set1f[[1]], set1f[[3]]}];
   α = (Pi / 2) - PlanarAngle[{set1f[[1]], set1f[[2]], set1f[[3]]}];
   Row[{"α = ", α (180 / Pi), ", ", "β = ", β (180 / Pi)}]]

In[ ]:= AerogelAngles[ocoordinates1]

Out[ ]= α = 44.4533, β = 90.0874
```

Projects

- Aerogel image analysis:
  - Put together a code to give values for x from the target pictures in the 1st and 2nd sets
  - Similarly, a code to give a value for n from the aerogel pictures and to analyse the results
- Plastic scintillator image analysis:
  - Finish data analysis section of the program
- Vertical Index of Refraction measurements
- Root
  - I would like to use root to create a 3D graph of the variations in n across the tile
  - I have started the process of getting access to the JLab version of Root for this semester
- Transmittance Measurements
  - A project for this coming semester?