# Multi-Angle Mie Sizing Method
# User Manual

by
Stephen D. LePera

January 3, 2013

# Acknowledgments

This work is a result of the original work from my dissertation[2], and all credit and thanks given there are certainly still applicable.

I remain indebted to the love of my life Alison, who not only continues to tolerate me and tirelessly support our family, but also works long and thankless hours in the service of our community and is an inspiration for all.

Steve LePera
VPI #351
NSS 30805
EMT-I
KG4GLX
BS, MS, PhD

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

There are many interesting engineering flows where in order to improve performance, it is important to be able to measure the size and size distribution of droplets or particles; for instance characterization of a fuel spray or a fuel jet atomizing into the flow of air through an engine. Even though numerous excellent methods exist within the literature and many commercial systems are available which can give this information, it is likely impossible to have one system which is perfect for use in all occasions.

The work herein details a system based on planar multi-angle Mie scattering. Sizing information gained from this technique consists of a mean droplet diameter and droplet distribution estimates for every individual point within a planar area of interest. Only characteristics of the entire distribution at each location are measured, however the advantage is that when compared to some well known techniques, such as Malvern or Phase Doppler Particle Anemometry (PDPA), spatially accurate results for an entire flow-field may be obtained quickly and relatively economically.

## 1.2 Scope/Objectives

The User Manual describes the use of Mie scattering from an entire 2D plane to obtain information about the size and size distribution of particles in a typical spray. The planar method makes acquisition of data within a large field of interest possible, while the use of relatively inexpensive instrumentation makes the method available to a large number of research situations and facilities.

The result of this current work is:

- presentation of a relatively inexpensive method which can produce spatially accurate information about the size of droplets in a spray, and
- an exploration of the abilities and limitations of the developed method.

The following section briefly reviews sprays of the type concerned in this work, reviews past and current optical sizing methods, overviews the general Mie theory of scattering from small particles, and specifically analyzes Mie theory as it will relate to the planar scattering method developed within this work.

## 1.3 Particle Sizing Background

# Chapter 2

# The Experimental Method

## 2.1 Optical Method for Planar Measurement of Spray Characteristics using Mie Scattering Theory

The theory and background required to build the foundation of a planar (2D) optical spray characterization method was presented in Section 1.3. From that discussion it is clear that information about the mean droplet size, and the width and shape of of a spray size distribution is encoded within the distribution of intensities scattered from droplets within a particular plane of interest. With the Mie theory in hand, if the characteristics of a spray are known then this distribution of scattered intensities may be calculated.

The following sections address the "reverse problem" where scattered intensities at a discrete number of locations are measured, and then this information is used to deduce the size and size distribution of unknown droplets within the planar interrogation region.

The procedure used here is as follows:

1. The spray of droplets must be located within a specific optical setup and the geometry of all optical and experimental components must be well known, controlled, and placed in a specifically designed layout conducive to collection of the most information from the least number of discrete measurements. The region of interest is illuminated by a collimated, planar laser sheet.

2. Images from multiple angular locations are taken of the region of interest, either simultaneously or by moving a single detector during a continuous (stationary) experiment. A minimum of two measurements are required - one reference angle and one data angle. Additional data angles provide improvements in mean droplet diameter determination and add information about the droplet size distribution.

3. After the experiment is complete, the images are "registered" spatially into a common

coordinate system, corrected for perspective and detector non-linearity, and processed to produce "intensity ratios" between data angles and a reference angle. This ratio is therefore available at every point/area within the planar region of interest.

4. The measured intensity ratios are compared against known Mie theory scattering intensity ratios to deduce the size and size distribution of the droplets at all points/areas within the planar region of interest.

The following sections detail the "nuts and bolts" of the current technique. A complete step-by-step example of the above procedure is presented in Appendix B.

## 2.2 Optical Diagnostics

## 2.3 Data Processing

### 2.3.1 Computation of Mie Theory Scattering Functions

### 2.3.2 Image Processing

### 2.3.3 Sizing Calculations

# Chapter 3

# Results and Discussion

## 3.1 Introduction

The previous chapters presented the theory and practice of an optical method for planar measurement of spray characteristics using Mie scattering. Light scattering from an entire 2D plane was collected in a series of images to obtain information about the size and size distribution of particles within that slice of a typical spray. Custom built Matlab image processing and droplet sizing routines were presented and shown to produce spatially accurate information about the spray's size and distribution characteristics. The planar method enabled determination of sizing data within a large field of interest in a fast and inexpensive manner.

The questions that remain to be discussed are:

- What are the limitations of the method?
- How sensitive are the sizing results to the experimental parameters?
- Given the above two items, is the method applicable in a real-world experimental environment?

The following sections address each of these respective topics.

## 3.2 Limitations of Theory

## 3.3 Sensitivity and Limitations of the Method

### 3.3.1 Droplet Size Range

### 3.3.2 Quantity and Location of Angular Images

### 3.3.3 Signal Dynamic Range

### 3.3.4 Identification of Distributions

### 3.3.5 Angular Location of Detector

### 3.3.6 Setup Method Comparison

## 3.4 Conclusions and Recommendations

Sizing information gained from the optical technique herein consists of a mean droplet diameter and droplet distribution estimates for every individual point within a planar area of interest. The planar method makes possible the fast acquisition of data within a large field of interest, and uses relatively inexpensive instrumentation. This technique is a significant *advance in accessibility* to quantitative sizing information - droplet size information previously reserved only for researchers in possession of much more expensive diagnostic systems is available to "everyone."

The performance of the current planar optical method has been demonstrated in a real application, presented in the original dissertation [2] using experiments with three laboratory scale simplex atomizers. As presented therein, the method demonstrated the ability to measure droplets across the range of 5-50$\mu$m within +/-10% of known values, and in addition return an estimate of the shape and width of the correct size distribution at each location within the planar region of interest.

The necessary assumptions currently required to use this method, as presented within the current work, are summarized below:

- spherical, homogeneous, isotropic non-reflective droplets,
- normal or log-normal distributions,
- insignificant multiple scattering,
- monochromatic incident light (either un-polarized or linearly polarized),

- no active optical components between the incident light and the detector other than, if desired, a linear polarized filter.

This method is believed capable of measuring droplet distribution characteristics and means within a nominal range of $0.3\mu$m up to $150\mu$m and higher. The cost to build a system with this capability, a 75mW diode laser ($100), a RAW image format digital camera ($300), and assorted lenses and optical components ($500), are all estimated to be available for a total less than $1000.

Future advancements in the technique are already in progress, most importantly including an advanced pattern recognition sizing algorithm capable of quantitatively learning the droplet size and distribution information within a spray "at a glance," with no separate reference image required and significantly less sensitivity to the exact location of angular data images. Incorporation of an improved image-by-image, planar-location-specific signal strength weighting factor promises to reduce errors in poor dynamic signal ratio areas.

Additionally, publicly available machine vision algorithms are already capable of locating the position and orientation of a camera detector based on information within the camera detector's field of view. Combined with the pattern recognition algorithm, meticulous knowledge of the optical geometry might be eliminated.

In conclusion, while work up to this point has demonstrated that this basic method can be a successful addition to the list of many optical diagnostic sizing techniques, the results presented are potentially only the beginning. As with many things, there is always more to be done. Anyone wishing to contribute will be welcomed and should consult the first few sections of the example in Appendix B, where specific instructions about getting started have been included.

# Bibliography

[1] C. F. Bohren and D. Huffman. *Light Scattering and Absorption by Small Particles*. Wiley, New York, 1983.

[2] Stephen D LePera. *Development of a Novel Planar Mie Scattering Method for Measurement of Spray Characteristics*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, February 2012.

[3] Christian Mätzler. Matlab functions for mie scattering and absorption, version 2. Technical report, Institut für Angewandte Physik, Sidlerstrasse 5 CH-3012 Bern Schweiz, August 2002.

[4] Inc. The MathWorks. *www.mathworks.com*, r2008b edition, 2008.

# Appendix B

# Example - Optical Method for Planar Measurement of Spray Characteristics

The following sections present a step-by-step, Matlab-command-specific example of the entire multi-angle Mie scattering sizing method. It will be assumed that Matlab is installed and working. In addition, the example is executed using the linux version of Matlab; a few unix-specific commands are necessary within the sizing routines however it should be straightforward for a Windows user to make appropriate changes in these few cases. Interested Windows users should contact the author; addition of a Windows-specific tree or improved coding allowing unrestricted use on both platforms would be welcomed.

## B.1 Get the Code

### B.1.1 Downloading the Dissertation Code

The Matlab routines necessary to implement the "Multi-Angle-Mie-Sizing" method described in the original dissertation, "Development of a Novel Planar Mie Scattering Method for Measurement of Spray Characteristics," by Stephen LePera are hosted at:

https://github.com/leperas/Multi-Angle-Mie-Droplet-Sizing

The version directly referenced within the dissertation is denoted as Version 1, or v1 and is found in the download section.

WARNING! Absolutely NO development has occurred on this version after the dissertation was finished. It is highly recommended that if you wish to actually implement the technique, you should use the latest version from the repository! The old dissertation code is kept available ONLY to aid in understanding and to maintain the completeness of the dissertation.

### B.1.2 Downloading the GitHub repository

The actual GitHub repository is where the latest updates, capabilities, and new versions are found. Significant bug fixes, additions and improvements have already been made since the release of the dissertation, so it is highly recommended to use the latest updated GitHub version found at:

https://github.com/leperas/Multi-Angle-Mie-Droplet-Sizing

The README within that repository should contain the most updated information about the project.

The Pro Git book, written by Scott Chacon, is highly recommended, particularly if you are unfamiliar with using the **git** version control system.

### B.1.3 Installing the Code

Installation instructions are part of the code and are found in the file README.install. The most up-to-date install and configuration information will be maintained within that file.

### B.1.4 Contributing to the Code

Currently documentation consists of the original dissertation and this User Manual, and comments within the code. The goal of this User Manual is to contain the basic details of the method and up-to-date use and examples of the method as it evolves beyond the original dissertation work.

Users interested in contributing new or improved code or documentation should start by installing **git** and then getting the latest software from https://github.com/leperas/Multi-Angle-Mie-Droplet-Sizing. Although certainly not required, users are encouraged to contact the author in order to avoid duplication, and to coordinate efforts.

## B.2 Creation of the Scattering Cross Section Database

Before anything may be done, a scattering cross section database must exist or be created. Two files are included with the complete code download:

```
./multi_angle_mie_sizing/database_files/
        scattering_coefficients_water_sub_angs.mat
```

and

```
 ./multi_angle_mie_sizing/database_files/
          scattering_coefficients_water_all_angs.mat
```

Both files are for water, with index of refraction $1.33 + 0i$. The first is the data set that was used for the current work. This set only contains scattering coefficients at angles important to the sizing method, but has a high angular resolution and wide size parameter range. The second file was used to make a number of plots in the current work and contains data from angles 0-180°, but the data is only at limited resolution, and is not suitable for most sizing calculations.

If you plan to measure other substances, a new scattering coefficient database MUST be generated. This takes a long time; the following example will include commands suitable for sending the database creation into the background where it may run uninterrupted for days.

Change to the directory containing `create_Sx_database.m`, and open the routine in the editor.

```
>> cd ./multi_angle_mie_sizing/mie_m_code
>> edit create_Sx_database.m
```

For this example we will create a small (but useless) database. Set the database name (it will be created in the current directory), edit the maximum diameter, `max_d`, to be $50\mu$m, and the size increment `d_inc` to be $5\mu$m. Edit $theta = (pi/180) * [0 : 5 : 180]$, and set the index of refraction to $m = 1.448 + 0i$. Now run:

```
>> create_Sx_database.m
```

In a few seconds, there should be a new database in your current directory. If you were making a real database, it might take days, and you would want to run the process out-of-the-way in the background, like this:

```
>> system_string=strcat(...
           ['nice -n 8 /opt/matlab/bin/matlab -r "cd ''',pwd,...
           ''';create_Sx_database ;exit" &']);
>> system(system_string)
```

A status file is periodically updated while this is running - at any time just:

```
>> load status_scat_calc
>> [jj calc_time(jj)]
```

```
ans =

   37.0000    4.3135
```

The returned information is the current iteration number, jj, and the amount of total time taken up to this point, calc_time.

If the database was created correctly, you should be able to run this and see the same output:

```
>> load scattering_coeff_database.mat
>> A

A =

1x37 struct array with fields:
    S1
    S2

>> B

B =

        m: 1.4480
        x: [1x10 double]
    theta: [1x37 double]
```

For any new materials used, a new scattering coefficient database is required. In addition, a new sizing look-up database (discussed in Section 2.3.3) must be built from scratch before any sizing calculations may be started.

## B.3 Make an Image Data Set

### B.3.1 Test the Intensity Function

Before jumping in and making images, it is a good idea to test the intensity function, Irr_int.m

Change to the directory containing Irr_int.m. Make the scattering coefficient database variables global, and load the water database that came with the downloaded code package as shown:

```
>> cd ./multi_angle_mie_sizing/mie_m_code
>> global A B
>> load ../../multi_angle_mie_sizing/database_files/
               scattering_coefficients_water_sub_angs.mat
```

Define the input variables (note calculation of size factor, x):

```
>> diameter=25.02;
>> sigma=10;
>> wavelength=514.5;
>> x=2*pi*(diameter/2)*1e3/wavelength;
>> sigx=2*pi*(sigma)*1e3/wavelength;
>> PDF_type='single';
>> theta=139.03*pi()/180;
>> phi=0*pi()/180;
>> half_cone_ang=0.2*pi()/180;
>> gamma_ref=pi()/2;
>> xi=pi()/2;
>> method='full';
```

Run the routine:

```
>> [intensity I Q U V matches]=Irr_int(x,sigx,PDF_type,...
         theta, phi, half_cone_ang, gamma_ref, xi, method)

intensity =

    0.3261


I =

   1.0e+03 *

         0         0    9.0955         0         0
         0    9.8185    9.8185    9.8185         0
    9.9671    9.9672    9.9673    9.9672    9.9671
         0    9.5866    9.5866    9.5866         0
         0         0    8.8292         0         0


Q =
   1.0e+03 *

         0         0   -9.0955         0         0
         0   -9.8184   -9.8185   -9.8184         0
   -9.9670   -9.9672   -9.9673   -9.9672   -9.9670
         0   -9.5866   -9.5866   -9.5866         0
         0         0   -8.8292         0         0


U =

         0         0    0.0000         0         0
         0  -25.8268    0.0000   25.8268         0
  -52.5150  -26.2580    0.0000   26.2580   52.5150
         0  -25.2935    0.0000   25.2935         0
         0         0    0.0000         0         0


V =

     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0


matches =

    2.4260  152.6527
```

Your output should match the above.

### B.3.2 Real Images

The PNG format uses lossless compression and is an excellent choice for use with this method, as most proprietary RAW or other high-color-depth image types are easily converted with free tools such as `http://www.imagemagick.org/`. In addition, the format enjoys full support from MATLAB.

Two possible conversion routes for real images from a Canon S90, for example, both use the `ImageMagick` program `convert` and are demonstrated below.

```
Conversion route #1:
(uses imagemagik which calls ufraw to do conversion)

$ convert cr2:IMG_0595.CR2 png:image.png

Conversion route #2:
(uses dcraw first, then imagemagik)

$ dcraw -4 IMG_0595.CR2

    then

$ convert IMG_0595.ppm image.png
```

The above should only be considered a starting point - currently very little testing of real high-bit-depth images has been finished. Be VERY wary of the effects of any conversion on the linearity of your images.

### B.3.3 Make Simulated Data Images

It is useful to generate a simulated data set in order to test all the routines, and also to allow much of the computational effort in an actual experiment to be completed ahead of time. The simulated image routine is capable of producing 8-bit and 12-bit color depth (per channel) JPEG images, and also 8-bit and 16-bit color depth (per channel) PNG images.

For simplicity, the simulated images contain both data and registration information. This is reflected in the .ini files by assigning the same image both as data and as registration. In practice, it is almost impossible to get good data signal-to-noise images and at the same time capture a good image of the registration points, so two separate images taken from the exact samem location, but with different exposure/lighting, are used.

The function for making simulated images, `make_data_images.m`, is controlled by an initialization data file. If no initialization file exists, the routine will create an example file. The newly created file may be edited as desired.

For this example, change to the directory containing `make_data_images.m` and and run the routine as shown:

```
>> cd ./multi_angle_mie_sizing/image_creation/
>> make_data_images('new_initialization_file.ini')
```

The GUI asks where to save the images; for the purposes of this example create a directory called `test_images` and let the routine create the initialization file there. In practice this directory and filename may be at any desired location.

The function will now create the initialization file, and ask if the user would like to create images. Select "No" if changes to the configuration are needed. For this example, select "Yes."

The GUI will ask for a filename and location to save the image files. The filename supplied will be used as the base filename for the image set; all images will be in the directory selected from the GUI. For the example, keep the default `save_file` name and directory and select "Save."

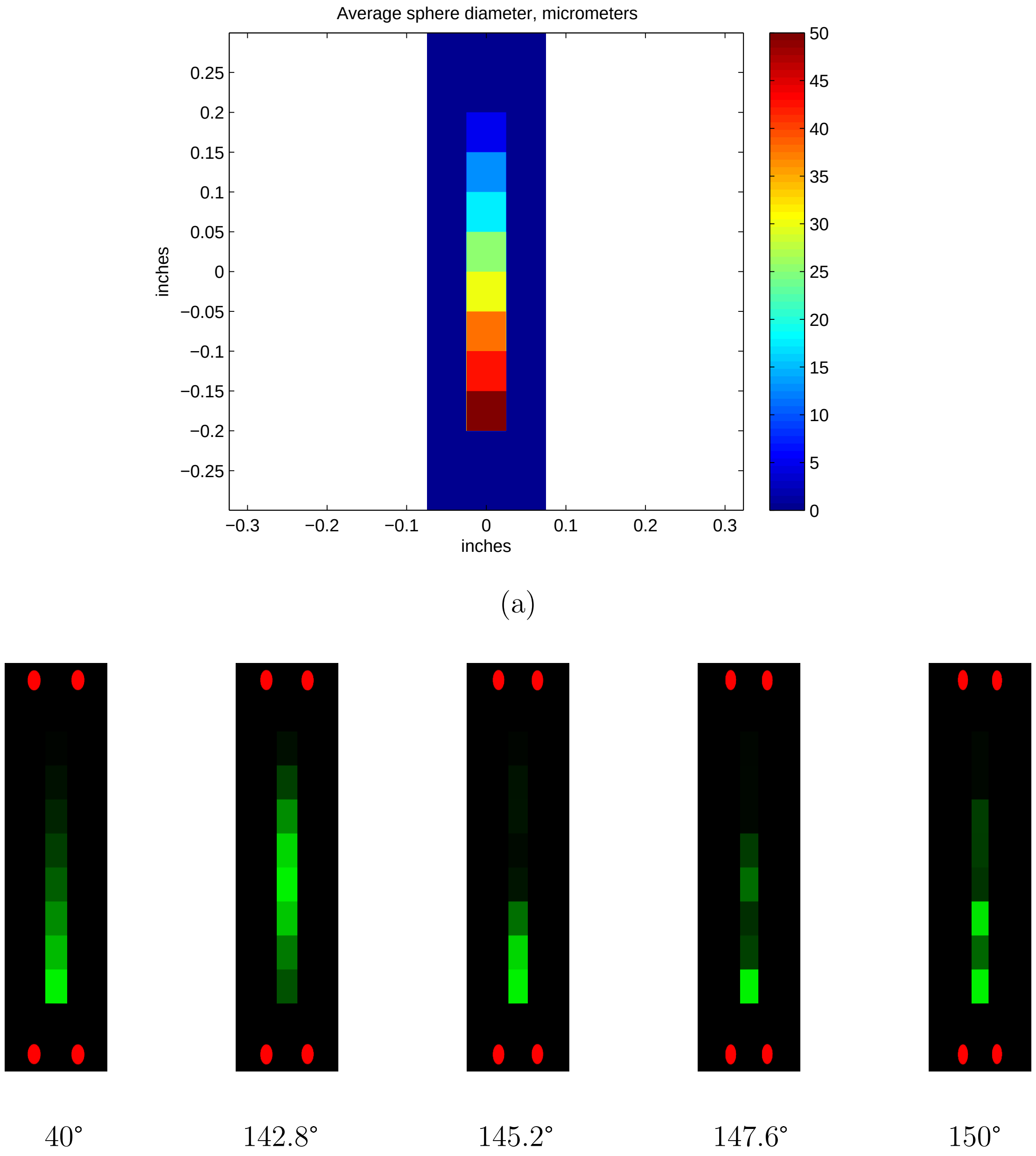
Using the default configuration as above (without any changes) will make a data set based on setup Method #3 (MTD3), that consists of one reference image at 40°and six data images evenly spaced between 138-150°. Droplets will cover a range between 0.5-50$\mu$m with a log-normal distribution and $\sigma = 10$. The default configuration uses a linearly vertically polarized laser as the incident light source, and a linearly oriented polarizing filter at the detector camera.

Eight figures should open. The first seven figures are the images described above, the eighth image is a contour plot showing the simulated sizes. The data images are saved in the chosen directory as 16-bit PNG, however the contour plot is not saved. If you want to keep it then save it manually (but remember that all data required to recreate the contour plot is automatically saved in the chosen directory as .mat files). For each image there is a `''dg''` file and a `''In''` file, containing respectively the diameter information and intensity information. There is also one `''info''` file which contains all the configuration parameters used by `make_data_images.m` to produce the simulated data set. In addition, a template image processing initialization file is created in that directory. The images created should look similar to those in Figure B.1. Most printouts and screens do not have the dynamic range to show the brightest and darkest parts of the image together, however the PNG images created in this example have 16-bit color depth per channel (65536 possible intensities) and have valid pixel values for the entire size range, even in squares of the image that appear "dark" on the screen or in print. In this example, the value of the darkest region is 83, compared to the 65535 highest possible value.

The initialization file may be edited to create customized data sets. The options for this are documented earlier in Section 2.3.1 and also sparsely documented within the code. Take a few minutes to read the configuration parameters available.

To create a data set based on an arbitrary initialization file, supply the full file name and path and re-run the routine, for instance:

```
>> make_data_images('./test/arbitary_initialization_file.ini')
```



(a)

(b)

Figure B.1: (a) Plot of the simulated mean droplet sizes within the images. From top to bottom of the image, 0.5, 6.7, 13, 19, 25, 31, 38, 44, and 50$\mu$m. (b) Simulated images at a few representative angles.

## B.4 Image Processing the Data Set

A data processing initialization file, `save_file-image_processing.ini`, was created as part of the simulated data set in the previous section, and is located in the directory with those images. The contents and use of this file is well documented in Section 2.3.2.

The contents may be left alone, or edited by hand. For this example, open the file and scroll down to the {Output} section. Change the parameter `output_eps_figures = 'Yes'` so that some images will be generated during processing. Save changes to the file.

Change to the directory containing the function `processor_f.m` and run the routine:

```
>> cd ./image_processing/sub_functions
>> processor_f(0)
```

The processor function will open a GUI; select the data processing initialization file, `test-image_processing.ini`. Eleven figures will open, and a prompt to process the next angle will appear; select "No" and the function will close, but the figures will remain.

The following output is on the screen, and this same information has been saved into the log file, `./test_images/processed_output_sim_data/sim_data_processing_log.txt`:

```
Figure origin [0  y-offset  z-offset] [ 0  0.000000  0.000000 ]
**** Reference Image ****
Percentage saturated pixels: 0.000000
Actual data block limits, Left, Right, Top, Bottom:
                        [-0.024804 0.024804 0.250196 -0.199869]
Block height: 0.450065  width: 0.049608
Actual super pixel square side length: 1.222222e-02 inches.
Single pixel area: 5.168952e-07 square inches.
Number of actual pixels averaged in a superpixel: 289
Total super pixel area: 1.493827e-04 square inches.
```

The processor has just worked on the reference image. Note that it's good to check that there are no saturated pixels. If a saturation warning occurs, check to be sure the saturation is not in the data part of the image. If other areas outside the data area are saturated, it may effect signal-to-noise but probably won't wreck the data processing.

In the processing initialization file, a sub-block of data was requested. The exact size of the block is returned; due to the discrete number of pixels in the image this is unlikely to be exactly what was requested. The same is true for the requested super-pixel dimensions.

Eleven figures are opened. Figure 1 is a histogram of the reference image; there is not a high number of "saturation" pixels (would be bunched at the right side of the plot) which is good. Only the data areas have high pixel counts.

Figure 2 illustrates the image registration information. Check that what is shown are the dimensions of the data region.

Figure 3 is original image, Figure 4 is the perspective corrected "flat" image, and Figure 5 is only data from the color channel containing data, in the example that is the green channel.

Figure 6 shows the image area, with the image values converted to the 0-1 range as the "brightness matrix" and Figure 7 is the same information, but just showing the sub-block of data requested in the initialization file.

Figure 8 and Figure 9 are the "sister" images of Figures 6 and 7, showing the application of the exposure time. Because the exposure time of the reference image in this case was close to one (0.988), very little difference is observed.

Figure 10 shows the location of the super-pixels as a "+" overlaying the super-pixel contour values. Last, Figure 11 shows the same super-pixel values, but uses the actual shape of the super-pixel. The fill-color of the super-pixels represents the value of the averaged super-pixel.

Open the data processing initialization file, `test-image_processing.ini` again and change the `output_eps_figures` value back to `'No'`, then run the processor again, but with no pauses by setting the input parameter to "2", like this:

```
>> processor_f(2)
```

The routine, now that no images are required, will very quickly re-process the reference and the other 6 data images. The on-screen output is the same in the example; in a real data set it is good to look at each image and be sure each image has processed correctly. The intensity ratio data file has been created, `sim_data_ratio.mat` in the directory with all the other processing output, `./test_images/processed_output_sim_data/`. The sizing processing initialization file is also saved there. The data set is ready for size processing.

The data processing in this example goes very fast, but large images and large data sets may take longer. If many data sets need to be processed, the GUI interface tool `start_image_processing.m` is useful. Simply create a text file with the full path to every data image processing initialization file that needs to be processed, as many as needed, as below:

```
/full_path_to/test_images/test-image_processing.ini
/full_path_to/other_test_images/other_test-image_processing.ini
/full_path_to/more_test_images/more_test-image_processing.ini
```

Edit the top of `start_image_processing.m` to reflect the number of processors available, and run the routine. After using the GUI to select the above file with the list of filenames, the routine will start as many processing jobs as there are processors. When each job finished, the routine will start another process until all the files have been processed.

## B.5 Size Processing the Data Set

The last step is to run the size processing routine. The sizing processing initialization file was created and stored in the same directory with the other image processing results. Options within the sizing initialization file are covered within Section 2.3.3, however the file created in this example may be used as-is.

If not installed, the sizing look-up database must be built from scratch (which takes a long time). The use and building of this database is discussed in Section 2.3.3, but for this example just be sure that the provided sizing look-up database has been installed in the location shown:

```
 ./multi_angle_mie_sizing/database_files/sz3_data_file_water.mat
```

Now, change to the directory with the size processing function `analyze_sizes_f.m`, and run the `analyze_sizes_f.m` function.

```
 >> cd ./multi_angle_mie_sizing/sizing_functions/sub_functions
 >> [best_mean best_std best_sigma best_dist match_fraction conf_num]=...
                      analyze_sizes_f(0)
```

Use the GUI to select the sizing initialization file created by the data processing routine, `sim_data_sizing_ini.ini`. After 30 seconds or so, the output file `.../test_images/processed_output_sim_data/sizing/output.mat` will be created, and the following returned to the screen:

```
 best_mean =

  305.3359
  268.6839
  232.0355
  189.2858
  152.6546
  116.0216
   79.3848
   40.8050
    3.1657


best_std =
    0.0352
    0.0187
    0.0228
    0.0073
    0.0072
    0.0136
    0.0274
    0.0468
    0.1301


best_sigma =

    10
    10
    10
    10
    10
    10
    10
    10
    10


best_dist =

    'logn'
    'logn'
    'logn'
    'logn'
    'logn'
    'logn'
    'logn'
    'logn'
    'logn'


match_fraction =

    1.0000
    0.8333
    1.0000
    0.8333
    1.0000
    0.8333
    1.0000
    1.0000
    1.0000


conf_num =

    1.0000
    0.7129
    1.0000
    0.6554
    1.0000
    0.8565
    1.0000
    1.0000
    1.0000
```

The best_mean is the final size output at each location in the region requested by the sizing initialization file, in this case the middle "column" of the image. The output is summarized as below (after converting the size parameter back to diameter).

| Mean, $\mu$m | 0.5 | 6.7 | 13.0 | 19.0 | 25.0 | 31.0 | 38.0 | 44.0 | 50.0 |
|---|---|---|---|---|---|---|---|---|---|
| std dev | 0.04 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.01 |
| $\sigma$ | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 |
| Distribution | logn | logn | logn | logn | logn | logn | logn | logn | logn |
| CN | 1.00 | 1.00 | 1.00 | 0.86 | 1.00 | 0.66 | 1.00 | 0.71 | 1.00 |
| Error % | 3.69 | 0.07 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 |