

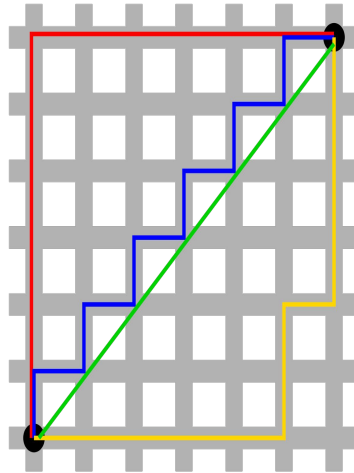
# Floc Size and Count App



Discusses the most recent developments in the development of the  
AguaClara Floc App.

More information is available at  
[https://confluence.cornell.edu/display/AGUACLARA/Floc+Size+and+Count  
+Application+Software](https://confluence.cornell.edu/display/AGUACLARA/Floc+Size+and+Count+Application+Software)

## Properly scaled distances yield useful data



Apps and Algorithms | Symposium Spring 2017

Christian:

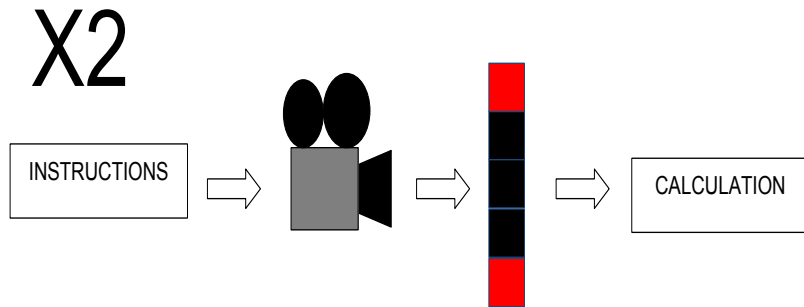
The original software provided by Casey came with a fixed scale to convert from pixels to actual distance (3.75 pixels per micrometers). This factor, however, depends on two variables, the camera and the distance it and the objects of interest, and must therefore be adjusted for different apparatuses. Setting this variable to an incorrect value would produce only relative data in some non-standard unit. With a proper scaling factor set, the Floc App would be able to produce reliable data in micrometers, which would be more conducive to analysis by both AguaClara researchers.

Caption: The image describes Manhattan distance, which is the simple sum of the horizontal and vertical components of each path from one point to another. The pythagorean theorem gives the shortest distance from one point to another. Thus, these two measures of distance are different, but one is more useful than the other. In the case of the floc app, the original program returned the proper floc diameters for a specific lab setup, but this data would be meaningless if a different setup is used.

Source:

[https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Manhattan\\_distance.svg/1200px-Manhattan\\_distance.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Manhattan_distance.svg/1200px-Manhattan_distance.svg.png)

# The scaling factor must be calculated in an intuitive way



Apps and Algorithms | Symposium Spring 2017

Christian:

One way of confronting this issue is to have the user manually enter this number. We decided against this approach because it would require too much knowledge about the camera from the user. Rather, the team decided to create a subVI that would allow the user to set this factor in a more natural way. This subVI would be activated either upon a user's request or if the user wished to run the program, but fails to set this variable ahead of time:

In this subVI, a dialog box appears telling the user what he/she must do to set the variable properly. In the case of the scale factor, he/she must lay some ruler flat and take a picture of it with the system's camera. After the OK button is clicked, the user will be able to see a live video-feed from the camera and must click a button in order to take the proper picture. This will trigger two popup screens to be generated, one asking the user to select two points on the photo and another asking for the distance (in mm) between the two points. Finally, this data is used to calculate the desired output (scale factor in this case).

Because the dimensions of the floc and the area in focus are both quite small, we can ignore the effects of distance on a floc's perceived size

Bonus Content (Slide 10):

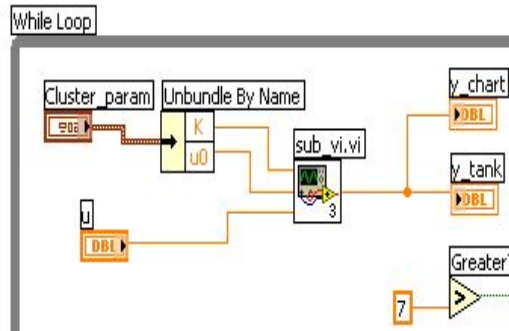
This same subVI is also used to calculate the volume of the area in focus. This data is useful because different setups with larger "focus areas" would be able to see larger flocs, therefore, normalizing the data may be necessary. The major difference between calculating this volume and the scale factor is in kind of photo that needs to be taken; calculating the volume requires the ruler to have portions in and out of focus and to select one point that is in and another that is out. The calculations also differ.

Caption:

These images show the step-by-step process by which a user would change the scaling factor and focus volume used by the program.

## Default Directory and SubVIs make program user-friendly

- No need to type folder name with default directory
- SubVIs make code more user-friendly



Apps and Algorithms | Symposium Spring 2017

Anthony and Deniz:

In order to allow the user of the floc app to easily pick a folder for the analyzed images to be stored, the default directory VI was used in the code. It works for all types of operating systems, and chooses a default folder so that the user does not have to manually type one in or browse for one.

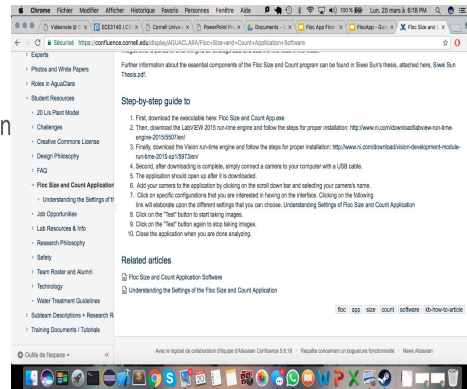
The default directory VI makes the executable version of the program more user-friendly, but the team also wanted to focus on making the code cleaner and easier to understand. As a result, many subVIs were created as a means of decluttering the code. In addition, more comments were included to make it easier to understand what the code does at each step in the process.

Caption: The image shows an example of LabVIEW code with a subVI. The subVI makes the code more visually appealing, and if a person looking at the code is interested in seeing what it does, he/she can simply click on it. However, more often than not people are more interested in seeing the overarching program and expect that subcomponents are contained into subVIs, just as different classes are created in Java object oriented programs rather than putting everything in one class.

# Run-time Engines must be downloaded for floc app



- LabVIEW 2015 run-time engine
- LabVIEW Vision run-time engine
- Updated the wiki with the information



Apps and Algorithms | Symposium Spring 2017

Deniz:

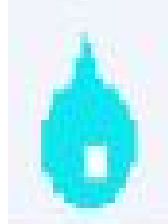
The floc app was not working properly on many computers in the lab, and the team could not figure out the problem for quite some time. Many different run-time engines were installed, until it was determined that the LabVIEW 2015 run-time engine and LabVIEW Vision run-time engines must be installed for the app to function smoothly. The AguaClara wiki page has been updated to include this information, making it much easier to simply follow the instructions given and utilize the floc app.

Image:

The updated wiki page with the required runtime engines for the executable program.

## Installed the program for another subteam to use

- Installed the software for the filter constriction team.
- Had some connectivity problems.
- Overall, good response so far.



Deniz:

We successfully uploaded the app for the filter constriction team. While the software worked with a simple webcam, we had problem running it with the professional high resolution. The program did not recognized the camera for reasons still unknown to us. We will be investigating in the coming days.

Image 1:

The water drop which is the symbol for the executable of our program.

Image 2:

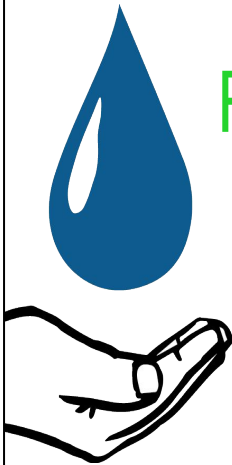
The camera that could not be found by the program.

# Future Work

- Finish Actual Distance SubVI
- Get more researchers to use the floc app
- Solve the camera problem
- Make updates from feedback



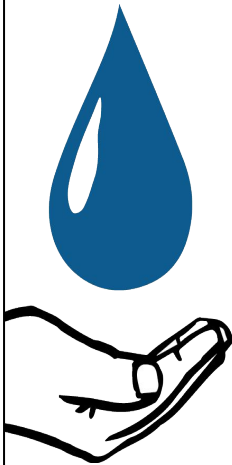
# Questions and Recommendations



Deniz Yilmazer  
BS Computer Science  
dy223@cornell.edu

Anthony Verghese  
BS Computer Science  
akv26@cornell.edu

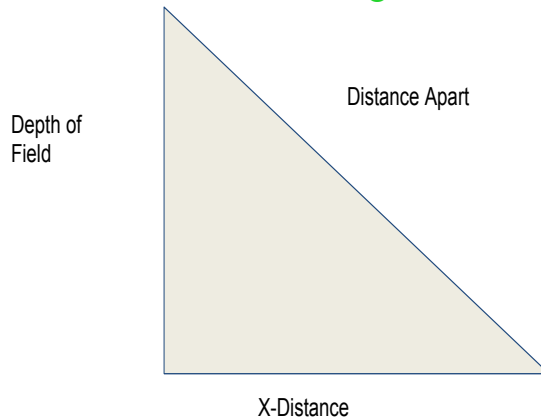
Christian Edward Rodriguez  
BS Computer Science  
cer95@cornell.edu



# Appendix Slides

From this point on, add any slides with figures that will help support your thesis. You might pull these figures from your Final Report.

## The volume in focus provides useful data for normalizing data



Apps and Algorithms | Symposium Spring 2017

Christian:

In addition to knowing the conversion factor from pixels to micrometers, our application should also know the volume within which flocs are in focus. With a larger volume in focus, the Floc App would be able to analyse a larger number of flocs with larger Ferret diameters. These differences would make comparisons of two sets of data useless if two experiments relied on two distinct cameras. To make such comparisons more informative, we decided that our analysis should consider not only the diameters of each floc, but also the total number of flocs analysed and the “focus volume.” The floc count is already tabulated by the Floc App, but the volume calculation requires the development of a new subVI.

The same subVI used to calculate the conversion factor from pixels to micrometers could also be used to calculate the volume of the area in focus. The major difference between calculating this volume and the scale factor is in kind of photo that needs to be taken; calculating the volume requires the ruler to have portions in and out of focus and to select one point that is in and another that is out. The calculations also differ.

To perform this calculation, the program must find the depth of field, multiply by 2, and multiply this product by the length and height of the camera lens. The height and the width are obtained directly from the camera

The depth of field, however, requires the use of the Pythagorean Theorem. The horizontal distance between the two points is calculated, squared, and then subtracted from the square distance between the two selected points.

The result is then rooted and converted appropriately.

Caption:

In order to determine the depth of the field , which is needed for finding the volume in focus,

the program uses the Pythagorean Theorem. The x-distance is calculated using the two data points posted by the user whereas the Distance Apart is given by the user.