Slide 1

AguaClara

Floc App

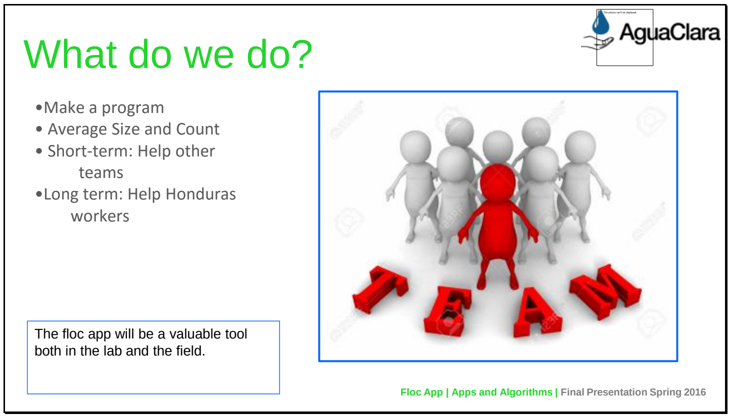
This presentation will describe the progress made by the Floc App Team as well as the steps made towards developing the final edition of the software. The team's page on the Agua Clara wiki site provides more extensive information about the team's research.

Floc App | Apps and Algorithms| Final Presentation Spring 2016

Abstract: Turbidity measurements are the primary sources of performance data in water treatment plants.   However, turbidity readings of flocculated suspensions do not provide any insight into the performance of subsequent treatment processes. The Floc Size and Count App Team's aim is to create an easy-to-use personal computer application that could measure floc size distribution using a digital camera with appropriate magnification. The app will be written in LabVIEW. The aim for this semester is to develop coding expertise in the LabVIEW environment and start working on the app that will be used in the AguaClara labs and eventually in AguaClara drinking water treatment plants.

Overleaf Research Report Link: https://www.overleaf.com/4434789sgbryb#/13262199/

Slide 2



What do we do?

- Make a program
- Average Size and Count
- Short-term: Help other teams
- Long term: Help Honduras workers

The floc app will be a valuable tool both in the lab and the field.

Floc App | Apps and Algorithms | Final Presentation Spring 2016

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.

Slide 3

Methods

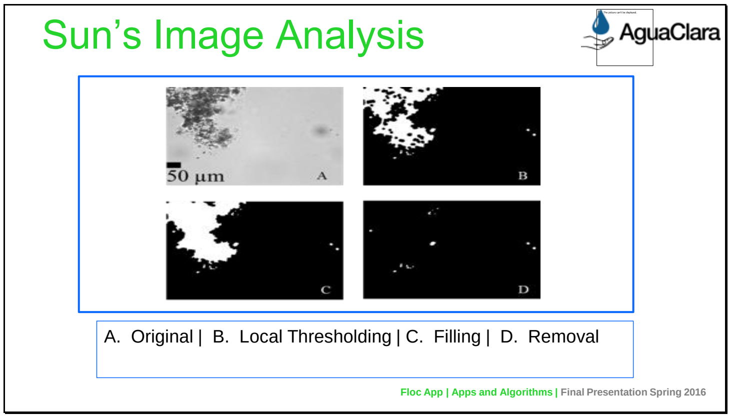
First Step: Research

Floc App | Apps and Algorithms | Final Presentation Spring 2016

To train ourselves with LabVIEW, all members of the team watched a series of LabVIEW tutorials on the National Instruments website (http://www.ni.com/academic/students/learn-labview/)

We then proceeded to watch supplemental YouTube videos from a variety of channels. Some of us also looked at the LabVIEW White Papers and practiced their exercises.

Slide 4



Sun’s Image Analysis

AguaClara

A. Original | B. Local Thresholding | C. Filling | D. Removal

Floc App | Apps and Algorithms | Final Presentation Spring 2016

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

Caption of Slide: These are the steps of image analysis as outlined in Siwei Sun’s Thesis. These steps are used in the original program and served as model we used when organizing the original code into subVIs (functions).

Summary:
These are the steps of image analysis as outlined in Sewei Sun’s Thesis.

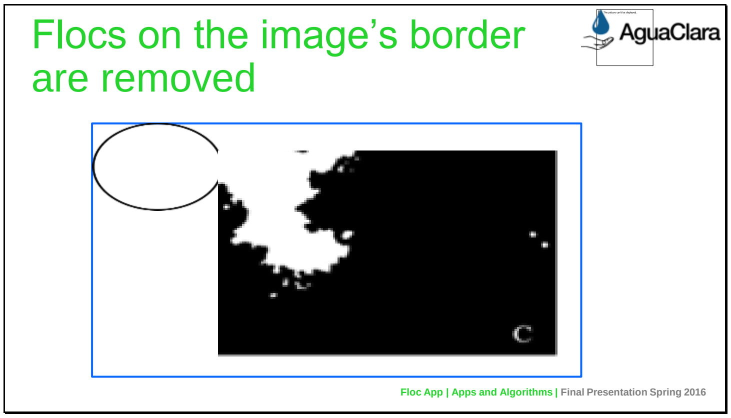
- Figure A Represents the image before any changes are made for processing.
- Figure B represents the original image after the Gaussian filter and local thresholding have been applied (Slides 11-13).
- Figure C is Figure B after large holes inside the flocs were filled. This makes the image easier to process.
- Figure D Represents Figure C after the removal of unanalyzable flocs. This group includes flocs that are out of focus, too small, or on the boundary of the generated picture.

The problem with “border” flocs is that measurements are ambiguous.

Flocs that are out of focus cannot be used to obtain accurate data.

The issue with small flocs is airy disks that form.

Slide 5



Flocs on the image’s border are removed

AguaClara

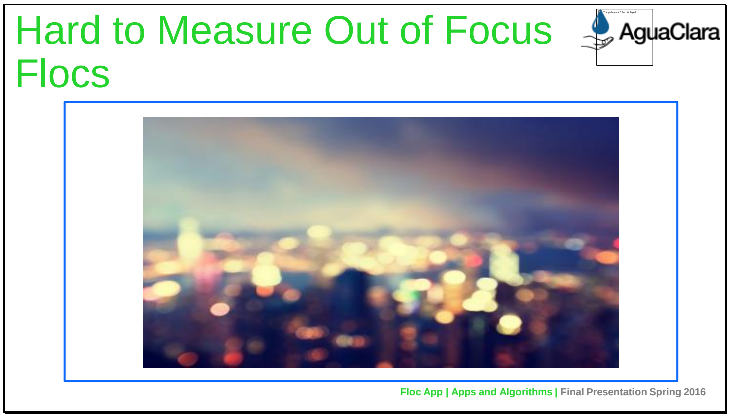
Floc App | Apps and Algorithms | Final Presentation Spring 2016

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

Caption of Slide: Explains why flocs touching boundary are removed.

Summary:The problem with “border” flocs is that measurements are ambiguous. One is unable to determine how far the floc extends past the allotted image size, so the border flocs are not counted as one of the flocs in the image.

Slide 6



Hard to Measure Out of Focus Flocs

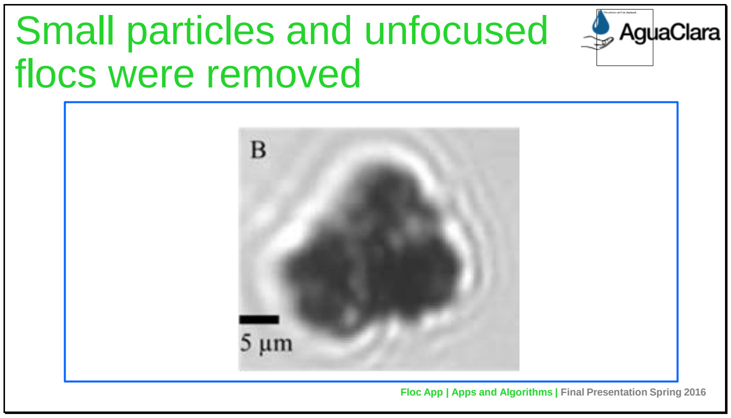
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image retrieved from:
Charara, S. (2014, September 12). The future of photography is out of focus. Retrieved May 20, 2016, from http://www.stuff.tv/features/future-photography-out-focus

Caption of Slide:Explains why out of focus flocs are removed.

Summary:It is hard for the program to take measurements of out of focus flocs, just as it is hard for you to determine what is in this picture exactly.

Slide 7



Small particles and unfocused flocs were removed

Floc App | Apps and Algorithms | Final Presentation Spring 2016

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

Caption of Slide: Explains why Casey’s Code Removes Small Particles.

Summary: If a particle is too small (with dimensions of 2.6 μm), “light rings” form around it. This phenomenon occurs when a particle’s dimensions are close to the wavelength of light. This makes measurements inaccurate, so such particles are removed.

Slide 8

Methods

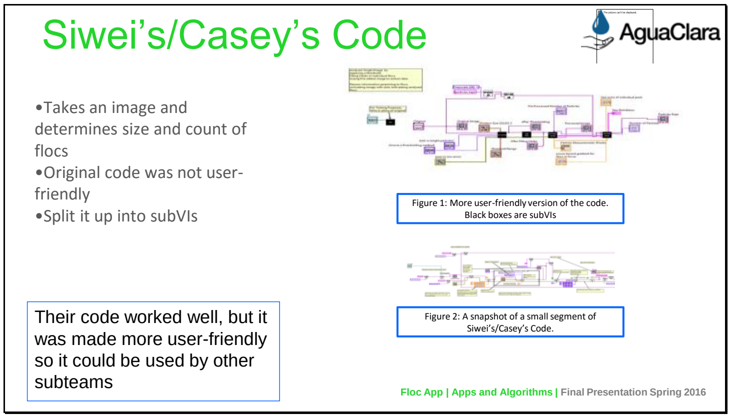
Second Step: Splitting up Tasks

Floc App | Apps and Algorithms | Final Presentation Spring 2016

Caption of Slide: Introduces the method that the team took of splitting up tasks.

Summary: One member worked on splitting the original code up into subVIs to create a new VI for the code. Another member worked on the executable of the code. The last member worked on the image acquisition code, essentially the second VI necessary for successful implementation of the floc app.

Slide 9



Siwei’s/Casey’s Code

AguaClara

- Takes an image and determines size and count of flocs
- Original code was not user-friendly
- Split it up into subVIs

Figure 1: More user-friendly version of the code. Black boxes are subVIs

Figure 2: A snapshot of a small segment of Siwei’s/Casey’s Code.

Their code worked well, but it was made more user-friendly so it could be used by other subteams

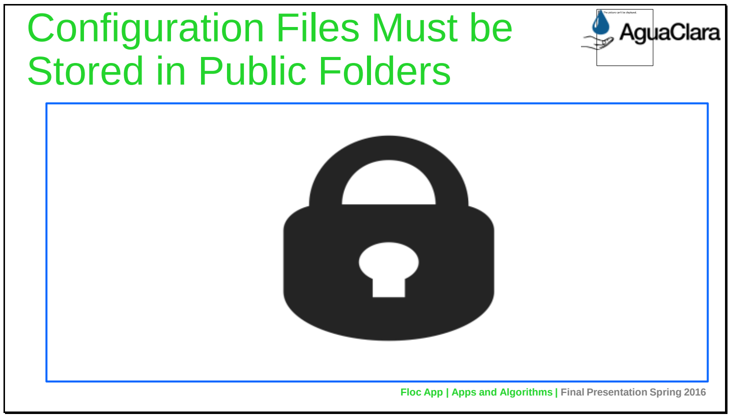
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Images are snapshots of Siwei’s/Casey’s code as well as our more compact code.

Caption of Slide: This slide presents one of the three tasks designated to a member of the Floc App Team. The first task in developing the code was to split up Siwe’s/Casey’s code into a more compact, user-friendly version, and this slide displays how that was accomplished.

Summary: The original code made by Siwei and Casey was very advanced and required extensive research into image analysis using LabVIEW. However, there were clear steps in the program: first the original image goes through a thresholding process,then holes in the image are filled, and finally the image is processed to create a processed image that shows the number of flocs left after the filtering process. Thus, with these clear steps, we were able to split the long and complex code into individual subVIs. Thresholding, fill hole, process image, and process data subVIs were made and wired together in a new VI to create a more user-friendly version of the original code. Some unnecessary aspects of Siwei’s and Casey’s code were removed, but most aspects were kept the same because they were crucial for determining the size and count of the flocs in an image.

Slide 10



Configuration Files Must be Stored in Public Folders

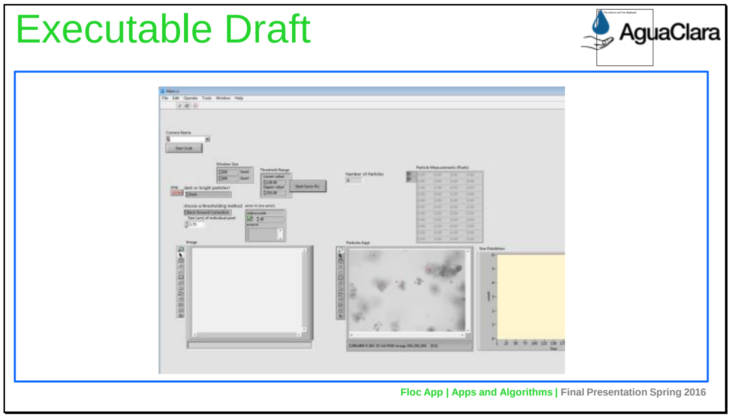
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image retrieved from clipart

Caption of Slide: Describes problem with creating draft executable.

Summary: Configuration files should be stored in public directories in order to be accessed by the executable regardless of user.

Slide 11



Executable Draft

AguaClara

Floc App | Apps and Algorithms | Final Presentation Spring 2016

Citation: Snapshot of image acquisition LabVIEW code

Caption of Slides: Shows final result of executable creation.

Summary: After some practice, we developed an executable that is supposed to analyse a single image. In the future, we hope to develop this program so that it can handle multiple images.

Slide 12

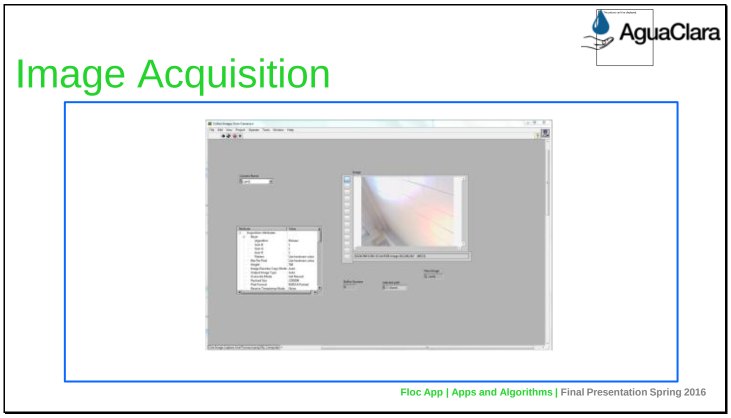


Image Acquisition

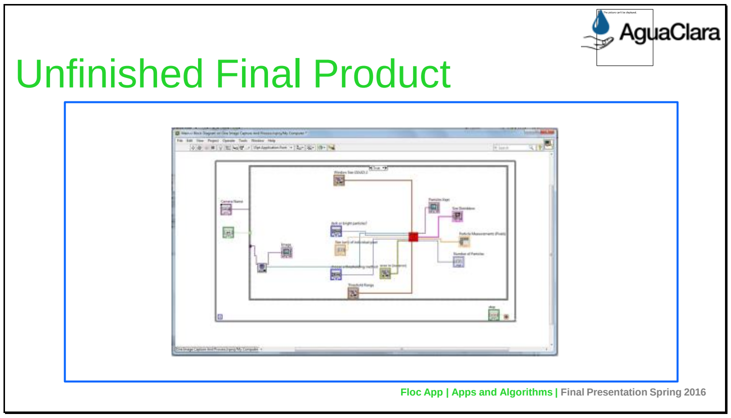
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Citation: Snapshot of image acquisition LabVIEW code

Caption of Slides: Working functionality of the image acquisition

Summary: The image acquisition VI is the part where it takes an image with the camera and saves it on the computer.

Slide 13



Unfinished Final Product

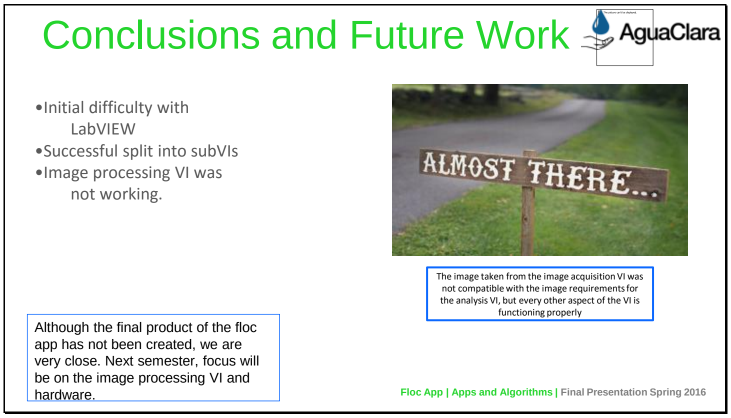
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Citation: Snapshot of image acquisition LabVIEW code

Caption of Slides: Unfinished final product

Summary: The VIs themselves work well however there are problems in the connection between the two that still needs to be addressed.

Slide 14



Conclusions and Future Work

AguaClara

- Initial difficulty with LabVIEW
- Successful split into subVIs
- Image processing VI was not working.

ALMOST THERE...

The image taken from the image acquisition VI was not compatible with the image requirements for the analysis VI, but every other aspect of the VI is functioning properly

Although the final product of the floc app has not been created, we are very close. Next semester, focus will be on the image processing VI and hardware.

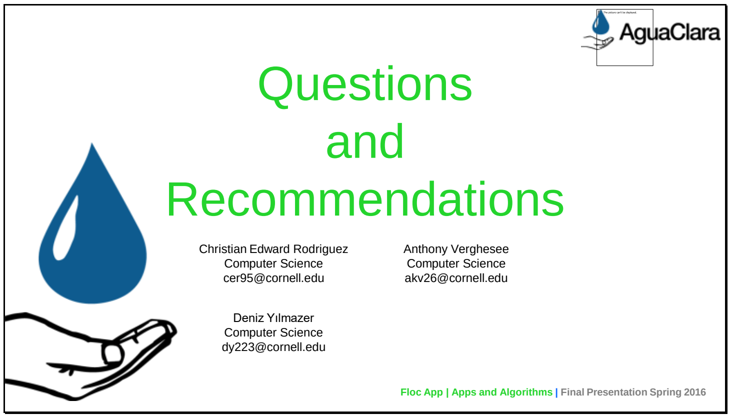
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image Citation: https://potpourriofthedamned.files.wordpress.com/2014/12/almost-there-sign.jpg

Caption of Slide: Explains what was accomplished during the semester and describes what the team’s future goals are. In addition, this slide shows what went wrong during the semester and how the team intends to fix those problems.

Summary: The final product will be acquiring the data (photos of the flocculator in this case) and analyzing it in a single program. The team finished working on both data analysis and capture parts of the functionality for the product. However, these were two separate VIs at the end of the semester. The attempts to combine them together had not been successful. The image taken from the image acquisition VI was not compatible with the image requirements for the analysis VI. Using the file path for the saved image could be a way to fix this problem.
Despite having difficulty with learning LabVIEW, a complex language to get accustomed to when used to languages such as java and matlab, for the first half of the semester, the first semester of the Floc App team was quite successful. Being one of the few teams using LabVIEW was quite tough because there were few people to aid the team when complications occurred. In addition, Casey and Siwei's code was quite complex, so trying to make sense of the given code that was necessary for editing was a tough process. However, after getting used to the LabVIEW platform, working on the code became slightly easier. The subVI program was created successfully and is much more user-friendly, a great benefit for other teams that will use the floc application for their own research.

Slide 15



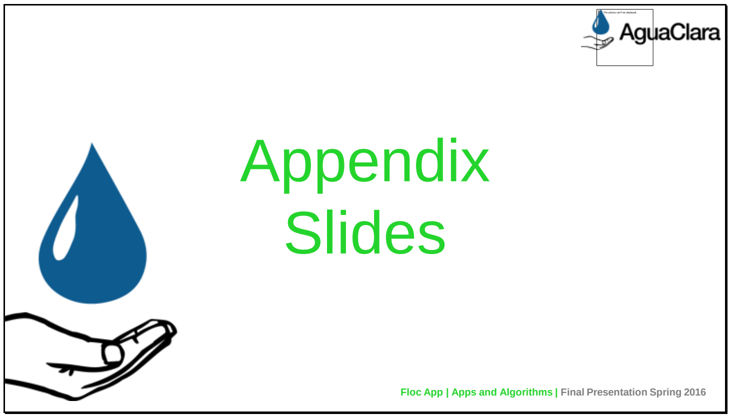
Questions and Recommendations

Christian Edward Rodriguez
Computer Science
cer95@cornell.edu

Anthony Verghesee
Computer Science
akv26@cornell.edu

Deniz Yılmazer
Computer Science
dy223@cornell.edu

Slide 16



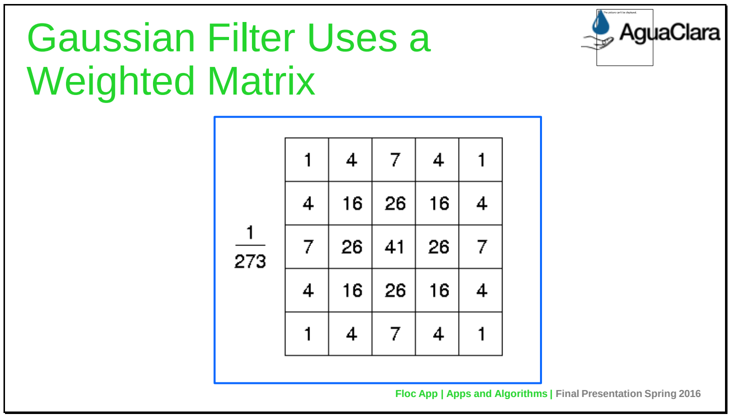
AguaClara

Appendix Slides

Floc App | Apps and Algorithms | Final Presentation Spring 2016

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

Slide 17



Gaussian Filter Uses a Weighted Matrix

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

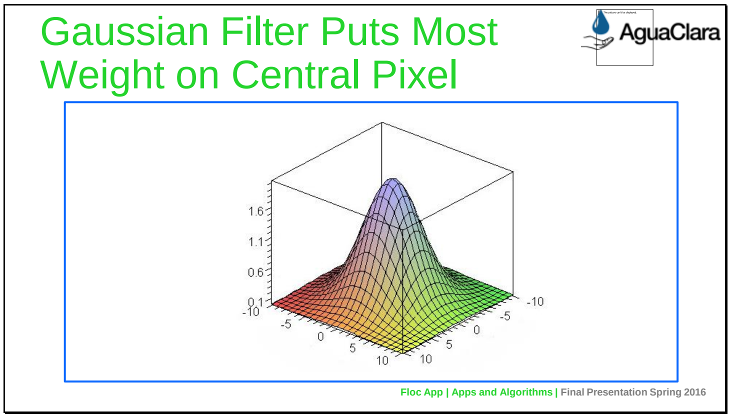
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image Citation: Fischer, R., Perkins, S., Walker, A., & Wolfart, E. (2003). Gaussian Smoothing. Retrieved May 14, 2016, from http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Caption of Slide: Describes in more detail how the gaussian filter works, a major component of the code created for the floc app. This slide simply provides a better understanding of how the floc app team created the code described in the main slides.

Summary: This illustrates how a Gaussian Filter works. The Gaussian Filter works with a matrix, kernel, of values based on a center pixel and surrounding pixels. The average of the old pixel values times their weights is generated and this average determines the pixel intensity of the central pixel. Note that the sum of all the numbers in the boxes is 273.

Slide 18



Gaussian Filter Puts Most Weight on Central Pixel

AguaClara

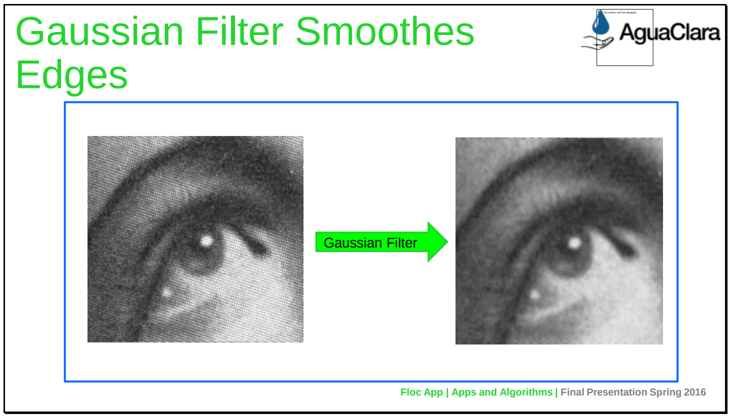
Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image Citation: Efficient Gaussian blur with linear sampling. (n.d.). Retrieved May 14, 2016, from http://rastergrid.com/blog/2010/09/efficient-gaussian-blur-with-linear-sampling/

Caption of Slide: This slide goes into further detail into how the gaussian filter works. It essentially explains how one step in the code works, thus giving a better understanding of how the final product is accomplished. It is in the appendix because it is not essential for other teams to understand how the gaussian filter works but it could be interesting or provide better understanding.

Summary: This illustrates the weighting given to each pixel in a Gaussian Filter. Notice that the center image, the one whose value is being determined, has the highest weight in determining its new value.

Slide 19



Gaussian Filter Smoothes Edges

Gaussian Filter

Floc App | Apps and Algorithms | Final Presentation Spring 2016

Image Citation: https://en.wikipedia.org/wiki/Gaussian_blur

Caption of Slide: This appendix slide provides more understanding into the image effects that the code accomplished. Although the main slides explain what the code does, describing how the code accomplishes certain things such as smoothing edges is very important, and this slide provides insight into the functionality of the code.

Summary: Shows results of gaussian blur. After the Gaussian filter is applied to an image, the edges are blurred, with surrounding pixels taking on similar values. This makes the images easier to analyze with the program.