AguaClara

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
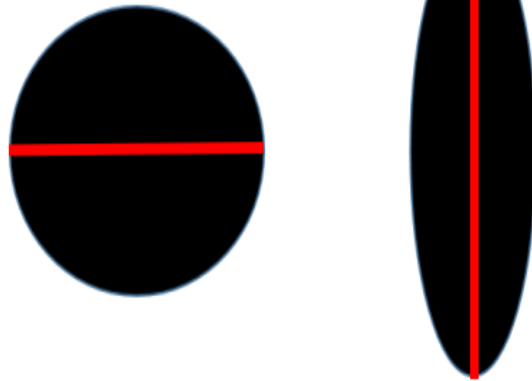
Anthony:
Image retrieved from clipart

Caption of Slide: This slide introduces the main goals of the floc app team and how it serves as a valuable resource for other subteams focused on research with flocculators.

Summary: The Floc App team is very important to Agua Clara. Developing software that can take in images and analyze them to determine the size and number of flocs is important for many of the research subteams in Agua Clara. Thus, the short term goal of the team is to help other Agua Clara subteams that are researching flocculation. However, helping research groups is not the only goal of the floc app; the long term goal of the floc app is to serve as a tool for on-site workers in Honduras so they can quickly determine whether their plant is functioning properly. By simply running the program, they can get an accurate estimate of the average size and number of flocs at specific points in the filtration process, thus making our plants more efficient and easier to check up on.

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.

Furthermore, the program that we were given originally measured the maximum feret diameter of flocs, or in other words, the longest distance between two points in a floc. This measurement, however, only accurately represents a circular floc. The more elliptical a floc gets, the more this size measurement exaggerates the flocs size. To deal with this, we decided to find an "average" diameter of a floc. For each floc, we simply took its measured area (images are 2-dimensional) and calculated the radius of the mass assuming that it was a circle.

Caption: Using the max feret diameter as an approximation for size exaggerates the size of thinner flocs, more elliptical, flocs.

The scaling factor must be calculated in an intuitive way

X2

INSTRUCTIONS ⟹ ⟹ ⟹ CALCULATION
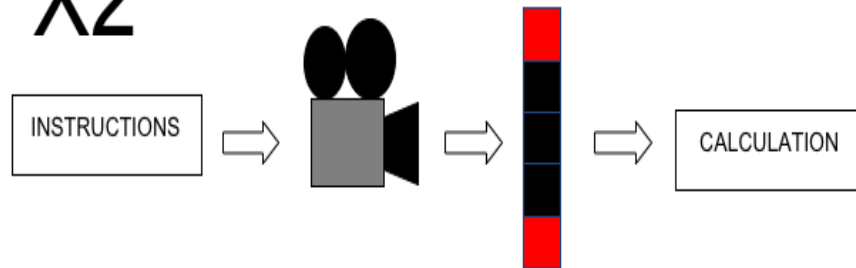
Apps and Algorithms | Symposium Spring 2017

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 or 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 a 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.
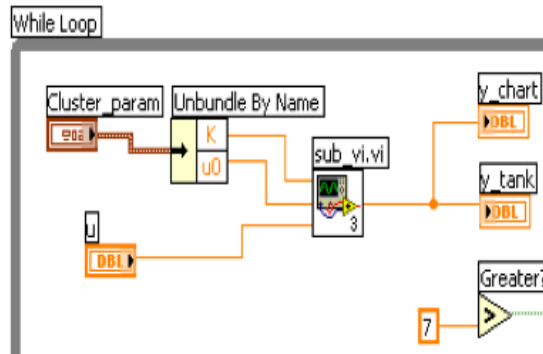
Christian:
In addition to providing Information to the user via a graph, the team decided that the user may want to know the measured (average) diameter of each individual floc that was measured. To accomplish this, the team decided that the Floc App should generate an excel file with this information at the end of each test. The output of such a file is shown on the screen. The team decided to allow the user to decide where to place this file after every test, but we have saved the user the trouble of naming the file by creating a default name of FlocAppInfoDD-MM-YY where D stands for day, M stands for month, and Y is year.

For every floc captured, we store both the average floc diameter as well as the time at which its data was recorded onto a row. We are currently unaware of any means by which to scale the sizes of each column without outputting an excel file as opposed to a csv file. The team has decided not to do this in order to support the use of multiple programs in analysing the data (e.g. Libre Office).

Default Directory and SubVIs make program user-friendly

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

While Loop

Cluster_param | Unbundle By Name

K
u0

U

sub_vi.vi
3

y_chart
DBL

y_tank
DBL

Greater?
>

7

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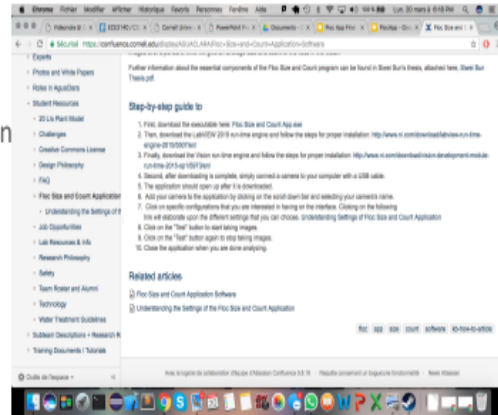
The default directory VI makes the executable version of the program more user-friendly, but the team also wanted to focus on making the code cleaner and easier to understand. As a result, many subVIs were created as a means of decluttering the code. In addition, more comments were included to make it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

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.

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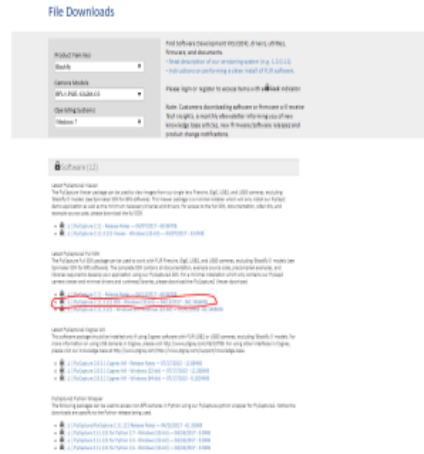The water drop which is the symbol for the executable of our program.

Image 2:
The camera that could not be found by the program.

Anthony: Connecting the PointGrey BlackFly camera to the lab computer proved to be very difficult. When all of the necessary software was downloaded and installed, there was a recurrent issue in NI MAX saying that there was a problem with the XML file with the camera. The team contacted FLIR and LabVIEW and eventually figured out, with the help of Casey, that the XML file should be deleted and then the camera can properly be shown on NI MAX, allowing for connection to the floc application.

URL of picture: https://www.ptgrey.com/support/downloads
Caption of left image: Firmware that must be downloaded to be able to connect PointGrey camera to floc application
Caption of right image: Software that must be downloaded to be able to connect PointGrey camera to floc application
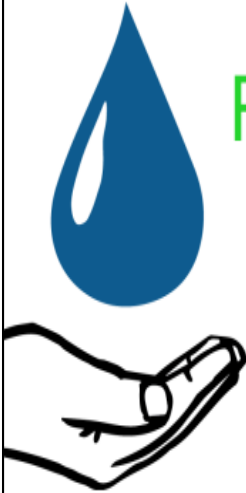
Future Work

- Finish Actual Distance SubVI
- Get more researchers to use the floc app
- Figure out way to store months of continuous data
- Take feedback, such as the video idea, into account

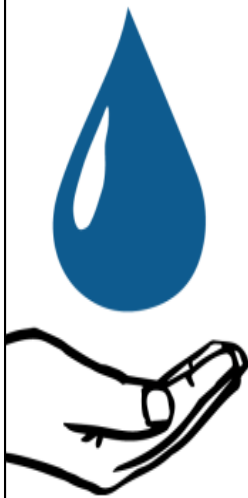# Questions and Recommendations

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

Apps and Algorithms | Symposium Spring 2017

Appendix
Slides

Apps and Algorithms | Symposium Spring 2017

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

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.