Floc App, Spring 2017

Christian Rodriguez (cer95), Anthony Verghese (akv26), Deniz Yilmazer (dy223)

May 19, 2017

Abstract

Turbidity measurements provide the primary source of performance monitoring at many water treatment plants. Turbidity provides an excellent way to measure overall plant performance, but it fails to provide insight into the specific processes that make up the plant. The earliest reduction in turbidity in an AguaClara plant occurs after the sedimentation tank with a time delay of 30 to 45 minutes. This delay makes feedback control challenging especially when raw water turbidity varies rapidly. The floc size and count app comes in to fill this shortcoming by providing an easy-to-use desktop application to measure floc distribution in various parts of the plant. The floc app can measure floc size distribution of the flocculated water to characterize flocculation efficiency. Flocculation residence time is expected to be less than 5 minutes in the next generation of AguaClara plants and this will significantly reduce the feedback time to provide guidance for setting the plant coagulant dose.

Introduction

The final goal of water treatment is to be able to decrease water turbidity (cloudiness) as much as possible and to at minimum reach the regulatory requirements. The World Health Organization (WHO) maximum turbidity is 5 NTU for drinking water and for US EPA the maximum turbidity is 0.3 NTU. Well designed and operated treatment plants can even achieve turbidities that are as low as 0.05 NTU. The treated water turbidity measurement are only meaningful after the sedimentation and filtration steps since those are the process where the flocs (particles in water) are removed. Therefore, there is no way for an operator to know what is happening in the plant until after sedimentation. This causes a great feedback delay since it takes water from 40 minutes (AguaClara) to 4 hours (conventional water treatment plants) to reach the exit of the sedimentation tanks. Therefore, if the water quality is not up to the standards, the operator would realize it and adjust the coagulant two hours late. This is a real issue since, depending on the volume of the plant, that might mean a lot of water that was supposed to be treated to the EPA standards actually being above the limits.

The floc app will solve this problem of delayed feedback. The floc app will measure the floc size distribution in the flocculated water just a few minutes after the coagulant addition. Plant operators will have the necessary information to adjust the coagulant dose with a response time of the flocculator residence time. With research coming along that aims to decrease flocculation below a minute, this would decrease the feedback time significantly and improve the stability of water quality for the plants.

The floc app analyzes the flocs at key point in a water treatment plant and returns arrays containing maximum particle diameters[1], average diameters, and projected particle areas.

Literature Review

Sun's thesis (2016) describes methods to characterize particle size distributions using image analysis. Sun's code was used as a starting point to develop the floc app. Sun used LabVIEW and its Vision Builder Toolkit to explore particle size distribution changes during flocculation and sedimentation. The thesis goes into great detail describing how she tested her floc-measuring device. Different concentrations of polyaluminum chloride as well as kaolinite clay were added to water to change turbidity and make sure the image analysis was applicable over a range of conditions. (Siwei Sun, 2016, p. 24, 40).

[1] The maximum Feret diameter is the longest possible distance between any two points on a shape.

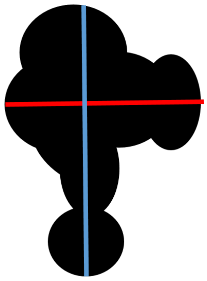


Figure 1: Feret Diameters are the the distances between endpoints of shapes. This figure shows two such diameters, of which the one represented by the blue line is the maximum Feret diameter

Sun describes the hardware of the imaging system consisting of an LED light source, flow cell, and a Flea3 FL3-GE-13S2M monochrome GigE camera (Siwei Sun, 2016, p. 25). A bright LED light provided enough illumination so the exposure time could be kept very short. This small exposure time keeps floc movement to a minimum while the picture is being taken. An exposure time that is too long causes blurring of the image.

Airy disks are a diffraction pattern that shows up at the scale of the wavelength of light and can result in an apparent increase in the particle size. (Siwei Sun, 2016, p. 27). Particle sizes can be overestimated if these airy disks are not taken into account because airy disks create a border around the particle that is larger than the particle. Particles that were too small (less than 2.6μm) were often too difficult to analyze and could not be clearly identified. The estimated radius of airy disk caused by yellow light (yyy nm wavelength) is around 1.29μm). Since each point around the particle has airy pattern, particles less than 1.29μm) in diameter cannot be clearly identified. The particle sizing apparatus used by Sun could measure particle sizes ranging from 2.6μm) (set by the airy disk) to more than 300μm) (set by the field of view). Standardized particles of known size were utilized to determine the error in particle size measurement caused by light diffraction (Siwei Sun, 2016, p. 28). The image processing steps used by Sun included:

- Begin applying a Gaussian filtering function to smooth the edges of particles for easier analysis(Siwei Sun, 2016, p. 29).
- Apply "local thresholding", effectively separating particles from surroundings by applying a dark color to the background and a light color for the particles(Siwei Sun, 2016, p. 29-30).
- Fill small holes and smooth the boundaries of the flocs to make them a closed solid object.
- Remove flocs that were out of focus. Out of focus flocs were identified based on a weak light intensity gradient at their borders (Siwei Sun, 2016, p. 32).
- Remove the flocs that have edges that cross the border of the picture frame. The application is unable to be sure of the dimensions of these types of flocs.
- Remove the flocs that are too small since they cannot be measured accurately because of the airy disk phenomenon.
- The remaining particles were then analyzed and their sizes were recorded in a comma separated value (csv) file (Siwei Sun, 2016, p. 37).

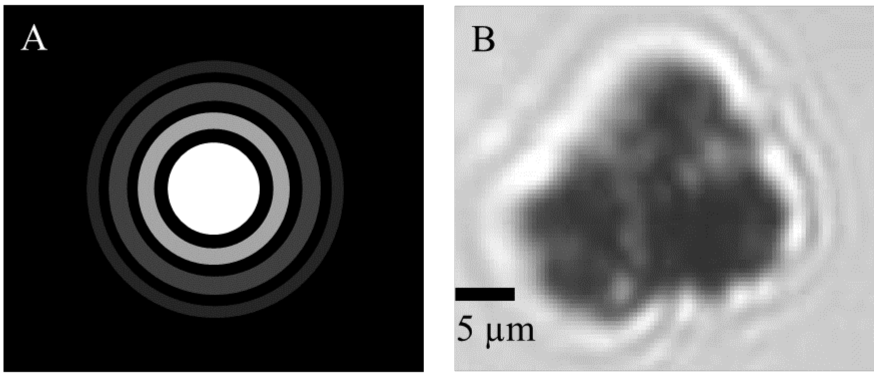


Figure 2: Effect of Airy Disks

Sun obtained a linear relationship between turbidity and the number of clay particles (Siwei Sun, 2016, p. 44). The slope of the line was determined to be $1.9 * 10^6 \pm 2.2 * 10^5 \frac{\text{particles}}{\text{L*NTU}}$ (Siwei Sun, 2016, p. 45). In addition, a plot was created of flocs versus coagulant dose (Siwei Sun, 2016, p. 50). Thus, with increased coagulant doses, the flocs became larger and the number of flocs decreased.

Previous Work

During the Fall 2016 semester, the team began developing an executable for the floc application. However, before successfully completing this task, edits needed to be made on the vision portion of the code as well as the overall GUI layout. The GUI was made more user-friendly and clear tabs were put in the GUI to allow the user to easily access the histogram and results sections. Once these actions were taken, the executable was made.

In order to allow researchers to download the executable, a tutorial and download link were put on the AguaClara Wiki page. This allowed users to find flaws in the executable and help the team make the Floc Size and Count Application even better.

Developments

Data

The original Floc App calculated the maximum Feret diameter, the average diameter, and the projected area, but only measured the first of these measurements to estimate a floc's size. The team decided that this was a pore choice because the sizes of thin flocs would be exaggerated by this measure, as the figure below demonstrates.

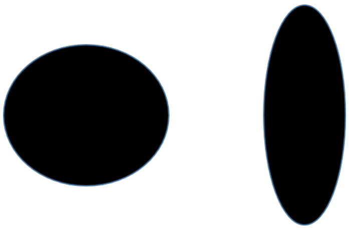


Figure 3: Max feret diameters and average diameters based on the projected area.

Instead, the team decided to base our measure on the floc's area. The team made the estimate that every floc as a circular object and used standard geometry to calculate an estimate diameter. This estimate diameter (d) is calculated using the following formula where α is the area of the floc:

$$d = 2\sqrt{\frac{\alpha}{\pi}}$$

Calculating Distance and Depth of Field

After Fall 2016, the Floc App generated a histogram of the maximum Feret diameters of all measured flocs in units of pixels. Digital cameras take images using a special sensor (either a CMOS or CCD sensor) that converts lights into electric signals through the use of numerous rectangular cells known as pixels (shown in Figure 4). The information from photographs is stored as a 2D array of the values of these pixels which is best thought of as a rectangle of values (noa (2013)). The original program took the number of these pixels separating two points (pixel length) and multiplied it by some constant specific to the creator's lab design. For instance, in the figure below, the red square is 5 pixels away from the blue square, so the program would say that the two points were $\lambda 5$ pixels apart, where λ was a constant scaling factor from pixels to micrometers.

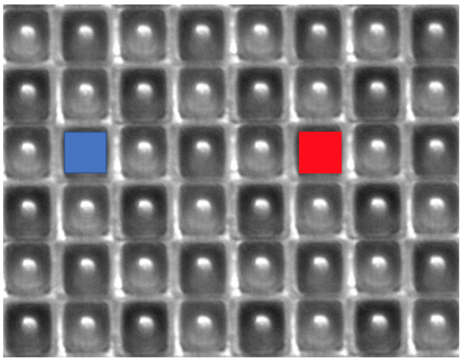


Figure 4: Cameras capture data from the environment through a series of pixels. Photo from noa (2000)

Unlike standard units of measuring length, such as millimeters, there is no standard pixel size. The pixels of cheaper cameras, for instance, tend to be larger than those of higher-quality cameras. Furthermore, the amount of pixels an object takes up may depend on the distance of the object from the lens, with closer objects appearing larger. This would make sharing data rather unreliable, since different users of the system may have different interpretations of a "pixel units," which is the length of a pixel, as shown in Figure.

Another possible concern is differences in the space considered to be in focus by the camera. Cameras differ in field of view and their depth of field. The field of view is the area visible to a camera. This is represented in Figure 5, by the two lines diverging from the camera's lens. Objects a certain distance away from a camera's lens (represented by the X) would appear as in focus as possible. Any objects closer to or farther from the camera would appear less in focus. The depth of field or focal depth coupled with the field of view defines a volume where particles are observed to be in focus. The definition of "in focus" is set by an image processing algorithm that assess the sharpness of particle borders. The depth of field is represented by the two horizontal lines in Figure 5.

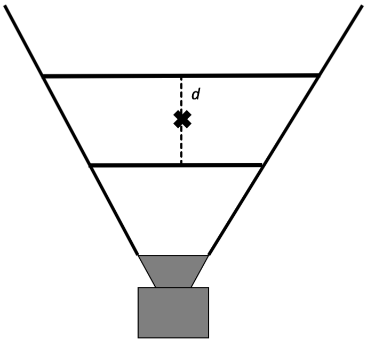


Figure 5: The depth of field and field of view influence the number of flocs analyzed by the Floc App

These two factors, the focal depth and the field of view determine the volume in which a floc can reside and be analyzed by the Floc App. An increase in either of these parameters would increase this sample volume and the number of analyzed flocs. The goal is to create a measure of floc concentration and therefore the flocs counts are divided by the sample volume. The horizontal and vertical components of this volume (in terms of pixels) are properties of the camera including the lens system. The depth of field requires some calculations.

Pixel dimensions from the image are converted to micrometers by multiplying the measurement by the number of micrometers represented by a single pixel (a scaling constant). For each analyzed picture, the program records the maximum Feret diameter and updates the array containing all of the floc data using micrometers. Then this new array is used to update the frequency histogram of the floc diameters measured. These frequencies, however, must each be divided by the volume in focus to normalize the data.

The pixel dimensions and depth of field are calculated in a Sample Volume calibration function (subvi). To calculate both the number of pixels per micrometer and the depth of field, the calibration function has the user take a photo of a ruler (using the system's camera) and mark two points. The steps of the process are shown in the figure below. First, two instructions are given to the user, and then the user must direct the camera towards the ruler to make the proper calibration calculations. The user is then asked to enter the actual distance between the two points on the ruler in millimeters.

The main differences between the calculations of these two parameters (the scale factor and the volume in focus) reside in both the calculations and the image to be taken. To calculate the proper scale factor, the user must take a ruler in focus before taking the photograph The x-y coordinates of the two points (which are pixels) are then used to calculate the pixel distance (λ) between the two points. The physical distance in micrometers is divided by the pixel distance, see equation below, where S is the scaling factor.

$$S = \frac{\sigma}{\lambda}$$

The depth of field (D) is measured by first taking a picture of the ruler placed at such an angle that part of the ruler is in focus and other parts are out of focus. The user selects the point that is most in focus and the farthest point from this picture that is in focus. Whereas the user put in the physical distance between the two points, the computer calculates only the horizontal distance (in pixel units) between the points. This allows us to use the Pythagorean Theorem to find half of the depth of field (See Figure). The physical distance is converted into pixel units and then squared. The pixel distance squared is then subtracted from this number. The square root of the result is multiplied by two and the scaling factor S. The Depth of Field is calculated using the following equation:

$$D = 2S\sqrt{\chi^2 - \lambda^2}$$

$$\chi = \frac{\sigma}{S}$$

To calculate the volume, this depth is simply multiplied by the scaled height (y) and scaled width (x)of the image. This works because the camera uses a fixed number of pixels to represent an image and we have already calculated the scaling factor from pixels to micrometers. Therefore, calculating the volume in focus is done through this equation:

$$V = Dxy$$

This calculation does not perfectly describe the sample volume. The error is small because the depth of field is extremely small relative to typical calculations. The sample volume can be approximated by a rectangular prism. The small depth of field also allows us to ignore the misrepresentations in size due to the variability of distance of particles from the camera lens.

Implementation

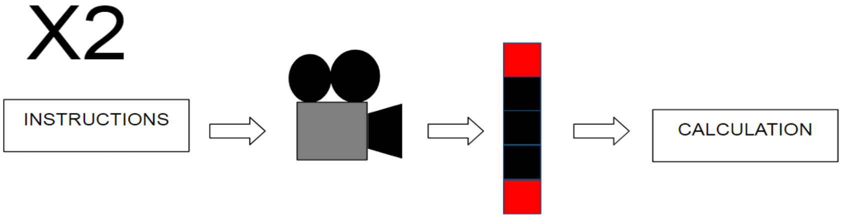


Figure 6: This diagram describes the calibration process, which is done twice, once for the scalar factor and the next time for the volume in focus. The user is first given instructions on how to position some ruler. He/she is then given a live streaming of what the camera sees until the "TAKE PHOTO" button is pressed. The user is then asked to select two points on the ruler and state their distance apart. Finally a parameter-specific calculations are run to get the desired variables.

The configuration function previously described will be executed by the Floc App when either before the user's first use of the application or if the user requests to do so by clicking a button. In both cases, two windows appear. The first of these is a dialog box asking the user to take a picture of a ruler completely in focus and then to select two points from the image, keeping track of the real-world distance between the two points. Upon pressing the box's OK button, this window terminates and the second window (which is in the background) displays a live streaming of the camera used by the Floc App. Above this display is a button which takes a picture.

Upon clicking this button, the program ends the stream and displays the last image taken. In this new window, the user selects his two points on the ruler and upon pressing ENTER is asked to provide the size of the actual distance between the two selected points in mm. After this, the program calculates the pixel distance between the two points (λ)by taking their x and y coordinates and using the standard formula $\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2)}$. The scaling factor S is then calculated using the formula in the section Calculating Distance and Depth of Field and stores the result in a configuration file.

To calculate the volume in focus, the function again displays a dialog box with instructions. This time, however, the ruler must be set at an angle so that there is a portion in focus and another portion of the ruler out of focus. This is done by setting the camera at an angle. Once the ruler is properly positioned, the user then takes the photo. Again, the user is asked to select two points (one in focus and the other right out of focus) and to input the real distance between the two points on the ruler in millimeters. To calculate the depth of field (D) and volume (V), we use the pythagorean theorem as shown in Figure 6 below. The calculations for both of these are provided in the section Calculating Distance and Depth of Field. The volume is stored in the configuration file.

After obtaining these two values, the function terminates and the user is returned to the main window of the Floc App program.

Storing Setting Data

Users frequently will use the same unit conversion and depth of field for multiple experiments. As a result, rather than forcing the user to go through the whole process of resetting the values, the program asks the user to change the settings upon first using the program and then stores them in a configuration file for future experiments. The user will always have the option of changing these settings by clicking a button in the controls tab of the program.

Making Default Values for GUI

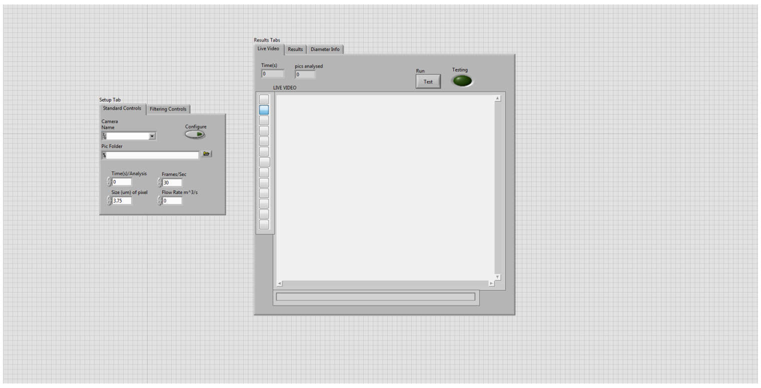
Default values for the main application variables that would be more user friendly were added to the program. Specifically, a default file path was hard coded to the program to save the images as well as a strictly positive integer for the seconds per analysis variables, which is the amount of time between two consecutive analysis. This enabled the user to just choose a camera and click run instead of having to manually enter this information every time the program was restarted.

Executable and Runtime Engines

An executable was created for distribution and user testing. The executable does not need the full software packages like LabVIEW or Vision extension, using only runtime versions for them would be sufficient. Detailed installment directions are provided on the AguaClara wiki.

Methods

When the user downloads the floc application, the GUI will look like Figure 5, shown below. There is a setup section and a Live Video/Results section.



Time(s): 0
pics analysed: 0
Time(s)/Analysis: 0
Frames/Sec: 30
Size (um) of pixel: 3.75
Flow Rate m^3/s: 0

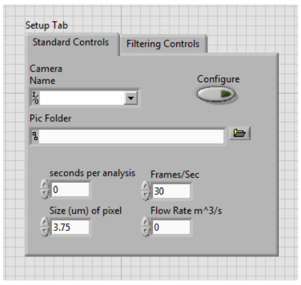
Figure 7: View of Floc App Executable Version

Under the Setup Section, there are two tabs. The first tab is for Standard Controls. In this first tab, one can select the camera from the drop down menu, Camera Name, and the folder to save the images, Pic Folder. If no folder is chosen, then a default folder is used. The user can then choose a seconds per analysis (for example if the user put 5, a picture is analyzed every 5 seconds). Within this tab, one can also find a button for calibrating the app so that it takes accurate measurements. This action, however, is executed automatically when running the program for the first time. The process is described in further detail in the Calculating Distance and Depth of Field sub section.

Furthermore, there are two "extra variables" included in this tab. One of them is a default frame rate setting. This variable stores that value of the frame rate (fps) to use if the camera used by the system does not store this information available. The cameras that will be used by AguaClara researchers, which are produced by Point Grey, seem to contain this information, so this setting will be of little consequence, at least within AguaClara. It was included primarily for the team's original testing of this program, which used a low-budget Logitech webcam.

There is also a flow rate parameter, which was originally included because the team wanted the program to calculate the ideal analysis per second parameter so that the same flocs do not show up multiple times. This idea has been abandoned for the time being, so this variable has no effect on the program.

The second tab of the Setup Section is the Filtering Controls. One would use this second tab if he or she is analyzing dark or bright particles (flocs are dark particles but you can use it to analyze other types of particles as well). As of now, the team does not know when these controls would be utilized, but they have been included just in case they may be useful.


Setup Tab

Standard Controls | Filtering Controls

Camera Name

Configure

Pic Folder

seconds per analysis: 0

Frames/Sec: 30

Size (um) of pixel: 3.75

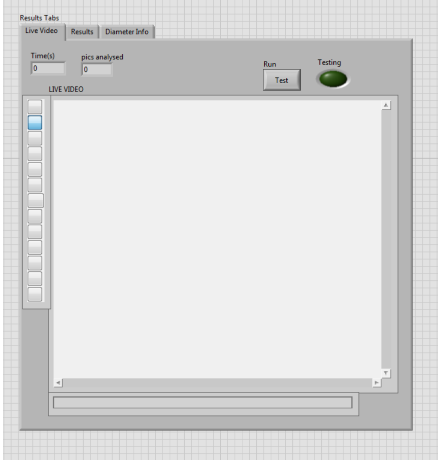
Flow Rate m^3/s: 0

Figure 8: Setup Portion of Executable

The Results section is divided into 3 subsections. The first is the Live Video subsection. In this section, one can see a continuous live video of the flocs as they are analyzed. This section also has the ”Test” button on its upper right corner (Figure 5) that starts the program. The green indicator next to the button (Figure 6) glows when the program is running.

The second subsection is called Results. It shows the last picture that was analyzed as well as total flocs found during the running time of the program.

The last subsection is the histogram that shows the diameter information of the flocs that were found by the program. The histogram distributes flocs according to their diameters.


Results Tabs

Live Video | Results | Diameter Info

Time(s): 0

pics analysed: 0

Run: Test

Testing

LIVE VIDEO

Figure 9: Results Section of Executable

Testing

Connecting Point Grey Camera to App

The team has struggled to properly connect a Point Grey BlackFly camera to the application; the proper software and firmware was installed, yet the XML file for the camera was giving an error in NI MAX. It was later figured out that by simply deleting everything in the XML file folder, which can be found in the following directory after the software for the camera is installed:C:(\)Public(\)Public Documents(\)National Instruments(\)NI-IMAQdx(\)Data(\)XML and then going to NI MAX and clicking on the Point Grey camera, an attribute error will pop up and by simply clicking the affirmative, the camera should be successfully connected to NI MAX. However, a new out-of-range error popped up after this procedure was done. Connecting simple USB web cameras, however, is easily done with the floc application.

Conclusions

The team focused on providing more useful, specific data for the user, simplifying and condensing the code by creating more subVIs, and working with other teams to see how useful the application truly is. The last of these three goals proved to be most difficult, because of issues that arose with connecting more advanced cameras, such as the Point Grey BlackFly camera, to the application and satisfying the users' needs. One research group wanted the floc application to display video, and throughout the semester we came up with other ideas, such as a warning signal for when flocs may be too big for a specific plant, that should be included in the application. Thus, actually utilizing the application with other teams created many new issues, but those can be resolved in the coming semesters.

Future Work

As it currently stands, the Floc App is not yet ready for field testing. It is capable of perceiving flocs and placing all relative data in a spread sheet and a histogram. We have yet to test the accuracy of the program's output, which will be critical for its use in research within AguaClara and possibly abroad. Although the team has acquired a high-quality Point Grey camera, technical difficulties with installing its necessary drivers have hindered the team's efforts to use it in testing, which should be conducted next semester.

Once it is confirmed that the Floc App collects accurate data, the team will move on to improving the functionality of the application. In particular, the process of calculating the volume in focus needs improvement. Currently, the process relies on the user's intuition to pinpoint the area at which the image begins to become out of focus. In analyzing its selected photographs, the Floc App already employs an algorithm to define in focus and out of focus portions of a photograph independently of what the user specifies. This would mean that the depth of field as defined by the user may be different from that used in the photo analysis. This problem should be dealt with next semester by employing Casey's algorithm in this portion of the program. The process could also be automated so that the user does not have to use a ruler to do the calibration. The goal is to convert the code to be event driven rather than locking the user into a sequence of actions. For example, it should be possible to take the two pictures of the ruler in any order. It should be possible to go back and replace one photo and repeat the analysis without forcing the user to repeat unnecessary steps.

In addition, the team must think about how data will be stored efficiently, especially because the application will likely run continuously for months at a time. Saving all images will probably be made optional given the large amount of disc space that would be required. A time stamp will also be added to the saved images to make it easier to identify a specific image. The team also aims to display results about long term trends for the plant operation in an easy-to-understand visual way. One challenge with this is going to be deciding on the representation of the data.

Another point to consider moving forward is that the program is getting more and more complicated each semester. The code volume is increasing greatly, and it is getting harder to divide it to subVIs since there are multiple distinct while loops that have to run together. The team might consider dividing the functionality between two distinct programs, one to get the images and basic size and count data, another for further analyzing, like for long-term trends. This could make the more ”basic” part run faster since the analysis is the bottleneck for the program running speed, therefore moving it to a distinct program can improve the data capture and enable better and more thorough analysis.

Although a first version of the floc application executable is up on the AguaClara Wiki, there is still much to be done, especially in terms of making the output data more usable.

Semester Schedule

Task Map

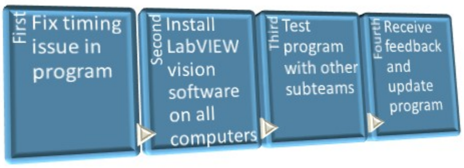

First
Fix timing issue in program
Second
Install LabVIEW vision software on all computers
Third
Test program with other subteams
Fourth
Receive feedback and update program

Figure 10: Task Map

Task List

1. ✓Improve the wiki tutorial page (2/10) - Deniz. Go through the wiki page and include more pictures to make it more understandable for the outside users.

2. ✓Transpose to LabVIEW 2015 (02/10) - Christian. Covert all LabVIEW 2014 code into LabVIEW 2015 code.

3. ✓Update the installation page to include runtime engines for LabVIEW 2015 and Vision software (02/14) - Anthony. This will solve the problem where lab computers could not use the app.

4. Test the LabVIEW 2015 version (02/17) - Christian. Check that the functionality holds for the converted code.

5. Revamp TeamForge (02/17) - Christian. Reorganize TeamForge folders and documents.

6. Incorporate Measurements (02/24) - Christian. Make the program show sizes in metric measurements instead of pixel numbers for more usable information.

7. Simplify timing (03/03) - Deniz. Simplify the timing for the image capturing part of the program and make it independent of the camera being used.

8. Extreme testing (03/31) - Anthony. Work with the team that uses the ”nice” camera and get their feedback. (Has this happened.)

9. Get research team using the app (all semester) - Deniz. Encourage the research teams to use the program. Go around and get feedback from the teams to further improve the app.

10. Revise wiki as the program changes through the semester (all semester) - Deniz. Add the new and changed functionalities to the tutorial wiki.

11. Improve the floc app by revising on the feedback we get from the research teams. (all semester) - All team.

Report Proofreader:

References

(2000). What is the difference between CCD and CMOS image sensors in a digital camera?

(2013). What a DSLR's CMOS Sensor Looks Like Under a Microscope.

Siwei Sun (2016). *Characterization of Flocs and Floc Size Distributions Using Image Analysis.* Partial Fulfillment of the Requrements of Degree of Master of Science, Cornell University, Ithaca, New York.