



LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-T1400515-v9

LIGO

July 27, 2015

Process for aLIGO Particle Imaging Using a Nikon D7100
and Matlab

Sabrina Waller

Distribution of this document:
LIGO Scientific Collaboration

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project

Massachusetts Institute of Technology
LIGO Project

LIGO Hanford Observatory

LIGO Livingston Observatory

<http://www.ligo.caltech.edu/>

1 Introduction

The purpose of this paper is to demonstrate how to use a Nikon D7100 and the accompanying MATLAB code to count and measure the particles on the surface of an optic.

The design for LIGO (Laser Interferometer Gravitational Wave Observatory) centers around the use of high intensity lasers aimed at a series of optics to detect gravitational waves. However, when the laser beams hit the optics, particulate contamination on the surface of the optics results in interference with the beam and damage to the optics. Therefore, there is a need to be able to efficiently and accurately identify the contamination level of the optics within Advanced LIGO. Toupview, the program currently used to process images and measure the pictured dust particles, forces the user to go through every step of processing the image by hand. The development of a straightforward, automated process is essential for enabling quick, accurate determinations of what optics need cleaning and if there is an unacceptable level of contamination remaining after cleaning. To meet this need, I developed the following process wherein a Nikon D7100 with an AF Micro Nikkor 200mm 1:4 D lens and a Nikon TC-201 2x Teleconverter can be used to take pictures of the optics. I then wrote code to process the images in MATLAB utilizing MATLAB's Image Processing Toolbox. The program crops the image, counts the number of particles, and measures the particles, providing immediate information about the contamination on the optics. With the test camera and distance, MATLAB detects and measures the particles on the images down to 10.0066881 microns with a maximum margin of error of 4 microns.

The following process is based on tests images taken at a distance of 40 inches of a 3 inch diameter optic surrounded by a ring of green LEDs that illuminate the dust on the surface of the optic.

A detailed account on early development of this process can be found in my two progress reports, the first of which is located in the document [LIGO-T1400474-v2](#) and second of which is located in the document [LIGO-T1400506-v2](#).

2 Taking Pictures

2.1 Camera Set Up

1. Remove the cap from the back of the Nikkor lens and the thinner but larger diameter cap of the TC-201 Teleconverter. There will be a white dot on the rim of each of device. Attach the two devices by lining up the white dots on the rim of both edges and turning counterclockwise until you hear a click.



End of the Teleconverter that Connects to the Lens

2. Remove the cap from the front of the Nikon D7100 and the thicker but smaller diameter cap from the TC-201 Teleconverter and repeat the process to connect the teleconverter and lens to the camera.



End of the Teleconverter that Connects to the Camera



Connecting the Teleconverter and the Camera

3. Carefully turn the camera over and use a screwdriver to attach the tripod mount to the mount platform on the bottom of the lens. Do not attach it to the camera itself because with the additional weight of the teleconverter and lens, the camera system would be unbalanced and unsteady.



Attaching the Tripod Mount to the Lens

4. Place the camera with the attached tripod mount on top of the tripod and rotate the knob on the side of the tripod until it locks into place.



Assembled Camera on the Tripod

5. Set up the Camera in the desired location.
6. Use the USB cable that comes with the camera to connect the camera to an external computer. Open the program ControlMyNikon, select the Nikon D7100 from the drop down menu next to the connect button and click connect. It may take several minutes to connect, so do not worry if it does not connect right away. You cannot use a USB cable other than the one that comes with the camera because the camera will not be able to connect to ControlMyNikon with a cable not made specifically for the Nikon D7100.
7. Change the camera to the following settings, which result in the clearest, most consistent images of the optic out of any other tested combination. These settings are for low levels of external light, so if you are working in a bright area, increase the shutter speed until the images reach an optimum quality.
 - a) On the body of the camera, there is a black switch located on the side of the camera, near where it attaches to the teleconverter with a release button on it and two options “AF” which stands for Autofocus and “M” which stands for Manual printed on the camera above it. Hold the release button down and move the switch so that it points to “M.”



Black Switch Turned to Manual

- b) On the top of the camera body, there is a main dial with ten options printed on top, including “M,” “A,” “S,” “P.” Press the release button on the top of the dial, spin the dial so that the white line on the camera points to “M” which stands for manual, and let go of the release button.



Main Dial Turned to Manual

- c) This dial is stacked on top of a second dial. Press the release button for this dial, which is located just to its left, and spin the dial so that the white line on the camera points to “S”, which stands for single frame, and let go of the release button.



Bottom Dial Turned to Single Frame

- d) If you look at the Nikkor lens, about halfway down the body of the lens, there is a black band with a silver button and a white box with the two letters “M” which stands for manual and “A” which stands for automatic written inside. Hold the silver button down and twist the band so that the arrow printed on the camera lines up with the “M” and release the button.



Ring on the Lens Turned to Manual

- e) Just below this band, there are two words, “LIMIT” and “FULL” printed on the side of the lens, with a sliding button next to them. Slide the button all the way towards the camera so that the white dot lines up with the word “FULL.”



Lens Set to Full

f) Aperture: f-stop 5.6

With the Teleconverter, the aperture reads as f0 in ControlMyNikon, and the program does not allow the operator to change the aperture remotely. Therefore, the operator has to change it on the camera itself. On the Nikkor lens, just above where it connects to the teleconverter is a black band with “32, 22, 16, 11, 8, 5.6, and 4” printed on it. This band controls aperture size. Just above this band is a white line with a white dot above it and to the right of this line is a small switch with an orange dot next to it. To unlock the band, slide this switch so that the white dot on the switch is not in line with the orange dot. Now, spin the band until “5.6” lines up with the white line just below the white dot printed on the body of the camera.



Unlocked Position of the Switch that Controls the Aperture Band



Aperture set to f-stop 5.6

g) Image Quality: Fine

Push the button that says menu and then push the down arrow once to go to the shooting menu, which is indicated by a symbol of a small camera. Select “image quality” by hitting the small OK button, and then select “JPEG fine.”



Setting Image Quality

h) Picture Control: Vivid

Go back to the shooting menu and scroll down until you reach “Set Picture Control.” Select “Set Picture Control” and then select “Vivid”



Setting Picture Control

- i) The green LEDs surrounding the optic should be set anywhere from 28 to 31 Volts.
- j) The following can be set using ControlMyNikon
 Shutter Speed: 1/40
 ISO: 200
 Comp: 2.0
 White Balance: Fluorescent

2.2 Taking the Images

- 1) Look through the viewfinder on the camera and adjust the tripod so that the region of the optic you want to take a picture of sits squarely in the viewfinder.
- 2) Make sure that the image includes the reference point that is printed on the ring around the optic, as you will need this later to calculate the resolution of the camera.
- 3) Autofocus does not function properly when working on such a small scale, so in order to take a clear image, you have to focus the camera manually for each shot. This means that the camera cannot be entirely remotely operated as remote focusing relies on the Autofocus function. Therefore, to focus, first twist the lens while looking through the viewfinder until the optic comes into focus. Then open live view, zoom in as close as possible on a group of particles, and carefully make adjustments to the focus until the particles are as focused as possible. Refocus the camera every time you take a picture because the camera tends to go out of focus in between shots, usually from the movement of the camera shutter opening and closing.
- 4) Once you focus the camera, exit live view and click the button in ControlMyNikon that says “Shoot.”
- 5) To view the images, go up to the menu bar, select “Tools,” and then select “Image Browser.”

3 Processing Pictures

3.1 Calculating Pixel/Meter Ratio (**Do not need to do each time**)

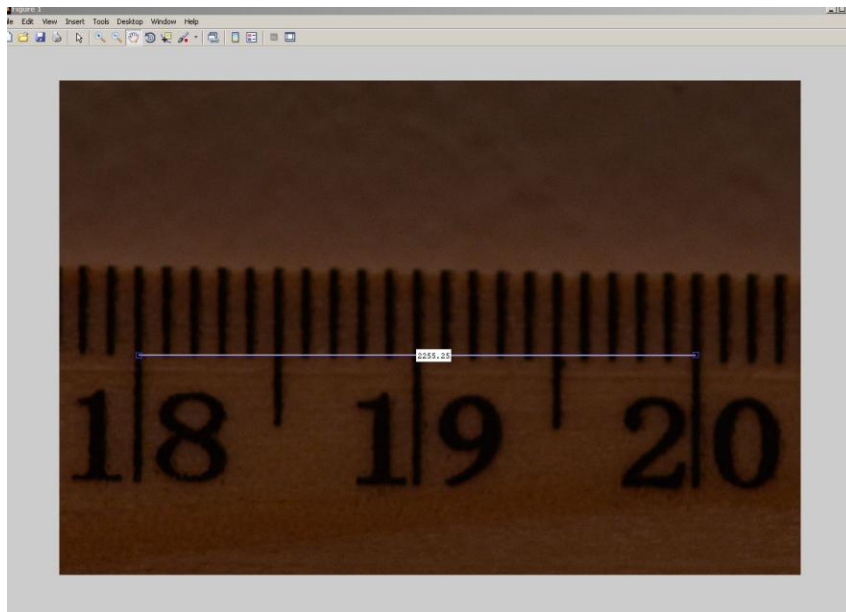
The code uses the pixel/meter ratio of the camera to calculate particle size, so when you change camera model or distance to the optic, these numbers need to be calculated and changed in the code as follows. Once you use a specific camera at a certain distance, you can simple reuse this ratio whenever you use that setup; you do not have to recalculate it every time.

- 1) Open the folder containing this image in MATLAB. Make sure that you have MATLAB Image Processing Toolbox Version 9.0 or greater because previous versions do not have the necessary functions for processing the images.
- 2) Open the MATLAB code titled *Matlab Code to Calculate Resolution* that accompanies this document on the DCC.

- 3) Type the name of the image into the code as shown below:

```
%read image
rgb = imread('name of image');
figure
imshow(rgb)
```

- 4) The code will generate a distance line you can move to measure a specific distance in pixels. Place the distance line across the length of the reference point referenced earlier. The small number it shows in the white box is the distance in pixels. Record this number.



Using Distance Line to Measure 20 mm in Pixels

- 5) Open the other code, *Matlab Code for Processing Images*, which accompanies this document on the DCC.
- 6) Convert the millimeter measurement of the reference point printed on the optic to meters and plug this number into the following line at the beginning of the code as the variable “lengthmeters.” Then plug the number of pixels across the reference point recorded from the previous step into the code as the variable “lengthpixels.”

```
lengthpixels= length of reference point in pixels
lengthmeters= length of reference point in meters
```

3.2 Running the Code

- 1) Ensure that you have the correct measurements for variables “lengthpixels” and “lengthmeters,” which you inserted either from step 3.1 or from a previous time you used the same setup.
- 2) Plug the diameter of the optic in meters into the following line of code as the variable “diameterofoptic”

```
Diameterofoptic= diameter of the optic in meters
```

- 3) Insert the name of the original image into the following line of code, which reads the image.

```
%read image
rgb = imread('name of image');
```

- 4) Next, adjust the threshold level the code uses. It is necessary to set this level by hand because each image requires unique settings; therefore, the automatic settings create inaccurate results. If the threshold is too high, MATLAB will read certain particles or sections of the particles as part of the background and color them black, effectively shrinking them or erasing them from the image. However, if the threshold is too low, MATLAB will read brighter parts of the background as particles and detect ghost particles that do not actually exist.

Adjust the highlighted value the following line to a number between 0 and 1 - the larger the number, the higher the threshold. 0.1 is usually a good place to start.

```
%greyscale to binary image
BW = im2bw(I, .1);
```

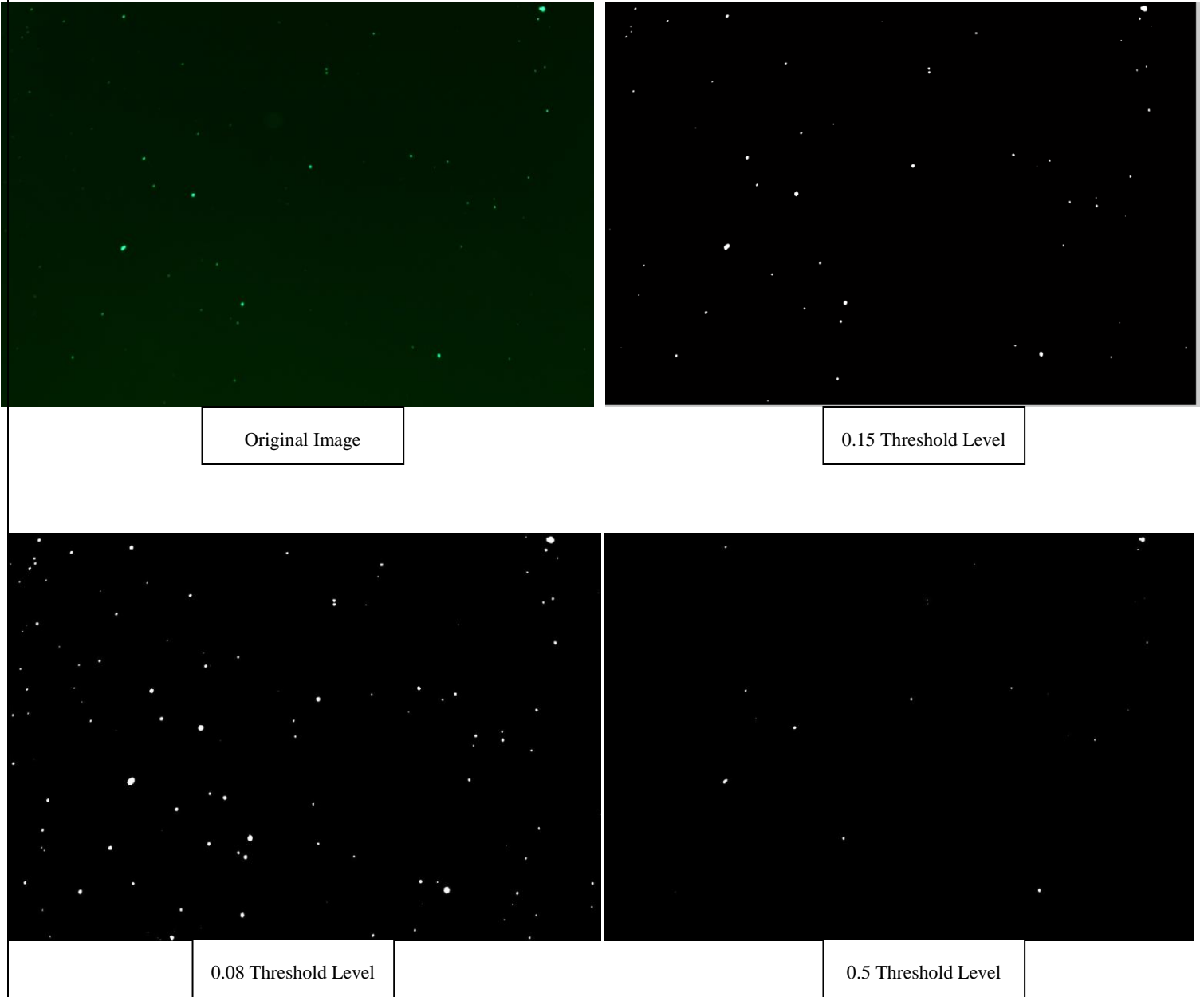
- 5) Adjust the upper and lower limits of what size particles you want to detect. Set the lower limit by changing the highlighted number in the line “lower=(ratioperim*(.00001))” to the desired particle area in microns. Set the upper limit by changing the highlighted number in the line “upper=(ratioperim*(.02))” to the desired particle area in microns.

```
%Remove remaining glare by eliminating particles over desired size
lower = (ratioperim*(.00001)); %lower limit of 10 microns
Z = round(lower); %rounds the lower limit number to nearest integer
LB = Z;

upper = (ratioperim*(.02)); %upper limit of 20,000 microns
Y = round(upper); %rounds the upper limit number to nearest integer
UB = Y;
```

- 6) Run the Code. You will reach a breakpoint in the code.
- 7) At this breakpoint, compare the processed image to the original image.
 - 1) Adjust the threshold level until you find the correct threshold level for the image; by trying several levels and comparing them to the original image, it is fairly easy to determine which one is right one to use.

Here is an example of an image thresholded at different levels:



By comparing the images processed at different levels to the original image, it is evident that the 0.15 threshold is the correct level for this image.

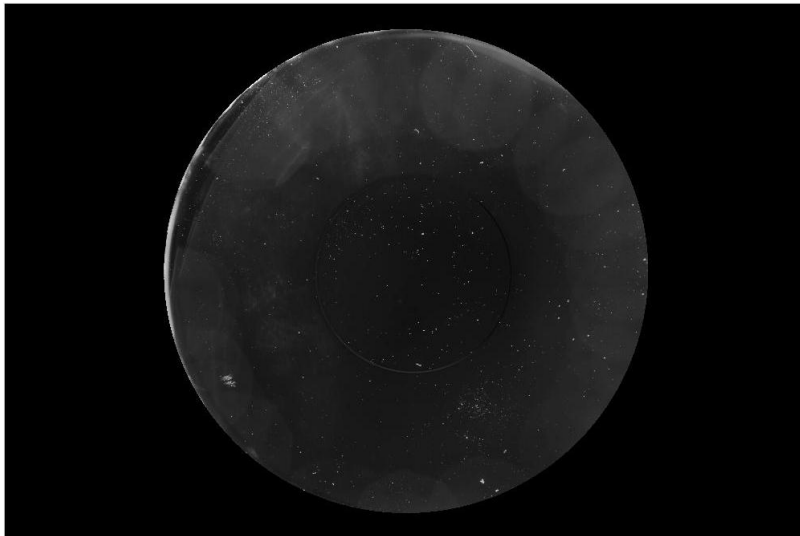
- 2) The code automatically crops the background out of the image, leaving only the optic behind. However, it makes the assumption that the optic is perfectly centered in the picture. If the optic is not centered, you can move the crop circle by adjusting the following line of code under the note %cropcircle.

```
ci = [((x+0)/2), ((y+0)/2)]
```

- By adding a number to x, the circle will move down by that many pixels
- By subtracting a number from x, the circle will move up by that many pixels
- By adding a number to y, the circle will move right by that many pixels
- By subtracting a number from y, the circle will move left by that many pixels.

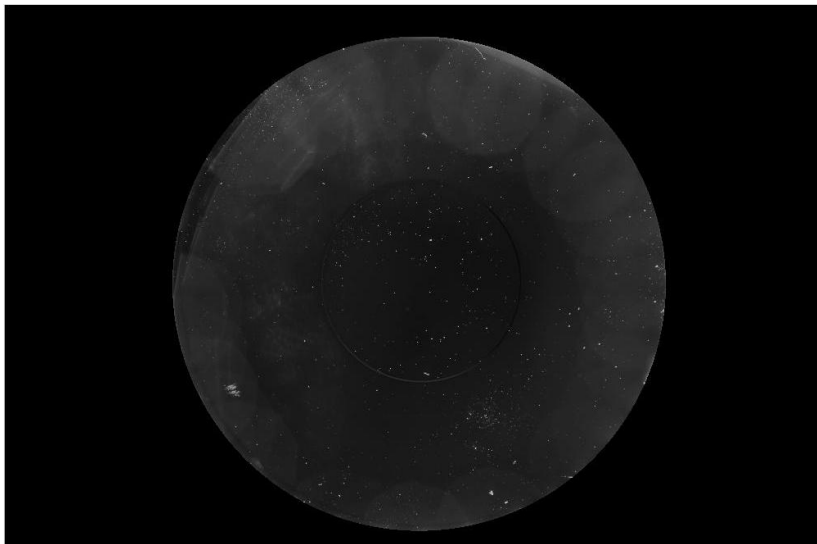
***Keep in mind that the x and y coordinates are in pixels, so adjust accordingly**

An example of an image that needs to have the crop circle adjusted slightly is as follows:



By changing the code so that it reads as follows, the circle is now properly cropped

```
ci = [((x+80)/2), ((y+80)/2)]
```



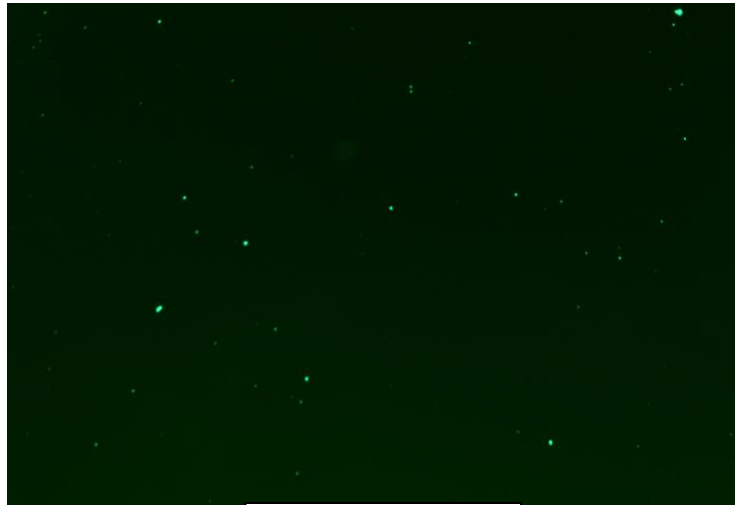
8) Rerun the code. Once the threshold level is correct and image properly cropped click continue to finish running the code.

9) The area and perimeter of the measured particles is located in the workspace next to the code under the variable names “area_microns,” “perim_microns,” “radius_microns,” and “diameter_microns.”

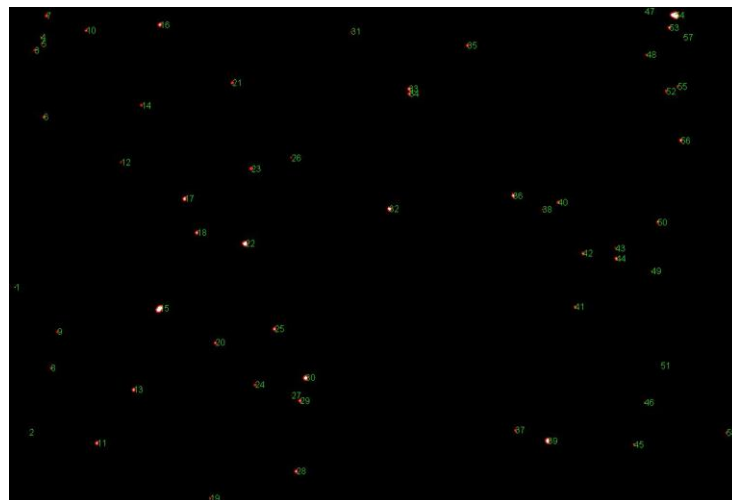
4 About the Results

The code measures the diameter (if the particles were perfect circles), radius (if the particles were perfect circles), perimeter, and area of each of the particles in microns. MATLAB also outlines the detected particles, which allows you to see which particles MATLAB detects, and therefore, catch any errors in detection that may occur, such as if it misses particles, detects ones that do not exist, or under or over estimates their sizes. It displays the grayscale image, the binary image before particles not within the desired bracket are removed, and the final processed image with the particles both circled and tagged with serial numbers.

Shown below are the original test image and the processed test image.



Original Image



Processed Image

MATLAB measures a particle's perimeter by drawing a line through the centers of its outer pixels; therefore, according to MATLAB, particles of the smallest diameter, which resolve as a single pixel, have a perimeter of zero. As seen in the image below, MATLAB cannot outline, and thus cannot find the perimeter of a particle only a single pixel in size.



The larger the size, the fewer particles there are of that same size, which occurs for several reasons. First of all, there could simply be more small particles than large particles on the optic, but it is also because there is a larger gap between particle sizes when they are smaller than when they are larger; in the test images, the smallest two sizes are 10.0066881um and 14.15159403um with a gap of 4.15 microns between them, while two of the adjacent larger sizes are 93.33617298um and 93.87105514um with only a 0.5 micron difference. The system resolves the particles that are in the gaps between sizes to the nearest pixel and places them in the category closest to their size, which inflates the number of pixels of that particular size. While this grouping does not mean that the overall number of particles detected is incorrect, it does mean that due to the large size gaps between the smaller particles, the true sizes of the smaller particles may be within a couple microns of their measurements. In the test image, the number of particles of the same size steadily drops from 229 at 10.0066881 um, to 72 at 24.51127986 um, to 34 at 42.45478208 um, and finally down to single digits around 60 um. Also, the gaps between the sizes went from multiple microns to one micron at around 33 um and to fractions of microns at around 50 um. Therefore, while the lower sizes should not be completely discounted, it is important to remember that the smaller the particle, the larger the margin of error regarding its exact size. This margin of error is unique for each setup because it directly connects to the pixel/meter ratio; the larger the ratio, the smaller the margin of error. Therefore, when the camera and distance to the optic change, the margin of error changes as well.

More information about the test results can be found in Progress Report 2 under document number [LIGO-T1400506-v1](#) and the document titled “Data for aLigo Particle Imaging: Resolution of Nikon D7100 with 2x Teleconverter” under document number [LIGO-T1400491-v1](#).

5 Related Documents

[LIGO-T1400474-v2](#): LIGO SURF Progress Report 1: Particle Imaging and Analysis for Advanced LIGO

[LIGO-T1400506-v2](#): LIGO SURF Progress Report 2: Particle Imaging and Analysis for Advanced LIGO

[LIGO-T1400491-v1](#): Data for aLigo Particle Imaging: Resolution of Nikon D7100 with 2x Teleconverter

[LIGO-T1400467-v1](#): Data for aLigo Particle Imaging: Resolution of Nikon D7100

[LIGO-T1300665-v9](#): The LIGO Particulate Evaluation Tool (PET)

[LIGO-T1500403-v1](#): Proof of Concept - Challenges: Matlab Code for Detecting Particles

[LIGO-T1500267-v4](#): Proof of Concept - Matlab Code for Detecting Particles