Side Notes: Multiple thresholds are needed in order to measure all possible crystals (typically 1-4 thresholds is suitable enough. DO NOT do more than four because this can increase error in analysis performance and it gets too complicated. The images should be clear enough to do this). Thresholds are taken from Northern Eclipse software.

(1) Read raw table from threshold text file

↓

(2) Select desired columns from the first threshold file. Turn them into array

↓

(3) Sort for roundness for the first threshold text file (roundness > 0.7)

Side note: Eclipse only has inverted roundness (1/R). So you need to convert that into Roundness

↓

Repeats steps (1-3) for second threshold data now

↓

(4) Now, we want to remove duplicate crystals from the second threshold that were already measured in the first threshold. To do this we need to check their centroids in the x and y direction (these are the positions of the crystals in the (x,y) direction). I found that this number does not really change as much over time. In fact, I found it, manually, to be less than a 0.2 micron difference, so this will be our set condition. In other words, the difference in the x,y positions of the same crystals in the first and second threshold should be less than 0.2 microns. So now, we will choose the duplicate crystals from the second threshold that were present in the first threshold. Make the duplicate crystals into its own array.

↓

(5) Now that we have our duplicate crystals, we want to choose the non-duplicates from the second threshold. To do this, we are going to use the object number columns from our second threshold (after roundness sorting) and compare it to that of the duplicate crystal object number column we just made.I use the is not member method from Matlab to choose the crystals that do and do not match. These object numbers will be stored in an array with 1's and 0's where 0 is a match and 1 is not. We will call this 0,1 array C.

↓

(6) We want to only select from C those elements that have a 1. These are going to be our non-duplicate crystals. To do this we are going to index our non-zero elements from C. Indexing means we assign a number to each element in our array in sequential order. Ie: The first one in [1 1 0 0 1] has an index of 1, the second 1 has an index of 2, and the third 1 has an index of 5. This is done by

using the find function. This will not include elements that have a 0. We will make these indices into an array called ind and rename it to x for better convenience.

↓

(7) I make a table to select only the rows with the indices of x from our second threshold data table (after roundness sorting). Elements were renamed and this will be turned into a new array called non-duplicates. This is now our table of non duplicate ice crystals from our second threshold that contains all necessary data now (ie: radius, diameter, centroid positions, etc..)

↓

(8) We now combine our crystals from the first threshold and our non duplicate crystals from our second threshold. These crystals will then be used in the same way so that we do not include crystals from our third threshold that are already in the first two thresholds. We repeat processes (1-7) depending on how many thresholds you have.

↓

(9) Conduct final calculations (find ice volume fraction and radius data) after combining all data from all thresholds.

MatLab supplemental:
https://www.mathworks.com/help/matlab/ref/double.ismember.html
https://www.mathworks.com/company/newsletters/articles/matrix-indexing-in-matlab.html