

5-2019

## Object Enhancement, Noise Reduction, Conversion and Collection of Spatiotemporal Image Data

Michael Izazguirre Jr.  
*The University of Texas Rio Grande Valley*

Follow this and additional works at: <https://scholarworks.utrgv.edu/etd>



Part of the [Physics Commons](#)

---

### Recommended Citation

Izazguirre, Michael Jr., "Object Enhancement, Noise Reduction, Conversion and Collection of Spatiotemporal Image Data" (2019). *Theses and Dissertations*. 482.  
<https://scholarworks.utrgv.edu/etd/482>

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact [justin.white@utrgv.edu](mailto:justin.white@utrgv.edu), [william.flores01@utrgv.edu](mailto:william.flores01@utrgv.edu).

OBJECT ENHANCEMENT, NOISE REDUCTION, CONVERSION  
AND COLLECTION OF SPATIOTEMPORAL IMAGE DATA

A Thesis

by

MICHAEL IZAGUIRRE JR

Submitted to the Graduate College of  
The University of Texas Rio Grande Valley  
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INTERDISCIPLINARY STUDIES

May 2019

Major Subject: Science and Technology



OBJECT ENHANCEMENT, NOISE REDUCTION, CONVERSION  
AND COLLECTION OF SPATIOTEMPORAL IMAGE DATA

A Thesis  
by  
MICHAEL IZAGUIRRE JR

COMMITTEE MEMBERS

Dr. Natalia V. Guevara  
Chair of Committee

Dr. Juan Guevara Jr.  
Committee Member

Dr. Edgar Corpuz  
Committee Member

Dr. Daniele Provenzano  
Committee Member

May 2019



Copyright 2019 Michael Izaguirre

All Rights Reserved



## ABSTRACT

Izaguirre, Michael, Jr., Object Enhancement, Noise Reduction, Conversion and Collection Of Spatiotemporal Image Data. Master of Science in Interdisciplinary Studies (MSIS), May, 2019, 60 pp, 25 figures, reference, 33 titles.

In this report, a variety of cellular dynamics are enhanced and analyzed utilizing various algorithms and filters for contrast enhancement. This report will also illustrate the underlying complexities of processing compressed data received from certain type of sensors, their default applications, various methods in converting compressed data to compatible universal uncompressed formats allowed in scientific applications, various methods of image and video capture, guidelines in ethical image manipulation, various methods of frame extraction, and analyzing/processing video images. These methods and processes purposely utilize freeware and public domain software to lower the cost of reproducibility for all.





## DEDICATION

To my parents whose struggles became my opportunities. From working the fields to granting me the ability to pursue my studies, their unwavering love and hard work has granted me the American dream. Without their struggles, I would be a different man. To my siblings, the natural competition and love we have shared has propelled us to unimaginable heights. The journey is far from over and there is much more we are set out to accomplish. To my grandparents, who also accepted the burden of the American dream for your children and grandchildren to one day benefit from your sacrifices. The completion of this degree is one of those benefits and for that I am eternally grateful. To close friends who have widened my view of this world and granted me eclectic perspectives.

The many ways in which we see this world will determine how we set out to change it.



## ACKNOWLEDGMENTS

I would like to thank Dr. Natalia Guevara whose passion towards providing understanding and patience to learn has granted me a strong foundation to further my academic journey. She constantly formed a caring and collaborative environment to learn, research and grow as a scientist. Thank you to Dr. Juan Guevara Jr. for his enthusiasm towards discovery and understanding that made me fall in love with the adventures of science again. The experience I have gained while working closely with these distinguished scientists have granted me a strong foundation to pursue my academic career and seek novel discovery and invention.



## TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES.....	viii
CHAPTER I. INTRODUCTION.....	1
Sample Images and Procedures.....	1
Digital Images.....	3
Ethical Guidelines for Manipulation of Scientific Digital Images.....	16
CHAPTER II. METHODS.....	21
Optical Video Micrographs.....	21
FFmpeg ImageJ Plugin.....	28
FFmpeg to ImageJ.....	28
CLAHE as Means for Improving Intensity Contrast.....	32
Filters.....	38
CHAPTER III. RESULTS.....	40

Histogram Stretch.....	40
Removal of Background.....	45
Filters (Median/Gaussian).....	47
Histogram Stretch and CLAHE.....	48
CHAPTER IV. DISCUSSION.....	53
Video Analysis .....	53
Capture Method.....	54
Histogram Stretch.....	54
Rolling Ball Background Subtraction .....	54
Filters.....	55
CLAHE.....	55
CHAPTER V. CONCLUSION.....	56
REFERENCES .....	57
BIOGRAPHICAL SKETCH .....	61

## LIST OF FIGURES

	Page
Figure 1: Into the Nucleus of a HeLa Cell.....	2
Figure 2: Surface of a Cell With Observable Membrane Proteins .....	3
Figure 3: RGB Color Space Encapsulated by the YUV Color Space [11].....	8
Figure 4: Example of Chroma Subsampling Schemes .....	9
Figure 5: YUV420 Subsampling .....	11
Figure 6: FFMPEG Retrieve Input Data Details.....	12
Figure 7: Kerr Illustration of YUV Subsampling .....	13
Figure 8: List Options Results using FFmpeg .....	14
Figure 9: Ethical Guidelines to Scientific Image Manipulation [19].....	18
Figure 10: Capture Formats and their Respective Properties .....	24
Figure 11: Color Banding Comparison Between Commercial Software and FFmpeg's vfwcap .	25
Figure 12: Color Blending Comparison Between Commercial Software and FFmpeg Vfwcap..	26
Figure 13: Proteins on the Surface of a HeLa Cell: Histogram Stretch.....	41
Figure 14: Into the Nucleus of a HeLa Cell: Histogram Stretch.....	42
Figure 15: Ecoli Grazing on the Surface of a Cell: Histogram Stretch .....	43
Figure 16: Viral Particles on the Surface of a Cell: Histogram Stretch.....	44
Figure 17: Proteins on the Surface of a HeLa Cell: Background Subtraction .....	45
Figure 18: Into the Nucleus of a HeLa Cell: Background Subtraction.....	46
Figure 19: Ecoli Grazing on the surface of a Cell: Background Subtraction .....	46



Figure 20: Viral Particles on the surface of a cell: Background Subtraction.....	47
Figure 21: Comparative View Between Median Filter and Gaussian Filter of an Image.....	48
Figure 22: Proteins on the Surface of a HeLa Cell: CLAHE.....	49
Figure 23: Into the Nucleus of a HeLa Cell: CLAHE .....	50
Figure 24: E.coli Grazing on the Surface of a Cell: CLAHE .....	51
Figure 25: Viral Particles on the Surface of a Cell: CLAHE.....	52

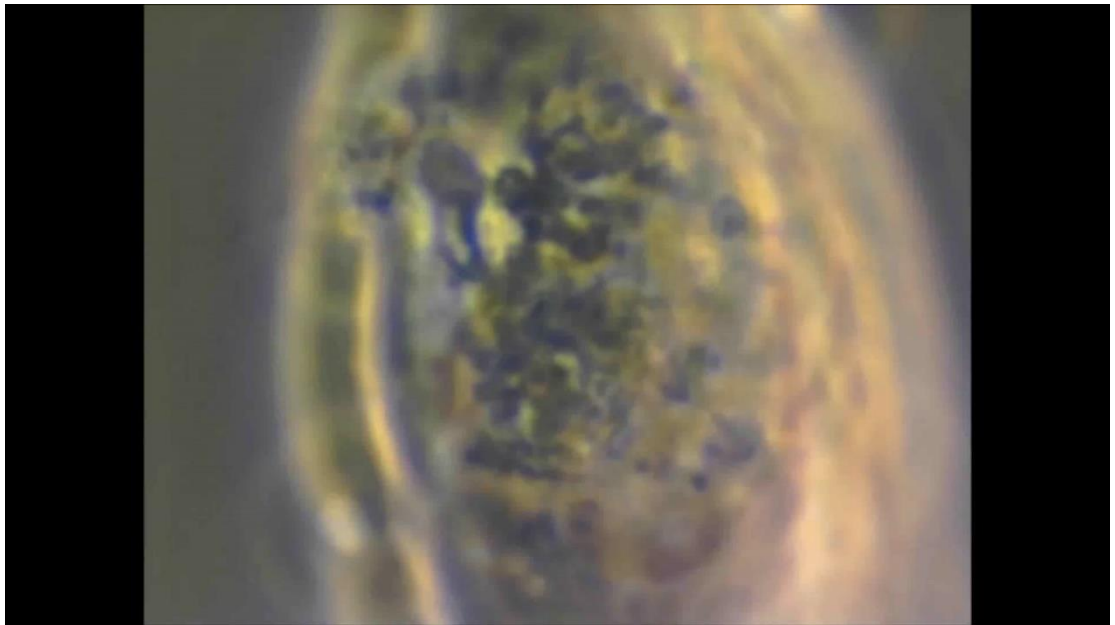
## CHAPTER I

### INTRODUCTION

#### **Sample Images and Procedures**

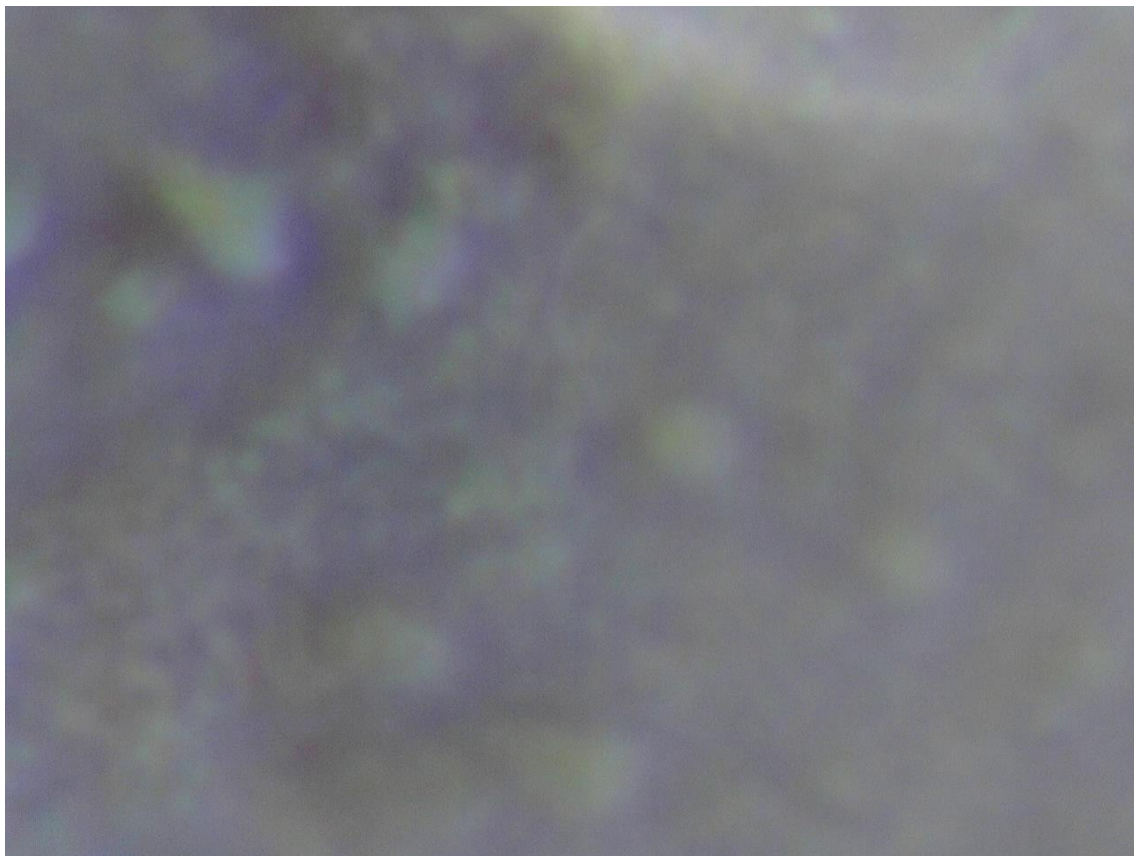
This section will display unprocessed images taken from early setups that utilized generic digital microscopes that will be enhanced and analyzed in a later section. These images were taken by various researchers from the Biophysics laboratory and few were first reported in Mr. Jaime Romo's Thesis "Optical Method for High Magnification Imaging And Video Recording Of Live Cells At Sub-Micron Resolution" in December of 2016. [1] As figures 1 and 2 illustrates, there appears to be a haze and distortions in the extracted frames from various videos. These visual obstructions lead to contrast gradation which are apparent in the following figures. This has been attributed to the relative scale of the nanographs as small changes in the focal plane, objects moving into and out of the fixed focal plane as well as light scattering and the fact that these cells are in an aqueous medium. This is a primary problem involving the nanographs collected using this microscopy system as the sensors are not selective in the light that they capture but rather captures all light that is being projected towards the sensor by the lens setup. The increased magnification of the system allows for the capture of smaller objects aplenty. These objects both biological and chemical in nature, are moving within the medium, cellular surfaces, and/or within the cell and its components. Objects close to the resolution limit that traverse in and out of the focal plane the microscopy system is on are the cause of various obstructions seen as a haze. This haze obstructs some qualitative observations by decreases the

local contrast. This haze is like unprocessed underwater imagery due the propagation of light in water, absorption and scattering. [2] It is seen later in this paper that treating the images and videos captured using this microscopy method as underwater images allows for better qualitative perception of the images and videos.



**Figure 1: Into the Nucleus of a HeLa Cell**

Figure 1. Single Frame from a two-minute video of cervical HeLa cells captured using an AM7013MZT (DMA). The original resolution of 1920x1080 pixels however after removal of the black regions around the usable image, the resolution is closer to 1436x1077. This frame specifically is chosen as the microscopist varied the focus to peer into the nucleus and observe what appears to be the chromatin of the cell.



**Figure 2: Surface of a Cell With Observable Membrane Proteins**

Figure 2 Surface of a Cell with Observable Membrane Proteins. This is the first frame from a roughly 35 minute video taken at 15 fps and resolution of 1280x960. The cells are live and unfixed and recorded using a Zeiss LD Plan-NeoFLU AR 63x/0.75 Ph2 Korr  $\infty/0-1,5$  objective. The FOV here is 32 microns which represents a pixel size of 25 nm. The proteins on the surface that can be seen moving in speed up renditions of the video range from 250-900 nm where smaller proteins are seen in linear shapes interacting with the larger proteins. This is enhanced later in this paper and become vividly apparent.

### **Digital Images**

Digital images are numerical representation of a three-dimensional objects. A moving image, video, adds time as another dimension where changes can be seen. The microscopy system used relies on a CMOS, complementary metal-oxide-semiconductor, sensor to capture and record the

light images projected onto it by the microscopy system. Generally, in scientific settings, CCD, charge-coupled devices, are used in microscopy set-ups as they have significant improvements when compared to the CMOS used in our setup. These improvements are also reflected in their prices and represent a financial barrier and burden to underfunded labs across the world.

## **ImageJ**

ImageJ, Fiji, is a prominent Image Processing software used in the bioimaging community as it is developed and supported by the NIH and has a large community of contributors. Developed under the public domain, this makes ImageJ a viable option to many researchers who do not have the finances to purchase more intensive and expensive Image processing software. ImageJ has various filters, algorithms, and processes already written and ready to be implemented by the end user. It also has limitations in regards to formats even with various projects allowing hundreds of different file formats, windows media video and mp4 formats are not able to be opened in ImageJ without a plugin. The most widely used scientific image file, TIFF, is favored in many instances. The video format native to ImageJ is an Uncompressed AVI or a compressed AVI in YUV 4:2:2 which will be discussed in a later section. ImageJ also allows macroinstructions such that if a time stack of images is imported then it is easy to process multiple images with just a few lines of codes such that each image in a time stack is processed with the same parameters and algorithms. [7]

## **FFMPEG**

FFMPEG, Fast Forward MPEG, is freeware under the LGPL 2.1+ (GNU Lesser General Public License) and GPL 2+ (GNU General Public License). It gives access to a variety of libraries and code used in handling multimedia data. It can also be used to gain information

about sensors connected to the PC as well as any existing video file. FFmpeg can encode, decode, transcode, mux, demux, stream, filter, and play pretty much any file format available. It is lightweight, portable, versatile and most importantly free. It can also be built on top of for future ventures and automation of processes. It also includes one of the biggest communities of users who specialize in video production, engineering, and editing. FFmpeg in this paper is used to convert, extract and capture image and video data from preexisting video and digital microscope sensors. Additional information can be found on their website. There are various projects that build upon FFmpeg that allow a graphical interface as well as other options not readily apparent by using just the command console and the FFmpeg manual; however, for this paper, the base program was used with simple lines used in the console. This is used for a problem that arises due to the default formats used by the software that is used with the digital microscopes, as later sections will show solutions and alternate methods to utilize FFmpeg to achieve a beneficial output pre-processing. [8]

### **Image Data Structure**

A digital image is a dot matrix data structure composed of a pixel array each holding unique values that determines the pixels color. In RGB images, assuming 256 bins per color, allows roughly 16.7 million different colors. These colors differ while many times color approximately close to each other are indistinguishable to the human eye. The more points in which a sensor can sample a distinctive color of an image, the higher the resolution. The more distinctive pixels per FOV the higher the respective resolution of an image. RGB is utilized in most computer monitors and applications due to its similarities with the human perception of the three sets of wavelengths of red, green and blue. Relative to the device used to visualize the

image, the color gamut is restrictive when the image contains values not within the color gamut of the device. The color gamut is a range of colors in which a device can visualize an image. [20]

### **Hue, Saturation, Brightness**

Hue is the description of color, such that, regardless of other attributes, lightness, darkness, intense, or pastel, i.e. dark blue vs. light blue. Saturation corresponds to the vibrance of the image. A grayish-blue would have less saturation than a deep or intense blue. Lower saturated images appear washed out as seen in some of the preprocessed images displayed in this paper. Brightness represents are dark or light a color is. The higher the brightness to closer to white the pixel is, zero represents a black pixel. Luminance and Chrominance are complimentary concepts where luminance is perceived brightness in relation to certain colors, i.e yellow compared to blue, while chrominance is a two-dimensional color space that represents hue and saturation separate from brightness. These two play a role when looking at color spaces such as YUV. [20]

### **Color Space**

Color is commonly described with two quantitative attribute, saturation and brightness, and one qualitative attribute, hue. Like the components in the structure of the eye, RGB, Red-Green-Blue, is the color space commonly used. This color space consists of three components ranging from 0-255 that determine the percentage of red, green and/or blue of the pixel. This causes for intensity and color to be mixed within the RGB geometric space, which is evidently cubic. HSI and HSV , Hue, Saturation, value and intensity respectively, are color space transformations of RGB in which color and intensity is separated. The geometric shape is

conical/cylindrical and appears more intuitive than RGB cubic geometric shape where the three component values are respective of their name. [9, 10] There are well solidified transformations to change RGB to these other color spaces. YUV is one of the most common color spaces for commercial products as during development it sprung from black and white TVs. As broadcasters wanted to have a format compatible with old TVs while introducing color, it was built upon the first component, luma value, and added two Chroma values that are essentially color difference signals. YUV was developed on convenience from the TV industry rather than any scientific initiative. It seems apparent that from RGB color space many conversions are available to transform our data into new color spaces. [11] With the advancements being made in image segmentation for recognition, referencing a database of biomolecules with our videos or creating the database, it might be viable to not only automatically recognize structures or objects but also track them, count them, and differentiate between edges and regions. The ability to save our video micrographs in a color space that benefit from these advancements would be beneficial as our method becomes more defined and sophisticated. Khattab D, et al showed that RGB space is currently the best color space for color image segmentations through a specific software. [9] Other color spaces still being researched are different types of CIE and various other RGB color spaces; however, for now it appears that RGB is the most beneficial color space to process our images in so that the images and videos are compatible and can be used with more advanced techniques.

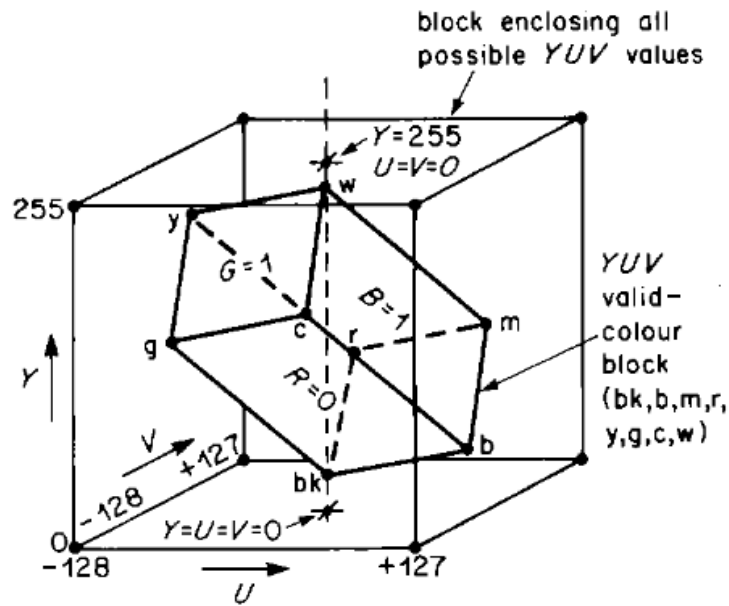
### **YUV and RGB**

As discussed previously, YUV was designed as an addition to the gray scale images early TV broadcasting used. This provided the ability to broadcast signals that were compatible with



older black and white TVs as well as newer color TVs and utilized the human's ability to better differentiate between contrast, luminance, which is the Y component of YUV to save data bandwidth using chroma subsampling. The chrominance values are contained in the U and V components typically as blue luminance and red luminance respectively and is seen as YCbCr at times instead of YUV. In general, U and V represent a color difference between the two data sets such that it creates a two-dimensional plane with various possible colors for each pixel while the luminance adds depth to the color space as seen in figure 3. [11]

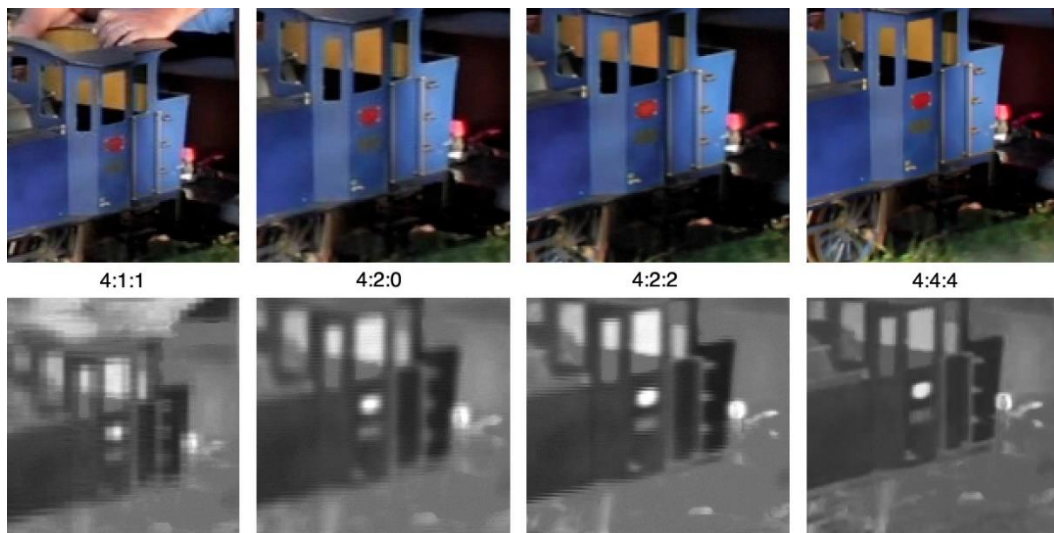
As normalized RGB pixel format is limited by  $0 < R, G, \text{ and } B < 1$ . It is apparent that the YUV color space is much larger than YUV such that RGB conversion limits the color space when converted into RGB color space and later displayed in an RGB monitor.



**Figure 3: RGB Color Space Encapsulated by the YUV Color Space [11]**

For all intents and purposes, the conversion from YUV to RGB is dependent on a limiting factor,  $k$ , such that any chroma values for pixels that extended out of the RGB color space get

converted to a value on the surface of the RGB color space shown in the previous figure. This limiting factor causes quantitative data to also be limited and some activity seen in the nanoscopic images to be dismissed and cause artifacts/topping out as well within the two chroma datasets. This can also be categorized as a lossy compression as the transformation as data and differences between data points are deleted. This appears to be overlooked as many commercial sensors and such utilize YUV and are then converted to RGB later to be displayed on the monitor. There are many various conversion formulas to go from YUV to RGB format in relation to the YUV subsampling and the RGB format desired. There are various RGB formats that allow a larger color gamut and possible colors. [11,12]



**Figure 4: Example of Chroma Subsampling Schemes**

This image displays four various subsampling schemes and their respective grey scale image. As less information is stored, edges in the grey scale images are blurred. [13]

## YUV 4:2:0

One of the formats of the videos taken is in YUV420 where Y represents the luma which is independent of the two chroma channels, U and V, or the color difference of the captured image. YUV is utilized to decrease the amount of data necessary to determine the chrominance components of the image which in turn grants the ability of color artifacts to appear in certain images especially when converting original image data in YUV to RGB. The conversion from YUV420 to RGB is as followed where the B-Y (U) and R-Y (V) equations are derived from the first equation:

$$Y = 0.299R + 0.587G + 0.114B$$

$$B - Y = 0.886B - 0.587G - 0.299R$$

$$R - Y = 0.701R - 0.587G - 0.114B$$

The subsampling used in some of the captured videos is 420p which allows an image's chroma data to be compressed and as a result allow a higher frame speed for lower quality sensors as less data is required to be transferred from the sensor to the computer/software. The values are arranged in 3 different data sets, luma, Y, values first followed by U then V. The number of Y components is equal to the number of pixels in the image, i.e, pixel width times pixel length of the image. The U and V components are a fourth of the number of Y components. The subsampling seen in the U and V components allows for higher rates due to compression and is done so since the human eye identifies luminance difference much better than chroma differences and thus is efficient for qualitative viewing of images and video. [11,14]

Figure 5 illustrates how these data sets are saved. Consequently, this causes color-based artifacts and color imprecision within the videos captured. YUV444 is another YUV format that has a color space larger than RGB but does not have the subsampling seen in 420.

Single Frame YUV420:



Position in byte stream:



**Figure 5: YUV420 Subsampling**

[15] Figure 5: Chroma subsampling is shown where a luminance value is determined for each pixel while only one U and V, color difference, value is collected per 2x2 pixel region. This decreases the amount of data that needs to be collected but increases color inaccuracy of the image.

Chroma subsampling is a method developed to lower the amount of data required to produce images that appear good enough for human perception by lowering the amount of data collected for chroma as the human visual system has a harder time determining chroma difference than differences in luminance, contrast. This also explains why luminance/contrast enhancements are more beneficial than chroma enhancements. Chroma subsampling is generally described by a 3 part numerical sequence, a:b:c, where a represents the horizontal sampling, b, the number of chrominance samples (U,V) for the first row of a pixels and c as the number of changes of chrominance samples between the 1<sup>st</sup> and 2<sup>nd</sup> a row. Thus, 4:4:4 would represent two chroma values assigned to each luma value and contain the maximum chroma resolution. This also creates larger video files and require more expensive sensors and material to record at higher frame rates. The camera setup used for older video captures and saves were chroma subsampled in 4:2:0 as displayed in figure 5[14]

ffmpeg -i A002\_1.wmv (to get information on video name\_of\_file.file\_extention)

```
Input #0, asf, from 'A002_1.wmv':
Metadata:
  WMFSDKNeeded      : 0.0.0.0000
  DeviceConformanceTemplate: @
  WMFSDKVersion     : 11.0.5721.5275
  IsVBR             : 0
  encoder           : Lavf57.72.101
Duration: 00:01:02.13, start: 0.000000, bitrate: 2667 kb/s
Stream #0:0(eng): Video: wmv1 (WMV1 / 0x31564D57), yuv420p, 1280x960, 5472 kb/s, 15 fps, 15 tbr, 1k tbn, 1k tbc
```

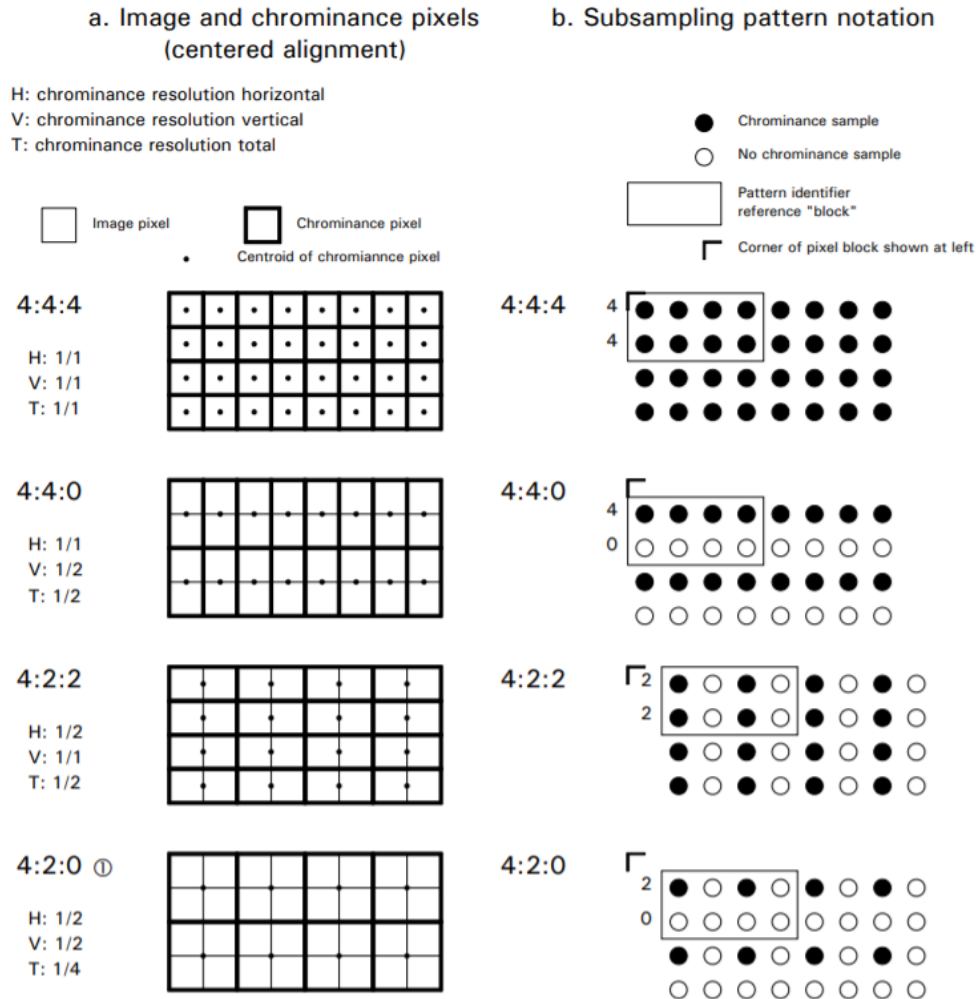
**Figure 6: FFMPEG Retrieve Input Data Details**

As seen in the figure 6, most of the videos initially recorded were in wmv, windows media video, format and in the YUV420p color space. This information will be discussed in later chapters as .wmv is not compatible in ImageJ or Fiji and necessary procedures are required to further process video micrographs.

## **YUV 4:2:2**

The newer (dino-lite) digital microscopes that have been adapted for our system use YUV 4:2:2 format which is different from the previously mentioned YUV 4:2:0 in that it provides more chroma data than the previously mentioned digital microscopes. The 4:2:2 represent subsampling information and informs that half of the horizontal resolution is lost while obtaining the full vertical resolution which is another loss of data relating to the sensor. Like the YUV 4:2:0, YUV 4:2:2 contains more chroma data and is an improvement when trying to qualitatively identify proteins, and cellular dynamics by allowing more resolution for color. This seems strange as the reason in which we can see color at such a magnification has yet to be determined but has created reason to believe that it is due to light scattering, absorption amongst other things that give us a wavelength difference that is then converted to a pseudo-color based off of the difference along the 2-D image. As seen in figure 7, YUV 4:2:2 has twice as much

chroma data but still lags in terms of chroma resolution than YUV4:4:4 and YUV 4:4:0 format. [16]



**Figure 7: Kerr Illustration of YUV Subsampling**

Figure7: Kerr illustrated the 4 main YUV subsampling patterns that assist in realizing what kind of data is lost when utilizing certain chroma subsampling algorithms. [16]

One of the major benefits of using YUV 4:2:2 color space is that ImageJ natively supports some YUV 4:2:2 compressed formats in an AVI container and would then allow the possibility to directly record the nanographs in a larger color space than RGB while also eliminating the need for conversion by using vfwcap or dshow. As seen in figure 8, the more

modern dino-lite capture devices have multiple options in video recording; however, have fps limitations in the highest resolution possible mostly due to the data transfer rates required and sophistication of the sensor being used.

```
[dshow @ 0000000001c6800] DirectShow video devices (some may be both video and audio devices)
[dshow @ 0000000001c6800] "Dino-Lite Premier"
[dshow @ 0000000001c6800] Alternative name "@device_pnp_\\?\usb#vid_a168&pid_0842&mi-00#6&2ec0c282&0&0000#[65e8773d-8f56-11d0-a3b9-00a0c9223196]\global"
[dshow @ 0000000001c6800] DirectShow audio devices
[dshow @ 0000000001c6800] Could not enumerate audio only devices (or none found).
dummy: Immediate exit requested

C:\Users\Michael\Desktop\ffmpeg\bin>ffmpeg -list_devices true -f dshow -i dummy

[dshow @ 0000000006a6800] DirectShow video device options (from video devices)
[dshow @ 0000000006a6800] Pin "Capture" (alternative pin name "0")
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=640x480 fps=5 max s=640x480 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=640x480 fps=5 max s=640x480 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=352x288 fps=5 max s=352x288 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=352x288 fps=5 max s=352x288 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=320x240 fps=5 max s=320x240 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=320x240 fps=5 max s=320x240 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=176x144 fps=5 max s=176x144 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=176x144 fps=5 max s=176x144 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=160x120 fps=5 max s=160x120 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=160x120 fps=5 max s=160x120 fps=30
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=1280x960 fps=5 max s=1280x960 fps=9
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=1280x960 fps=5 max s=1280x960 fps=9
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=1280x1024 fps=5 max s=1280x1024 fps=9
[dshow @ 0000000006a6800] pixel_format=yuyv422 min s=1280x1024 fps=5 max s=1280x1024 fps=9
video=Dino-Lite Premier: Immediate exit requested
```

**Figure 8: List Options Results using FFmpeg**

Figure 8: The following code retrieves a list of available devices as seen in the top most figure “ffmpeg -list\_devices true -f dshow -i dummy” [8]

Once the identifying name of the device is retrieved, the following code shows all possible capture methods including min and max fps and resolution combinations as well as available colorspaces “ffmpeg -f dshow -list\_options true -i video="Dino-Lite Premier"

The dino-lite software that is given with the device automatically saves recordings into an mp4 container without other options which is a source of compression and respectfully artifacts. Being able to choose which container to save in as well as other options such as resolution and fps, allows more freedom for the end user such that better qualitative and eventually quantities data can be retrieved by understanding the desired outcome of the recorded videos as well as reducing the potential for compression and their consequential artifacts. The dino-lite software is

the source of the mp4 compression and therefore making the initial save as uncompressed and placing it into a lossless video container, AVI, manually allows for uncompressed data to be saved. This also allows us to reduce the need to convert our data to be analyzed in most software currently used for bioinformatics.

## **Compression**

Image compression is typical in most commercial products as it allows for higher frame rates and smaller data files. These algorithms typically use visual perception to intelligently eliminate the amount of data that must be stored to provide an adequate image. [21] Lossy and lossless compression are the main types. Lossless compression, BMP and Tiff, allow for the recollection of the lost data while lossy compression is underivable. Lossy is used file formats such as wmv and mp4 which also causes compression artifacts due to chroma subsampling and reduced color space. It is best to save scientific images in a lossless format such that artifacts are reduced and data is maintained; however, this creates a need for enormous data storage when collecting data as lossless and uncompressed images are many times larger than compressed images/videos.

## **Containers**

A container is analogous to a language as it determines the format in which data and different elements of the data and metadata is saved in a computer file. There are many different types of containers developed and designed during different points in time that have their pros, cons, and specific utilities. The major containers important for this are wmv, avi and mp4. These 3 are the three main containers used in the videos captured in the lab. MP4 and WMV rely on compression methods that are patented and controlled by various entities. These compression



techniques are great when the captured data is to be used to stream and display over the internet as it reduces the amount of data needed to store the videos as well as decrease the bitrate to stream the videos. This is beneficial in some respects as computational space and internet bandwidth is limited for scientists around the world. [17]

AVI container is the most intriguing container such that it is the container that is native to ImageJ which means that saving in such a container saves time as well as reduced the total processing required to import the collected data into ImageJ. AVI, audio video interweave, is a great container for holding scientific images as it can be saved in a lossless way such that it has minimal compression and all compression that is performed to the video can be reversed to get the maximum quality possible. [18] This creates another problem that must be addressed as the file formats in which the original data is captured is not compatible with ImageJ. This results in an altered method of capture or additional processing which will be discussed in later sections.

### **Ethical Guidelines for Manipulation of Scientific Digital Images**

With advancements to editing tools and programs that videos and images can be manipulated and twisted in ways that are unscientific and misleading. It is of importance that proper handling and processing of videos and images are pursued such that artifacts are not amplified while the scientific integrity of images is kept. Many guidelines have been published and recommended yet there is not an exact rule book as transformational algorithms and processes for images are constantly being created and utilized. Most guidelines to follow are published by the journals that accept image data, the universities that deal with scientific imaging as well as various publications that try to guide other scientists towards ethical practices. This

section will be a quick review on how to keep the integrity of scientific images while processing and altering the original data.

The paper “Avoiding Twisted Pixels: Ethical Guidelines for the Appropriate Use and Manipulation of Scientific Digital Images’ by Mr. Douglas W. Cromey provides one of the best guidelines for manipulation of scientific digital images. As there is no strong foundation in bioimage processing within the University, it is important to fully understand what can and cannot be done as well as which processes lie within the gray area of scientific integrity as different journals allow some algorithms while others do not.

The most troubling yet smallest problem is intentional falsification of images where the unethical manipulation is intentional such that the scientist manipulates the data in a way that it appears legitimate while it is indeed false and fabricated to fool other scientists and members of the community. The bigger problem is unintentional yet inappropriate manipulation due to the ignorance of the scientist pushing the data for publication. This as adversely led to the increase of image manipulation guidelines by various journals and publications. A way in determining whether a manipulation will be accepted in a journal is to view previous articles and publications that have utilized similar algorithms and processes to determine whether the manipulation is viable within the desired journal. It is also crucial that all processing is logged and stated such that anyone can take the original data and follow the steps specified by the scientist and retrieve an equivalent output. [19]

---

### **Ethical guidelines for the appropriate use and manipulation of scientific digital images**

---

1. Scientific digital images are data that can be compromised by inappropriate manipulations.
  2. Manipulation of digital images should only be performed on a copy of the unprocessed image data file (*Always keep the original data file safe and unchanged!*).
  3. Simple adjustments to the entire image are usually acceptable.
  4. Cropping an image is usually acceptable.
  5. Digital images that will be compared to one another should be acquired under identical conditions, and any post-acquisition image processing should also be identical.
  6. Manipulations that are specific to one area of an image and are not performed on other areas are questionable.
  7. Use of software filters to improve image quality is usually not recommended for biological images.
  8. Cloning or copying objects into a digital image, from other parts of the same image or from a different image, is very questionable.
  9. Intensity measurements should be performed on uniformly processed image data, and the data should be calibrated to a known standard.
  10. Avoid the use of lossy compression.
  11. Magnification and resolution are important.
  12. Be careful when changing the size (in pixels) of a digital image.
- 

### **Figure 9: Ethical Guidelines to Scientific Image Manipulation [19]**

As discussed previously, it is crucial to only process copies of the original file and never inadvertently delete the original files as all procedures and processes performed to acquire the published output file must have the ability to be reproduced by any other scientist or entity. Also discussed previously, it is important to be able to convert the original file into a universally accepted file format so that others can import and analyze. The most accepted in the scientific community is the TIFF format which can be converted in ImageJ from an uncompressed AVI file. This is a reason in which AVI was chosen as the desired container for saving future video nanographs. Color Space is not really covered and there are little specifications on which color

space should be utilized in scientific images most which depend on later processing requirements; however, RGB is the universal color spaced used for PC monitors and other output devices. Simple procedures such as a Histogram stretch, used in the processing of the video nanographs in this paper, are seen as simple adjustments and are relatively save to do as long as there is no clipping, values that get piled up in the 0 or 255 bin of the histogram. A histogram stretch can be done in ImageJ for both individual images as well as image stacks, TIFF files. Histogram equalization procedures which are non-linear must be disclosed in the legend and/or methods as it can alter the relationship between brightness and structure of an image; however, many different types images processed through various histogram equalization algorithms have been published in high level journals. The general dogma is that any non-linear transformation must be reported; however, the safest route is to report every transformation, linear or not, within the legend or methods section to avoid any academic or ethical controversy. It is better safe than sorry as guidelines are different from organization and committee. Again, the importance of keeping track of the protocol is crucial in any image processing done. [19] The procedure of saving at each process or multiple process and altering the new save name to include the processes in order in which they are performed allows for the “protocol” to not only be recorded but also immediately apparent in the file name.

It is also recommended that commercial software where certain processes and filters that are not open source are avoided since the transformation is not entirely known. Many commercial software applications use patented transformations where the actual transformation algorithm is not available to the end user. To avoid misrepresentation of the collected data, it is best to use filters and processes that the end user can examine and understand as some patent filters have the possibility to create artifacts that cannot be identified. This is not to dismiss

possible artifacts caused in using transformations that are known. [19] One transformation that is known but is also to be wary of is subtracting the background as this causes the deletion of certain data to enhance other portions of the acquired data. Background subtraction is used in some of the later images; however, it is important to show the way in which the background is subtracted as well as display the background image that is subtracted from the video nanographs.

Some of the guidelines were not followed in previously acquired images prior to the time of the start of my research; however, the images and videos captured are still of significant importance's as they are in a magnification that has not been reached before as well as well as qualitatively display dynamics not previously seen in the scientific community. The rule that was not followed was the desire to avoid lossy compression as some of the older nanographs were saved in mp4 and wmv which utilize lossy compression to decrease the data size. This results in not being able to fully trust some of the qualitative information within these primitive captures that display high resolution and interesting cellular dynamics as some artifacts are clearly due to the compression process done to these videos/images, such as macroblocking and chroma subsampling. These types of artifacts are usually box like in nature and it becomes apparent that in certain videos, these artifacts are not detrimental in being able to qualitatively analyze the videos. This is also the reason in which previously a method of capturing micrographs in a lossless container and method were described. This method/procedure should be improved and utilized in future research to improve the quality and ethicality of the images and videos captured in the lab. For now, the previous videos captured in lossy compression will still be analyzed and enhanced with possibility of compression artifacts in mind as there is still a lot of qualitative information to be obtained from the videos that has not been seen before.

## CHAPTER II

### METHODS

This section will answer the following:

- What are viable methods to process videos that have already been saved in formats incompatible with ImageJ such that they can be imported into ImageJ then processed using the tools available in ImageJ?
- What are viable methods to capture future videos and nanographs using an optical setups similar to the one described in the introduction that are more efficient, less limiting, and provide more detail than the default capture method?
- Various Enhancement Techniques that can enhance the Qualitative nature of captured videos once they are imported into ImageJ.

The main goals are no matter where the starting point is in which the videos are acquired, the best possible quality is kept and carefully prepared for processing in ImageJ. Another goal is to test possible algorithms that will reduce the haze and blurriness of the nanographs as well as allow for qualitative analysis as well as object tracking.

#### **Optical Video Micrographs**

The physical nature of the capture of the nanographs are up to the microscopist (sample, medium, physical magnification, etc.) and are first focused using a physical eyepiece then the

mirror is flipped towards the digital microscope where the final focal plane is chosen by observing the image on the software being used on the PC.

The micrographs were captured using different sensors as well as in different formats by various microscopists. To obtain pertinent information of the original recordings, the videos are placed as an input into FFMPEG (sections 1.2.5) without any actions such that FFMPEG displays data regarding the videos that are not immediately apparent or that Windows does not display in properties, like format, frames per second and most importantly, color space. As seen previously, color space is a limiting factor both by reduced image quality and format as ImageJ utilizes RGB instead of YUV as a default. The only YUV color space that can be imported into ImageJ by default is YUV 4:2:2 in a AVI uncompressed format. The other problem is the container/format the videos are saved in as they undergo conversion or extraction such that they can later be imported into ImageJ.

### **Alternate Image Capture Method**

Videos that were captured using the software that comes with the digital microscope saved in mp4 and wmv file formats which are incompatible with ImageJ. Converting these file formats can cause loss of data as well as additional artifacts as they are numerical operations and rounding is down. Eliminating the need to convert future data collected using the microscopy method eliminates this additional step of conversion that can cause said artifacts.

The experimental capture method that was not been implemented on live cells but was tested using color changing LEDs utilized FFMPEG's vfwcap (can use dshow as well) command and the older Juision sensor. The captured video was set to the same fps as the default capture method but was saved immediately as a lossless AVI instead of an mp4 or wmv file format. This resulted in a larger file with a larger bitrate than the compressed mp4 and converted AVI from

the original mp4. Again, the conversion from mp4 to uncompressed AVI is needed to process video in ImageJ and the method used for doing so will be described later in this section.

The CMOS sensor from a cheap digital microscope purchased from NanoAndMore was fixed directed towards an LED system changing colors. (Juision) The LEDs went through a full color cycle that took roughly one minute. One video was taken using the Windows Camera application in the default compressed mp4 format, converted to an uncompressed AVI then displayed in ImageJ. Another video was taken under similar conditions, facing the LEDs, but was captured using FFPEG's vfwcap and immediately saved as a lossless AVI. The video captured using FFMPEG was larger than that of the converted AVI by 120% as seen on figure 10. It also contained a higher bitrate which generally means higher quality as more data is read per second. The mp4 properties are low due to compression which is widely used to save data and allow for faster transmission and streaming.

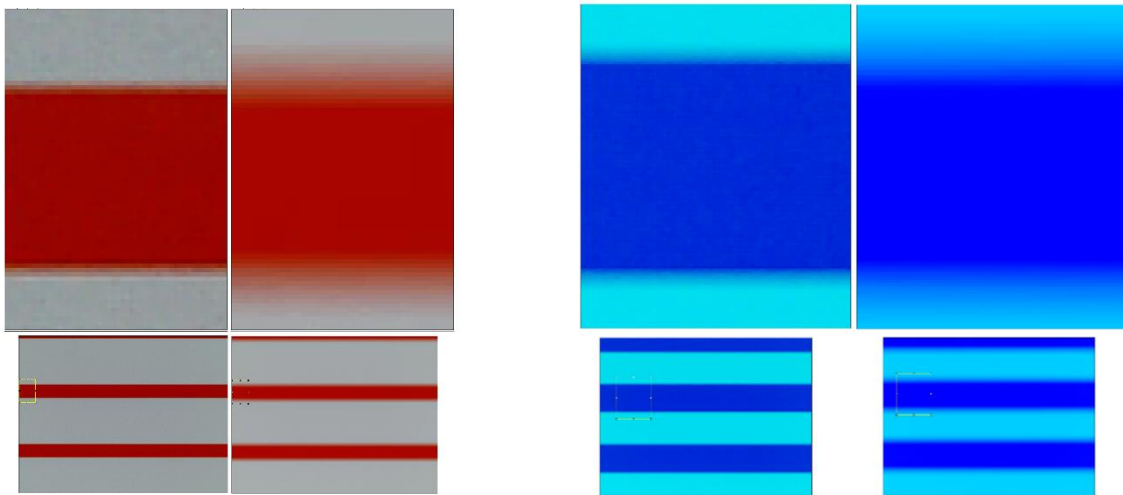


Length	00:00:57	
Frame width	640	
Frame height	480	
Data rate	1740kbps	MP4 Format
Total bitrate	1740kbps	
Frame rate	29 frames/second	
Size:	12.0 MB (12,651,300 bytes)	
Length	00:00:57	
Frame width	640	
Frame height	480	AVI converted from MP4
Data rate	110592kbps	
Total bitrate	110592kbps	
Frame rate	30 frames/second	
Size:	759 MB (796,309,730 bytes)	
Length	00:00:51	
Frame width	640	
Frame height	480	AVI captured using vfwcap
Data rate	290816kbps	
Total bitrate	290816kbps	
Frame rate	29 frames/second	
Size:	916 MB (960,981,070 bytes)	

**Figure 10: Capture Formats and their Respective Properties**

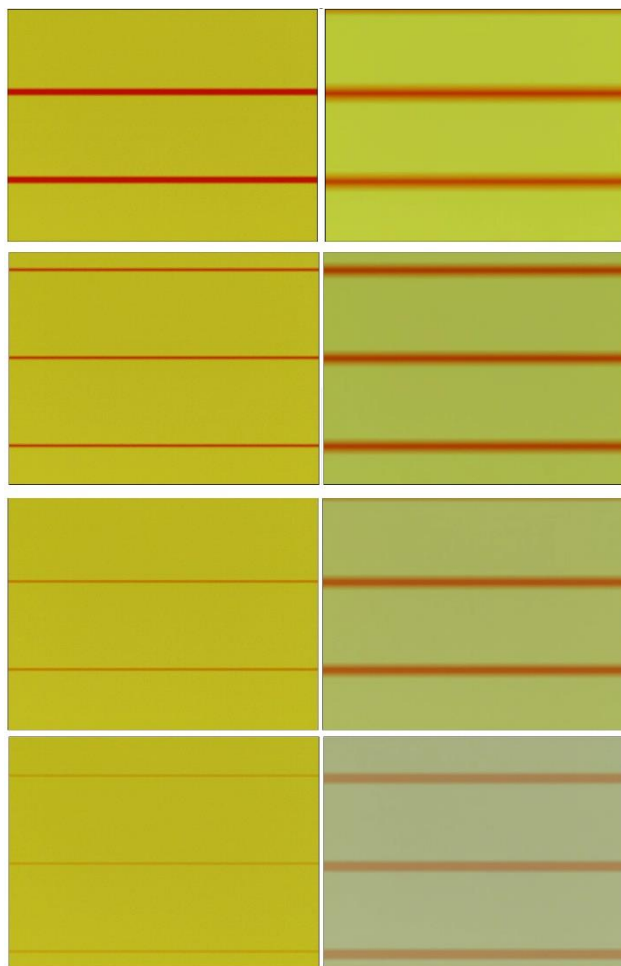
Once the converted MP4 file and vfwcap file were imported into ImageJ, the frames were then aligned so that the visual quality can be examined. As apparent in figures 11 and 12 the visual quality of the video taken using vfwcap is much higher as there is little to no color gradation as compared to the converted file. This allows for sharper frames and more detail to be captured. The only immediate negative for this capture method is the inability to preview the video as it is being captured using FFMPEG which does not allow the microscopist to make spatial changes while recording. There is likely a way to enable a video preview while recording but one was not identified during this experiment. Also, it is noteworthy that some frames were dropped when using this method which is likely due to the sensor trying to keep up to the framerate. It is likely frames were also dropped in the other methods, but the software used in those situations do not inform the user of such as the original fps was 29. This method also

allows the user to see all resolution and framerate possibilities for recording video and gives more options and freedoms than the other video capture methods used. The color bleeding, originally caused by the chroma subsampling, is likely enhanced during the conversion process and initial compression. Typically this color blurriness is dismissed in commercial videos; however, the ability to have sharper images from lower cost sensors proves to be a worthy venture as seen in figure .



**Figure 11: Color Banding Comparison Between Commercial Software and FFmpeg's vfwcap**

Figure11: Top left: zoomed in region of bottom-left. Tight bands are present with less mix of colors and more defined edges from the LED light source. Top-Right: Zoomed in region of bottom-right. Left side is captured using FFmpeg's vfwcap while the right side is captured in mp4 then converted to AVI.



**Figure 12: Color Blending Comparison Between Commercial Software and FFmpeg Vfwcap**

Figure 12: This shows the progression of the red percentage decreasing in the LED yet the converted mp4 files (left hand side) displays an overall degradation of the yellow sections with blurred red lines whereas the AVI save (right side) accurately represents the transition and maintains proper color.

The codec, container, and method used to capture videos and images does make a drastic difference in the quality of the images as seen in figures 11 and 12. Further implementation of this method will be investigated as the lab continues towards recording and capturing live cells once again. As this paper primarily focuses on enhancing the quality of previously captured videos, there has not been an opportunity to test this method for live cells; however, it is a

significant method to implement in the future to improve the quality of future nanoscale micrographs at the point of capture rather than simply relying on post-processing. This method also reduces the need to convert, saves computational time and allows the microscopist more freedom in determining the video properties. The ability to determine and more so lowering the framerate is also important as lower file sizes are then possible when looking at dynamics that take longer to proceed as well as higher image resolutions for each frame. File size of uncompressed or lossless compressed video files are always relatively high when compared to compressed files that are more prone to artifacts; however, this has long been a problem amongst the video engineering community as well as commercially. It is apparent that a lossless AVI compression container that is compatible with ImageJ is the best container to save the nanographs in as it provides the least chance for artifacts as well as eliminates the need for conversion which also causes artifacts as numerical data is rounded and calculated which always causes some degree of inaccuracy. It is then recommended to use FFmpeg to save captured nanographs in a compressed lossless AVI format when possible to avoid multiple sources of artifacts and achieve the best quality of data from the digital microscope sensor. Downfalls as discussed previously include large data files as well as the inability to preview the video being recorded which might have a solution but has not been pursued due to the situational nature of the research performed as the major interest currently is processing older videos which have already been recorded in wmv and mp4 containers. This section is primarily added such that in future ventures, the appropriate capture methods are utilized to prevent loss of quality and elimination of possible artifact sources.

## **FFmpeg ImageJ Plugin**

Prior to detailing the method utilized below, there exists a plugin for Fiji/ImageJ where a video can automatically be converted and played in ImageJ. Comparison between utilizing the plugin and the method below are extremely similar with very few differences from the output. The method below provides a result when the plugin created by Johannes Schindelin does not output results. Using this plugin will result in an estimated size of the video that is decompressed as well as view a virtual stack of the video. Processing cannot be done to this stack unless saved in a native state.

## **FFmpeg to ImageJ**

This section will outline the methods used in regard to pre-existing videos that have been captured and recorded in formats incompatible with ImageJ.

As discussed previously, the original videos captured at the early stages of the microscopy setup were captured in formats that are not compatible in ImageJ. This has led to detailing a simple method of converting the existing files into a format, AVI or image sequence, that is compatible in ImageJ to be processed. FFmpeg is used here and the process from download to utilizing it will be quickly detailed as well as the line of code needed to convert and what the parameters represent.

FFmpeg, available at <https://ffmpeg.org/download.html>, is a free software containing the necessary audio/video codex libraries for conversion and decompression of most video formats. The following steps are written for Windows builds in mind but should work on Linux and Mac OS. The program is typically in a zip file where it is extracted and moved to the C: drive or

another common directory. Locate and entered the folder titled “bin” that contains the ffmpeg application. Relocate the input video file into this folder as well. Remember to save an extra copy of the original file. This folder will have to be located through the command prompt windows application. An example command line for the command prompt to change directories to the ffmpeg-containing folder is:

```
cd desktop/ffmpeg/bin
```

“cd” is changing directories/folder in order to do an action in another folder. Here, the program is located folder is on the desktop.

### **Conversion using FFmpeg**

Once in the folder that contains the ffmpeg program and input video file, the following command is used:

```
ffmpeg -i filename.fmt -pix_fmt fmtname -f avi -vcodec rawvideo output.avi
```

Where, “-i” signifies introduction of the input file. “Filename.fmt” is replaced by the filename of the video to be converted and “fmt” is replaced by the format extension of the file e.g. WMV, MP4, MOV, AVI. Etc. “-pix\_fmt” allows the option to choose the pixel format of the input video. Writing “ffmpeg -pix\_fmts” in the command line and pressing enter grants all names of available pixel formats to replace with “fmtname”. Ommiting “-pix\_fmt fmtname” allows ffmpeg to automatically attempted to determine the pixel format of the input but may not do so correctly. “-f avi” ensures that the output file is saved in AVI format. It is not always necessary and in events where unspecified/omitted it can be assumed by ffmpeg with the file format extension written on the output file in the next step.. “-vcodec rawvideo” sets the video codec for the output file to rawvideo in order to avoid any compression of the output file. “Output” can be changed to the desired name of the converted video. “.avi” is used here because

it is a native format in ImageJ. Once the necessary changes to the command line are correct, hit enter to execute the line. FFMPEG runs the line, converts the input video, and saves an uncompressed AVI file as the output.

This method works well enough such that the output is a video that is compatible with ImageJ. There are other methods of conversion using FFmpeg some of which may very well be more efficient in saving the original quality. The major problem though with this method as well as others is file size. The file size of some videos can reach upwards to 100Gb or more depending on the length and number of frames in the video. This method converts each frame into an uncompressed file format which is a computational burden. However, it is still viable as the AVI file can be imported as a virtual stack then a smaller region can be extracted from the larger file and then processed using any of ImageJ's tools.

### **FFmpeg Frame Extraction**

This method provides more freedom than the method described in 2.3.1 as well as less computational resources to be performed. Instead of converting each frame of an input video, calculating the numerical values and transforming the data into a different video format, this method will view a frame in the input video and save an image of that frame as any format desired. Here the format PNG, Portable Network Graphics, can be chosen which utilizes lossless compression. The ability to choose the number of frames per second desired is granted and beneficial as less frames can be “converted” which reduces the amount of storage required as well as decrease the computational burden. The line of code is relatively simple and is the following:

```
ffmpeg -i input.mp4 -vf fps=1 out%04d.png
```

Here, “out” describes the beginning of the name for the saved images while “%04d” describes how many digits these images can have within their name. For example, image one would be “out0001.png.” Once these images/frames are extracted from the video, they are placed in a folder with the description of their earlier processing. To import into ImageJ, all that needs to be done is to select all images then click and drag image 1 into ImageJ and allow all the images to popup without interacting with the application or importing as an Image sequence and choosing out0001.png. Once this is done, following the basic procedure of Image>Stack>Images to Stack immediately converts the multitude of extracted frames into a time stack in the proper order. This can then be saved as a TIFF file which is the most commonly used file for scientific time stacks. It is a relatively efficient and easy method and allows for longer videos to be analyzed without a large computational burden. It is important to remember how many frames are extracted per second and how that relates to the original speed of the video especially when playing back during presentation.

### **AVI to TIFF vs Snapshots to Time Stack**

If the videos are converted to AVI and then imported into ImageJ, it is recommended to then save the AVI as a TIFF stack as this provides an ease in processing. If the snapshot method is utilized, then this is done when converting the images into a stack. It is also recommended that multiple saves are done at various points in processing in case later irreversible processing is done so that the user does not have to restart. Intermediate steps and their corresponding files can be deleted at later times once unnecessary.



## **Macroinstructions – Recording /Coding**

Once the images are imported into ImageJ and saves as TIFF time stack files, then they can be processed using a various of algorithms and processes as well as be processed as a while using macroinstructions feature. Prior to any processing, it is recommended that a duplicate stack ranging from 20-50 of the frames from the converted video is created to run test processes on to observe the qualitative changes done. This is recommended such that a preview can be obtained prior to commitment in processing the much larger stack which reduces time it takes to receive desirable results.

### **CLAHE as Means for Improving Intensity Contrast**

Contrast limited adaptive histogram equalization created with medical imaging in mind in at its creation has allowed for a much broader range of applications even reaching the forefront of underwater imagery. [2,22,23] A small region of the image is chosen where a histogram is created then equalized. This is a local based enhancement where the size of the region is typically chosen at the discretion of the scientist. This section will describe the algorithm and the information important to such.

## **Histograms**

A histogram displays the frequency in which values occur within a specific range. In images, it allows us to see the common values within the image as well as the dynamic range of the image, 0-255 for the images collected in the case of this study.

$$n = \sum_{i=1}^k m_i$$

Equation # shows the mathematical generalization for a histogram where n is the total number of values observed, k is the number of bins, 256 in the case of most images, and  $m_i$  is the function that represents the histogram.

Typically, low-tier photosensors have histograms with a low dynamic range with respect to the amount of bins, values, available to be assigned. Dynamic range represents the ratio between the largest and smallest values where that images collected from low-tier sensors typically have a much smaller ratio than the 0-255 ratio possible in normal 8-bit colored images,  $2^8$ . This is also directly related to the contrast ratio when only talking about the luminance of an image which compares the brightest color to the darkest color as discussed previously.

### **Histogram Equalization**

Histogram equalization is one of the foundational transformations created to help enhance the contrast of images. The objective of the algorithm is to output a more uniformly distributed histogram which enhances the contrast and dynamic range.

Typically done on the channel that contains the values for luminance, histogram equalization can also be done on chroma channels which provides a color enhancement. To further dissect this algorithm, let an input matrix  $Y$ ,  $i$  by  $j$ , represent an image where  $i$  goes from one to the maximum length of the image and  $j$  from 1 to the maximum width of the image each containing a value ranging from 0 to  $L-1$  where  $L$  is the number of possible values which is often 256 for 8-bit images. The values of the matrix are then normalized using the following equation where  $p(k)$  is the probability of a pixel containing the value of  $k$  from 0 to  $L-1$ .

$$p(k) = \frac{n_k}{n}$$

Where  $n_k$  is the number of pixels with the value of  $k$  and  $n$  is the total number of pixels. The cumulative distribution function, a function that gives a probability that  $k$  will take a value less than or equal to a value  $K$  such that

$$cdf(k) = p(K \leq k)$$

Consequently,

$$cdf(L - 1) = \sum_{K=0}^{L-1} p(K) = \frac{n}{n}$$

Again, the purpose of this histogram equalization is to remap the inputs into a new matrix such that the output has a linearized cumulative distribution function across the range of the image. Therefore, a transformation is necessary that takes each value,  $k$  to  $L-1$ , and remaps it across the typically 256 bins. The transformation is defined as following:

$$f(k) = v_o + (v_{L-1} - v_o) * cdf(k)$$

Where  $v_o$  is the lowest possible value and  $v_{L-1}$  is the highest value, 0-255 for 8-bit images. The final output with remapped intensities for the input matrix,  $Y$ , yields a result such that  $X$  is uniformly distributed on the range of  $[0, L-1]$  where

$$X_{i,j} = f(Y_{i,j})$$

Basic histogram equalization is a global enhancement technique in that it takes the histogram of the entire image and outputs an image with a linearized cumulative distribution function. [24]

### **Adaptive Histogram Equalization**

Adaptive histogram equalization is an improved method of histogram equalization that utilizes a contextual region around a pixel and is considered a local histogram equalization. This algorithm does the same procedures as a histogram equalization, however, locally by obtaining multiple histograms of smaller matrixes and remapping the intensities one pixels at a time for the entire image based off a chosen matrix size which are typically chosen through experimentation based off of the images being processed. Remapping based off local regions allows for better

contrast on images that have some regions much lighter or darker than others. This is beneficial when viewing the nanographs of cells where the membrane has a curvature and regional surfaces of the cell or intercellular dynamics are on a different focal plane and therefore have various intensities that make global transformations ineffective. Pixels along the borders are also treated differently as they do not have neighbors and therefore their remapping is calculated typically by mirroring the surrounding other surrounding pixels and at times this causes obscure intensities on the boarder of the images. An interpolation method also exists such that if one pixel is remapped, its surrounding pixels are remapped as well to save computational time using the following:

$$m(i) = a[bm_{--}(i) + (1 - b)m_{+-}(i)] + [1 - a][bm_{-+}(i) + (1 - b)m_{++}(i)]$$

Where,

$$a = \frac{y - y_-}{y_+ - y_-} \quad \text{and} \quad b = \frac{x - x_-}{x_+ - x_-}$$

This interpolated version increased the speed in which the pixels are remapped, provides faster results and avoids visibility of regional boundaries. It is also apparent that the smaller contextual regions increase computational times and there is a numerical ratio that needs to be identified such that there is a compromise between the details of the features desired to be observed and the amount of time desired to wait for a final. On single images, this compromise is not as important as computers have advanced and even the smallest possible contextual regions do not take much time to process; however, when dealing with larger stacks of images, this comprise is important especially if the processed is to be used in clinical applications where time and resources must be used efficiently. (find time data in using this interpolation method and not using this interpolation method and add)

This algorithm is limited by its ability to increase noise as it does not have a limiting factor in contextual regions that have homogenous intensities or values and therefore creates sharp histogram peaks throughout largely homogenous images. The ability to eliminate this undesired result has been granted by the addition of a contrast amplification limiter in the contrast limited adaptive histogram equalization algorithm. [25]

### **Contrast Limited Histogram Equalization (CLAHE)**

Contrast limited histogram equalization originally created for poor contrasted medical images in mind by Dr. Zuiderveld has provided a strong foundation for localized histogram equalizations where many versions of the algorithm accounting for specific situations have been derived from which makes an adjustment for the noise problem seen in the original adaptive histogram equalization algorithm.

Contrast limited histogram equalization eliminated the noise found in adaptive histogram equalization by allowing a clip limit to the smaller histograms created by the adaptive portion of the previous algorithm. The clip limit introduced in this algorithm allows a limiting factor in homogenous areas so that noise in regions that do not have a lot of activity are not amplified. This clip limit reduced the amount of values allowed in a specific bin in the histogram. In relation to the previous discussion in section 2.4.2,  $n_k$  is limited for every value of  $k$  such that if the number of pixels with the value of  $k$  exceeds the specified limit, the clipped pixels are then evenly distributed amongst the whole histogram and the total intensity count is kept identical. The reason for this is that background noise typically takes up a wide range of bins, a normal histogram equalization would not utilize these bins, while this process allows some pixels located in local regions to use these bins (range) and provide local contrast enhancements. This is beneficial when viewing cellular activity at the scale in which dynamics are viewed, in relation

to pixel size, since most dynamics and concurrently intensity values do not affect each other beyond a couple of microns. Utilizing this algorithm and choosing a contextual region and an efficient clip limit allows the ability to enhance contrast in the localized regions in which subcellular interactions takes place and allows the ability to observe interactional dynamics that were not as apparent previously.

As shown in Dr. Zuiderveld's initial paper detailing contrast limited adaptive histogram equalization downfalls, there is no 1 to 1 relationship between the values of the output and the input images and does not allow for promising quantitative measurements and all assumptions should be made based off physical meanings of image intensity. This does not necessarily interfere with the main goal as our lab seeks to follow foreign DNA into the nucleus which is tracking an object rather than identifying an object based off a quantitative, numerical value. Utilizing the contrast limited adaptive histogram equalization algorithm through experimentation, the best possible parameters are determined. [22]

### **Visual Representation of Global Equalization and Local Equalization**

When utilizing low-tier, inexpensive sensors and endoscopes, the dynamic range of the nanographs collected are relatively low. Coupling global and local histogram equalization, HE and CLAHE, allows for a larger dynamic range as well as contrast enriched images. Histogram equalization is done in ImageJ with standard parameters and saved accordingly. Contrast limited adaptive histogram equalization parameters are chosen experimentally. Typically, features that are desired to be observed are chosen and sized based off a square pixel region, or block size in ImageJ, surrounding the featured region. For example, when viewing integral proteins on the surface of the cell, a square will be sized around the proteins and its surroundings and the

number of pixels taken up by that space will be chosen for the block size parameter in ImageJ. Again, this is experimentally determined, and the information is qualitative such that multiple block sizes should be tried before choosing the best one to process multiple images with.

### **Filters**

Various filters were utilized to further enhance the visualized dynamics. Filters such as low, high and band pass filters are quite known whereas there are many other signal processing filters that are much more obscured and designed and implemented for various types of signals. Filters are utilized to remove or suppress some unwanted component of the signal received and are crucial when analyzing noisy signals.

### **Rolling Ball Background Subtraction Algorithm**

A background removal filter originally designed for biomedical imaging and used for images with uneven background lighting was researched. This algorithm is experimental with cellular nanographs as it had not been utilized for images such as the ones processed and analyzed here. The algorithm works on a few basic parameters. A third dimension is created on the 2-D gray scale image at every pixel of the image which creates a surface. A ball with a user chosen radius is created and rolled over the bottom side of the surface. The volume in which this ball can reach is then determined to be the background of the image leaving the finer details. Alternatively, a sliding paraboloid can be utilized in place of a ball which allows for more uneven surfaces to be analyzed as the paraboloid of rotation is approximated as a parabola in 4 directions: x,y, and two 45 degree directions. A light background and dark background subtraction is also available where light is chosen as the background of the image is light scattered by out of focus cellular objects and media. [26,27,28]

For this report, it appeared then when choosing a large rolling ball radius for the sliding paraboloid while separating each channel, R, G and B, resulted in a background shifting gradient background that did not contain any information regarding the individual proteins. Utilizing this algorithm assists in following the dynamics of the cellular proteins. The radius parameters were chosen to be larger than the dimensions of the image while keeping an eye on the created background so that no fine protein details are deleted. This top hat filter creates a color gradient of uneven background lighting that maintains edges while freeing up additional histogram bins.

### **Median Filter**

A median filter was also experimented with after utilizing a Gaussian blur filter and not receiving the best results. A median filter is best when trying to keep edges prominent while reducing noise. A non-linear filter, it runs through each pixel and replaces it with the median of its box neighbors. This filter is a smoothing technique similar to a linear gaussian blur which essentially convolves the image with a Gaussian function. At the scale of the nanographs, a median filter can retain edges of proteins much better while reducing noise as a Gaussian Blur filter. [30,31,32,33]



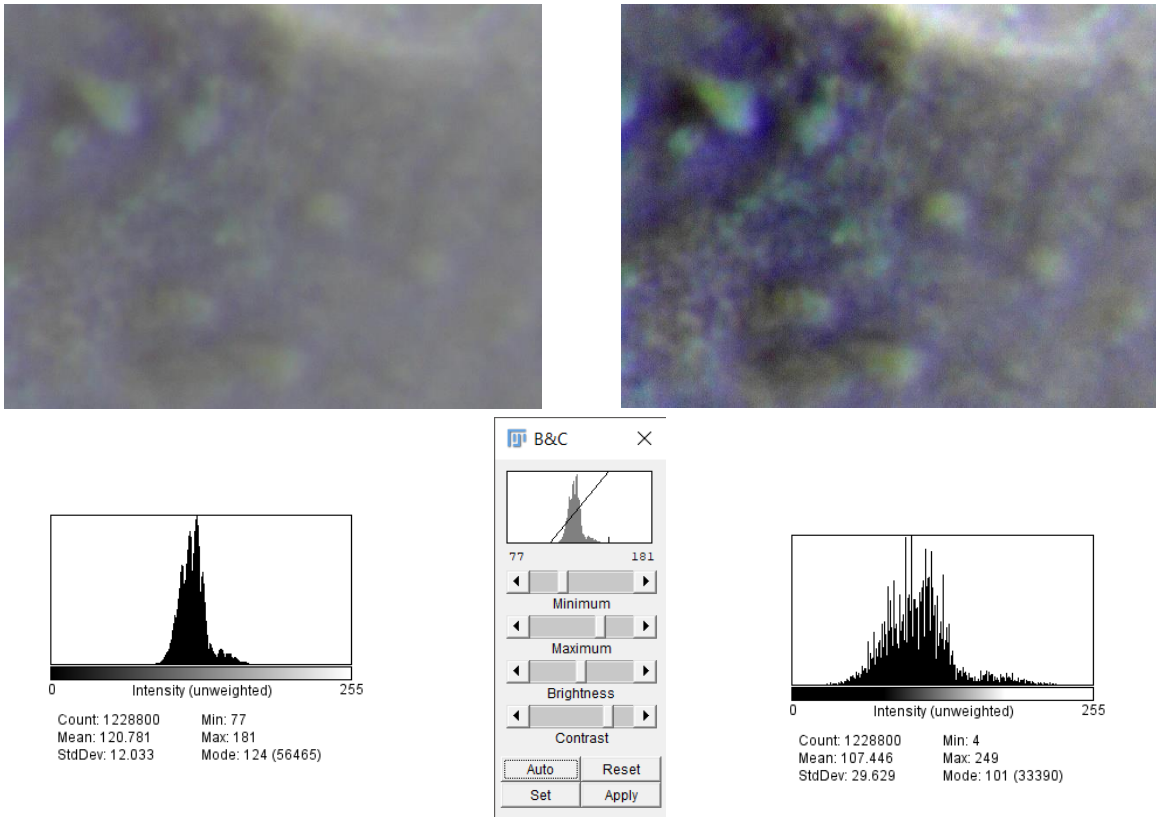
## CHAPTER III

### RESULTS

This section will showcase the aforementioned methods and the results of such. The images here were extracted using the method described in 2.3.2.

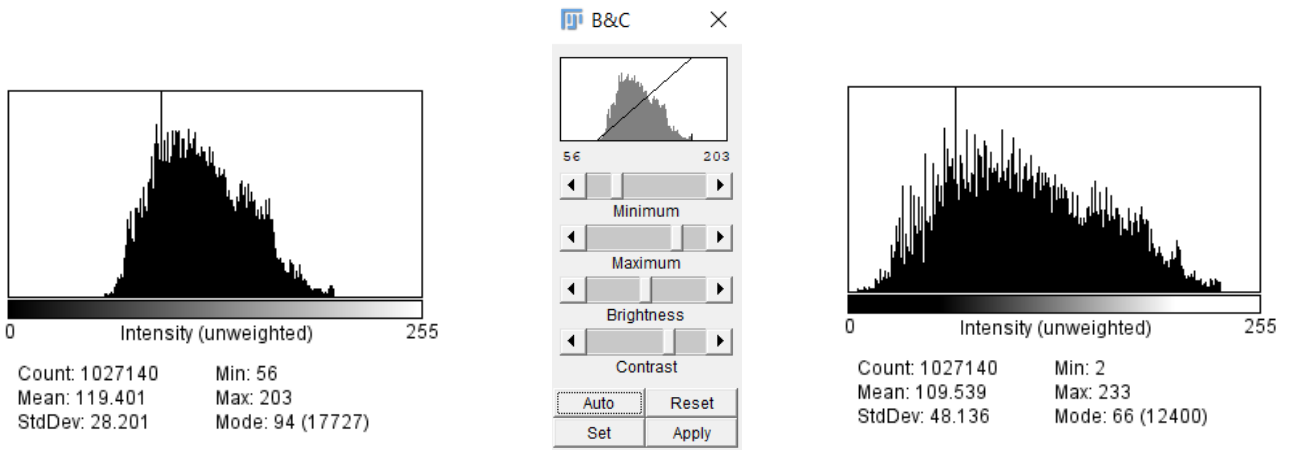
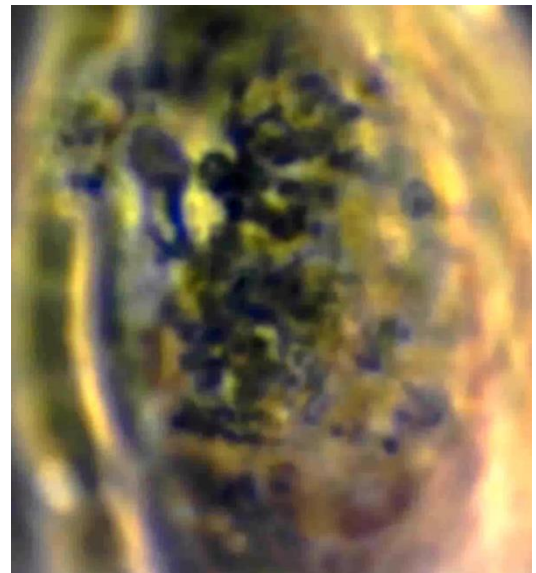
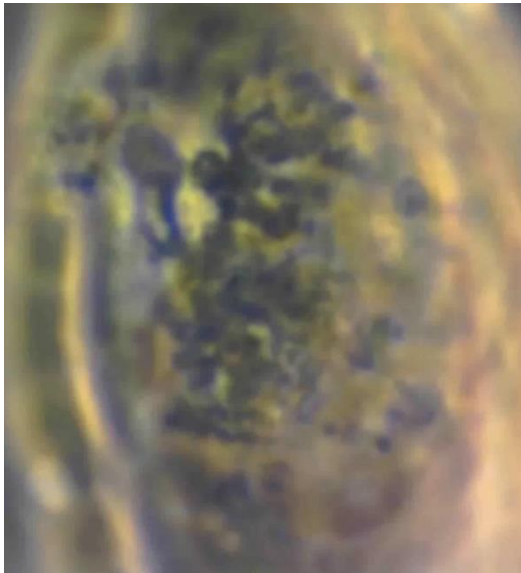
#### **Histogram Stretch**

The image or image stack is selected, and a histogram is created. If processing a stack, it is important to calculate the histogram of the whole stack. The min and max bin of the image is found and stretched linearly such that the new min is close to zero and max close to 25



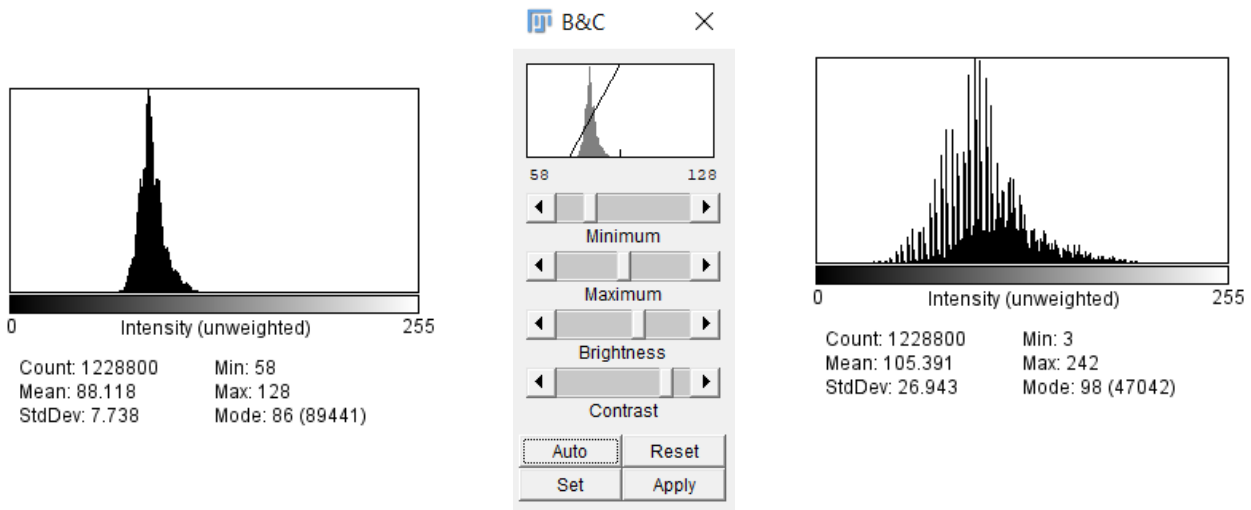
**Figure 13: Proteins on the Surface of a HeLa Cell: Histogram Stretch**

Figure 13: Shows a comparative view after a simple linear transformation of a histogram stretch. The two images as well as histograms are displayed, and details are immediately enhanced with this simple transformation.



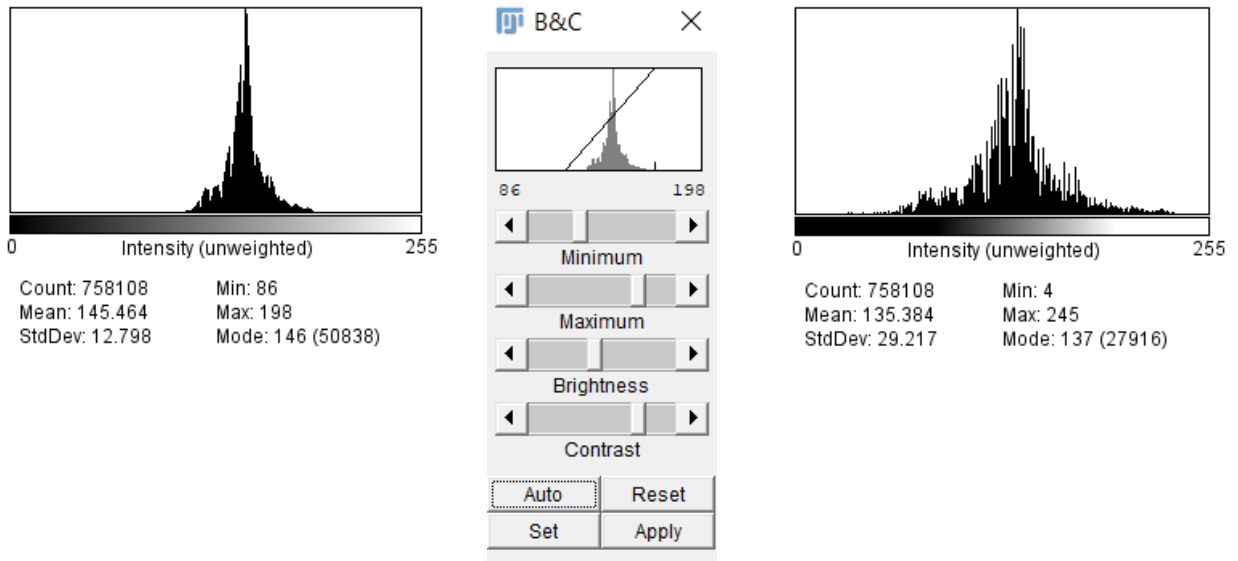
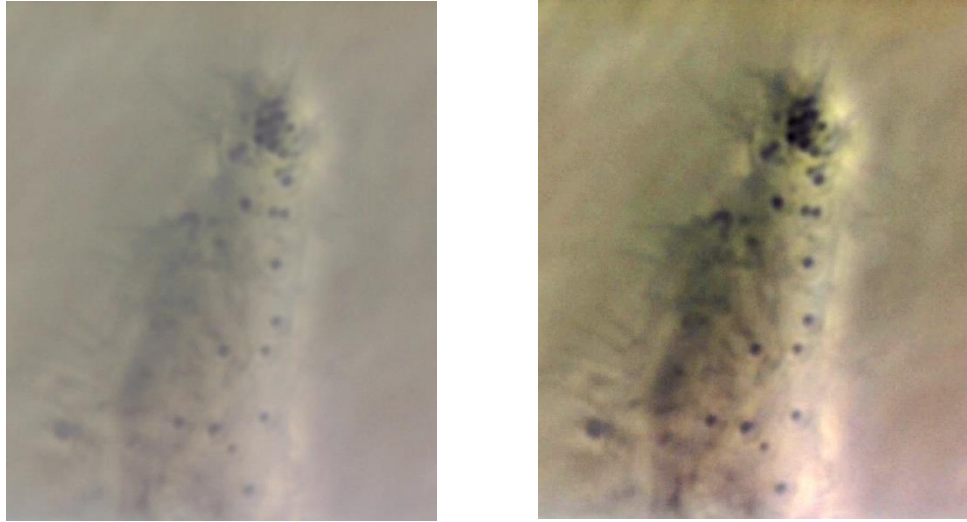
**Figure 14: Into the Nucleus of a HeLa Cell: Histogram Stretch**

Figure 14: Shows a comparative view after a simple linear transformation of a histogram stretch. The two images as well as histograms are displayed, and details are immediately enhanced with this simple transformation.



**Figure 15: Ecoli Grazing on the Surface of a Cell: Histogram Stretch**

Figure 15: Shows a comparative view after a simple linear transformation of a histogram stretch. The two images as well as histograms are displayed, and details are immediately enhanced with this simple transformation.

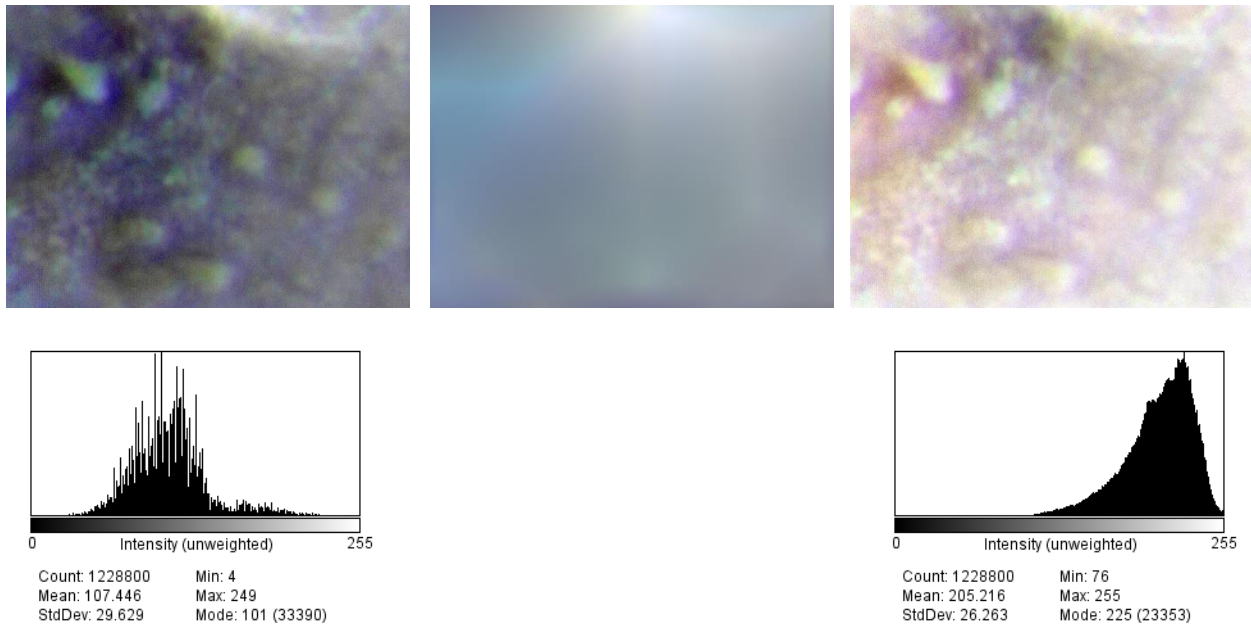


**Figure 16: Viral Particles on the Surface of a Cell: Histogram Stretch**

Figure 16: Shows a comparative view after a simple linear transformation of a histogram stretch. The two images as well as histograms are displayed, and details are immediately enhanced with this simple transformation.

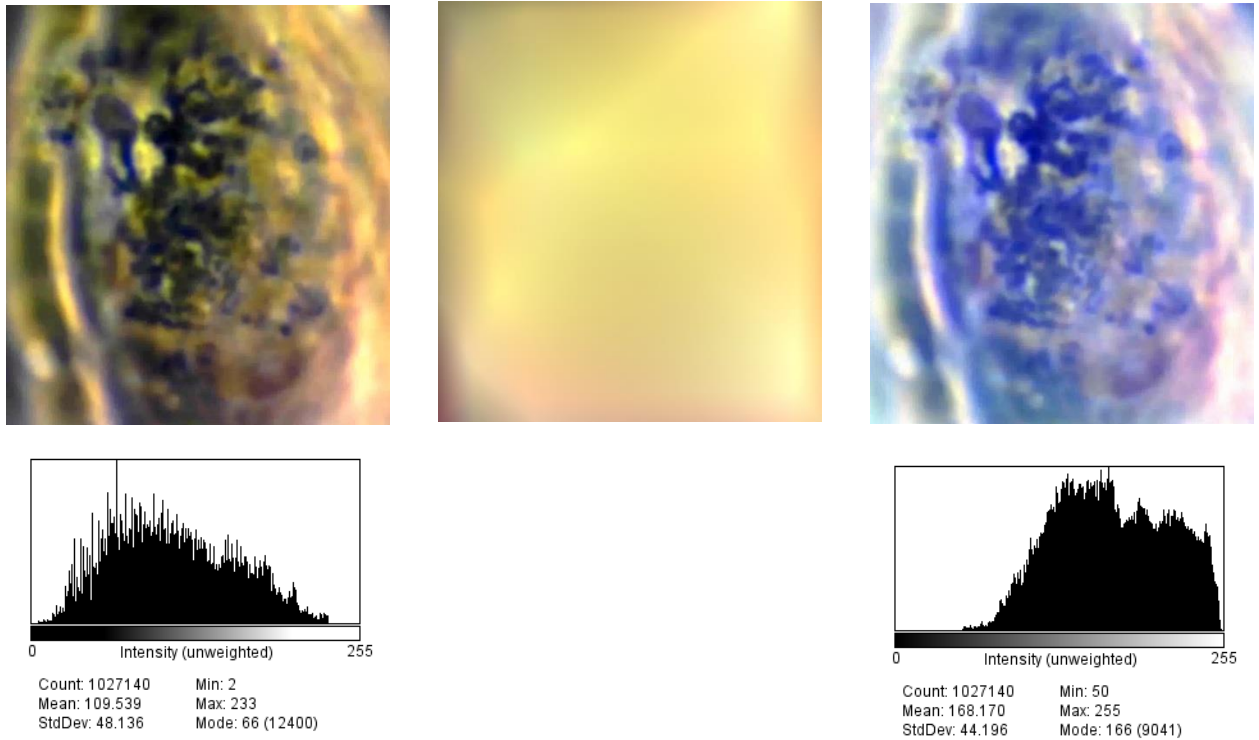
## Removal of Background

The parameters of this algorithm is light/dark background, separate colors, sliding paraboloid, Rolling ball radius and smoothing. Light background is chosen due to the attributes of an inverted microscope. Sliding paraboloid is turned on as well as smoothing to have a gradient background subtracted rather than blotches as this helps maintain edges. Separate colors allow all channels, R, G and B, to be utilized when subtracting the background. A Rolling ball radius greater than two thousand pixels is used to assist in maintaining edges of the proteins.

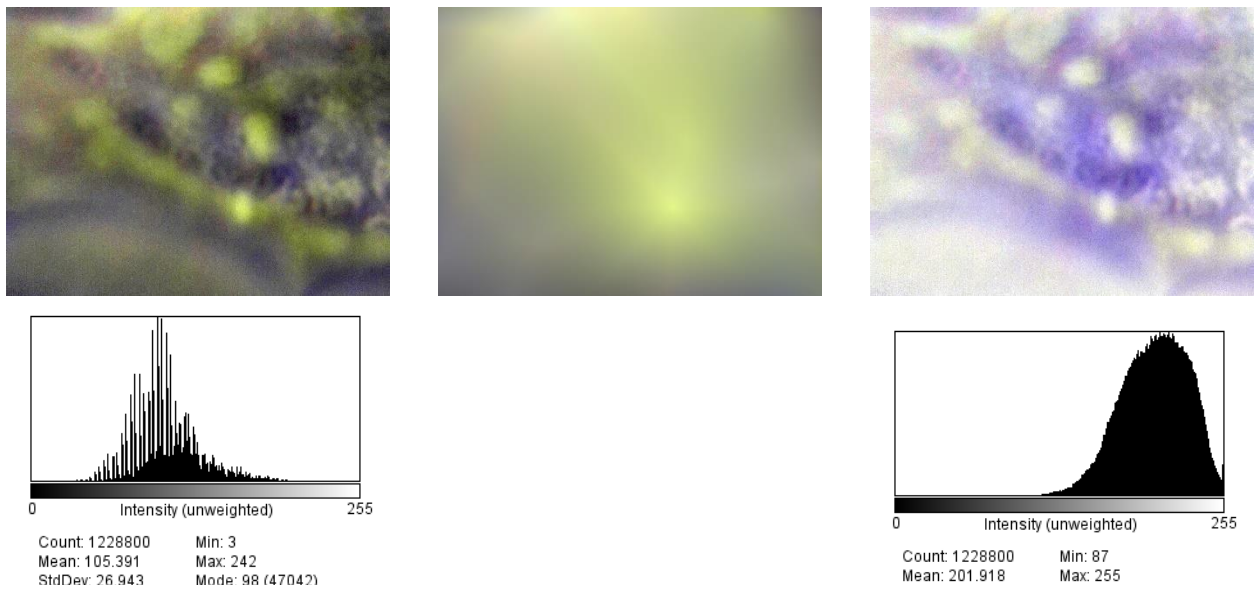


**Figure 17: Proteins on the Surface of a HeLa Cell: Background Subtraction**

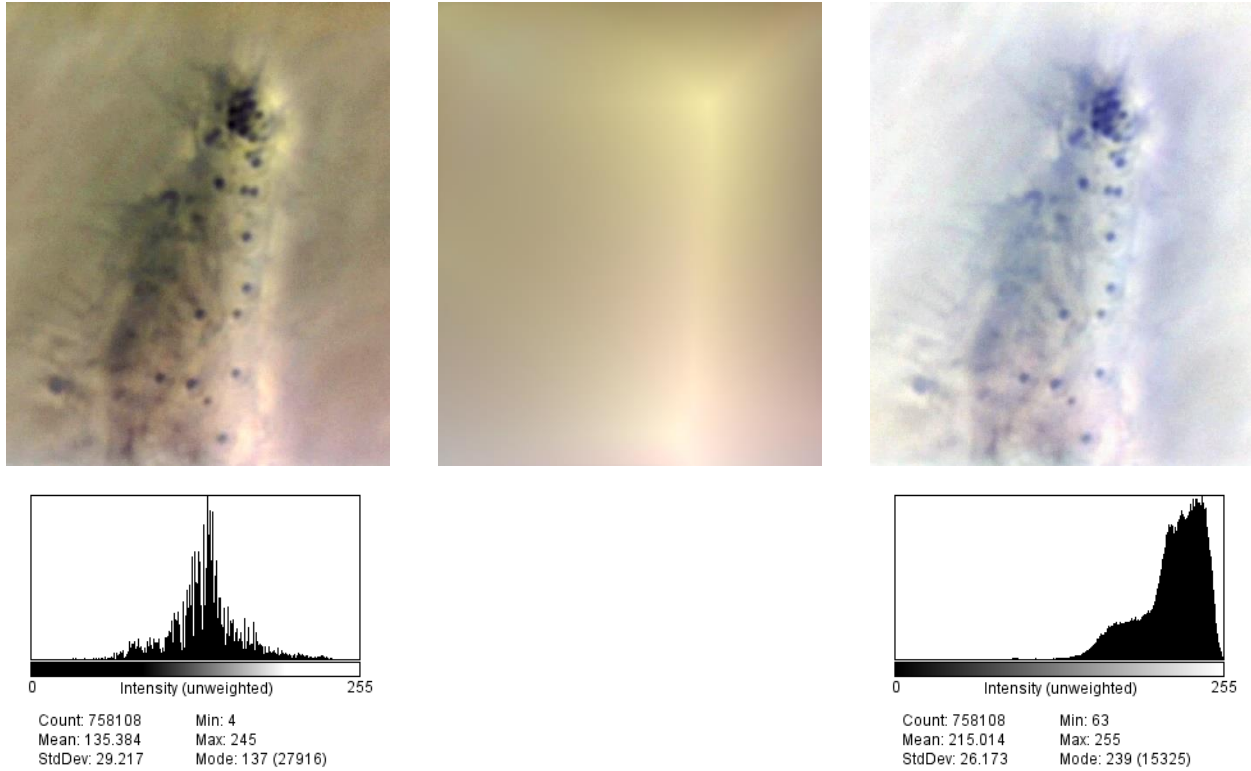
Figure 17-20: Displays the background that is subtracted from the original image. The parameters are chosen based on limiting the amount of fine details that is subtracted. As apparent from the middle image, fine details are maintained while a color gradient is subtracted from the original image.



**Figure 18: Into the Nucleus of a HeLa Cell: Background Subtraction**



**Figure 19: E. coli Grazing on the surface of a Cell: Background Subtraction**

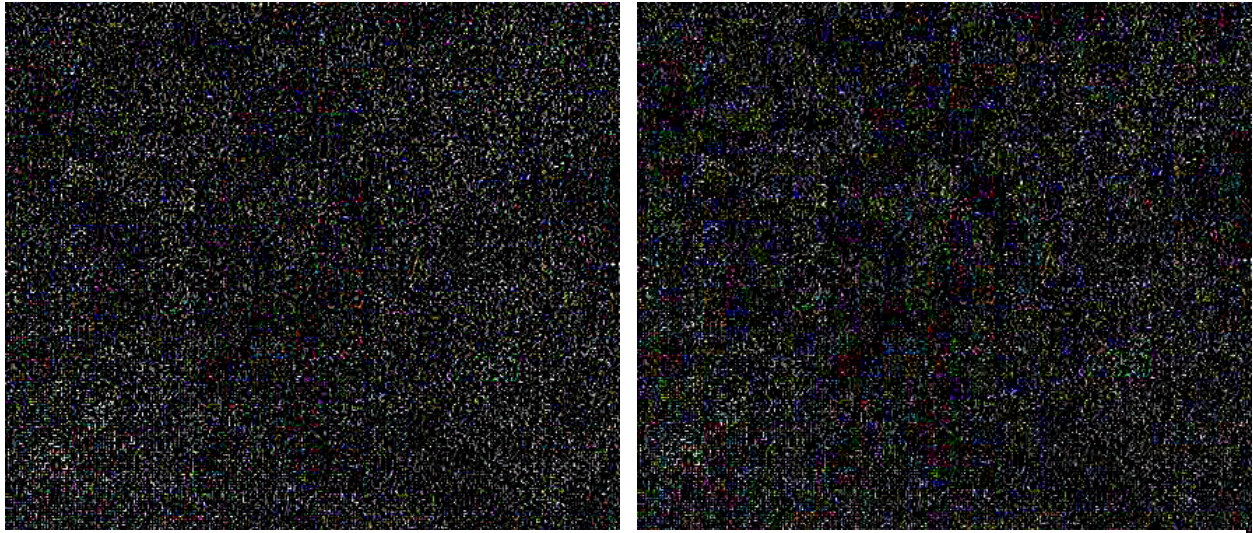


**Figure 20: Viral Particles on the surface of a cell: Background Subtraction**

### Filters (Median/Gaussian)

Various filters were tested where median filters appear to retain edges of proteins much more efficiently. Gaussian filters is also looked at as a good method of denoising. Both Median and Gaussian Filters are smoothing filters and are prone to losing some details as seen in figure 21.



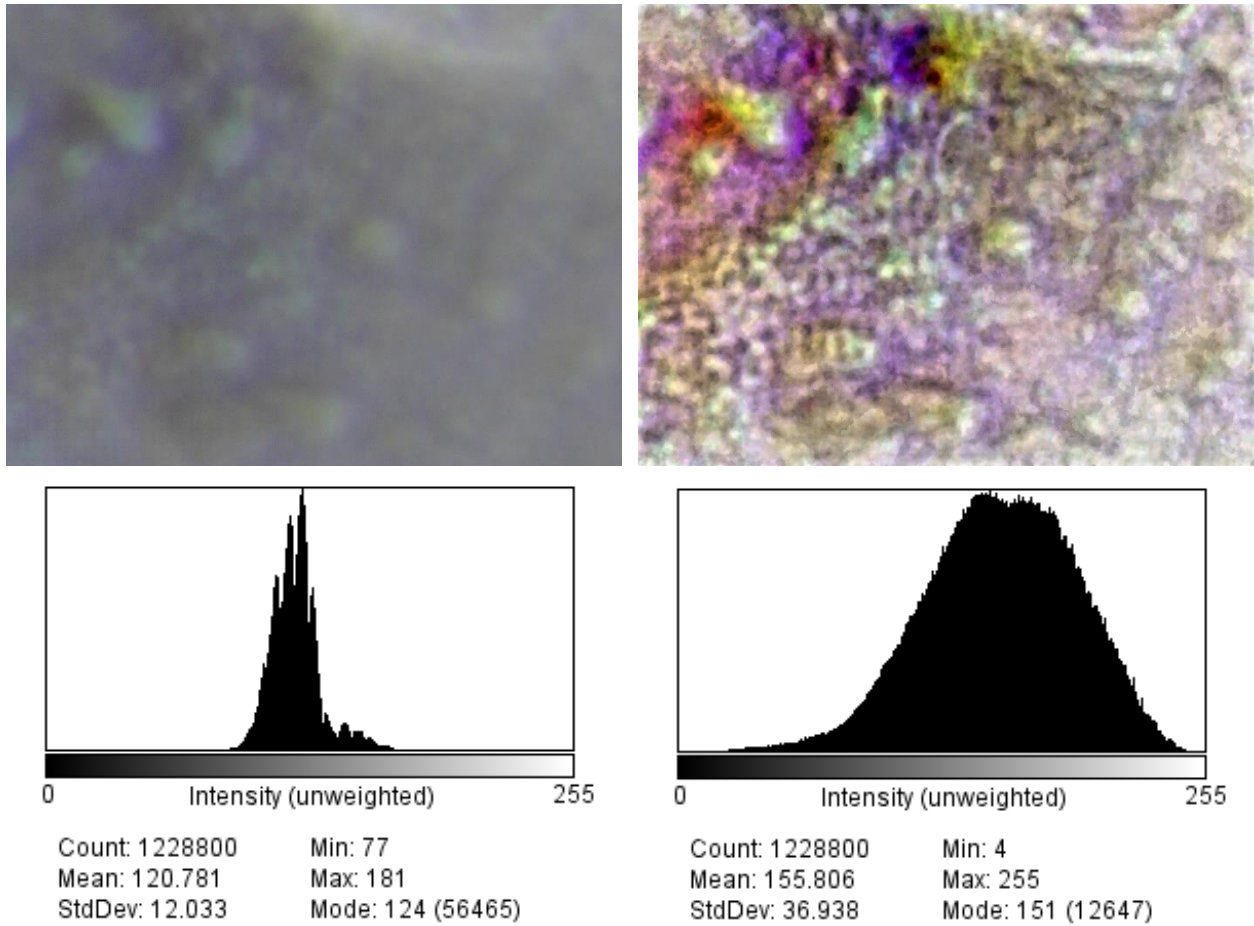


**Figure 21: Comparative View Between Median Filter and Gaussian Filter of an Image**

Figure 21: The image of proteins on the surface of a HeLa cell is processed using two different filters, Gaussian and Median. Left is median filter (pre-filter image minus filtered image) Right is Gaussian (pre-filter minus Gaussian) it is difficult to see that edges are maintained images are cropped to show better. (relook section)

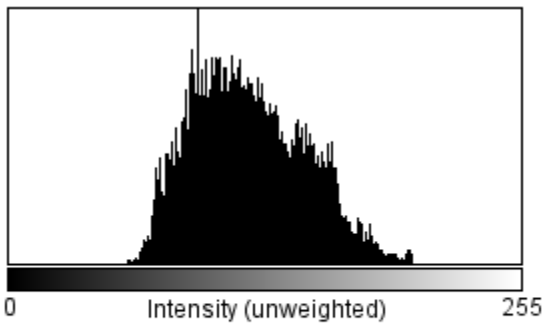
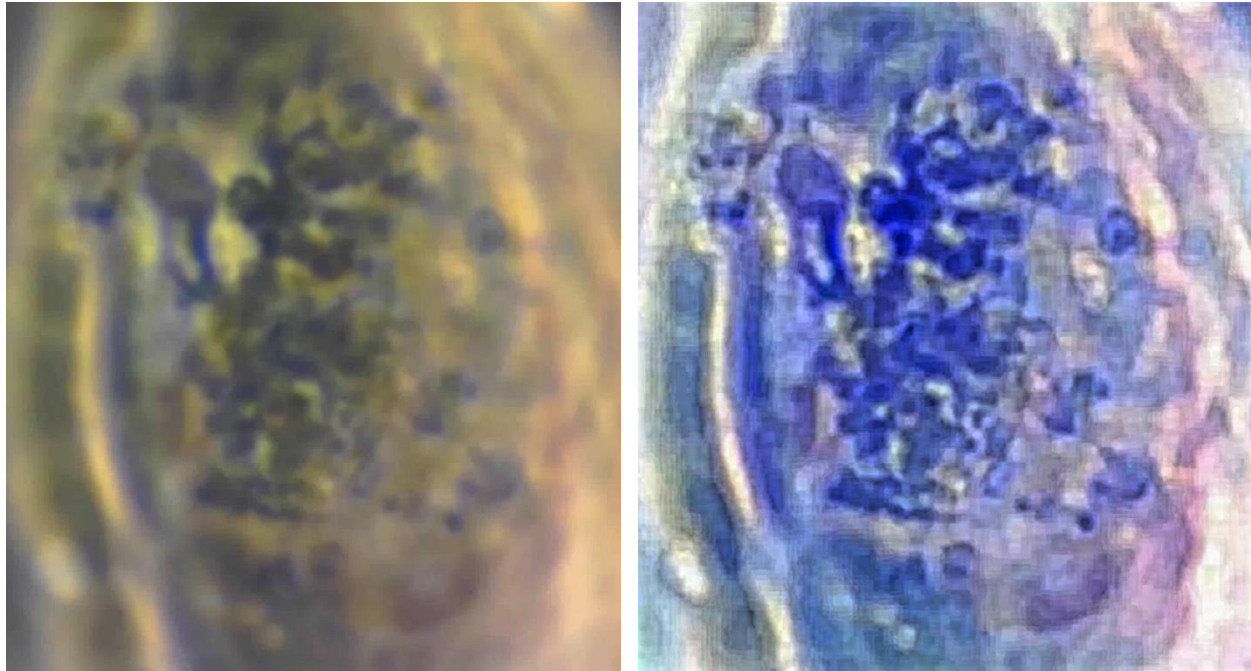
### **Histogram Stretch and CLAHE**

Once the background gradient is removed and noise removed, the final details are enhanced utilizing a final histogram stretch and using the contrast limited adaptive histogram equalization algorithm.

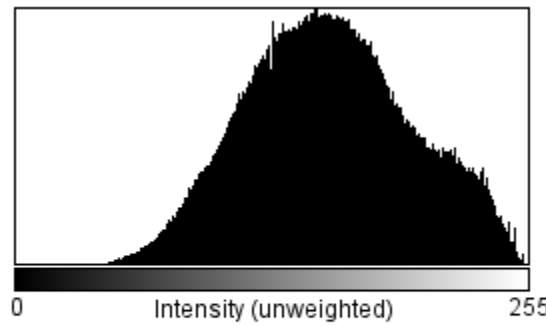


**Figure 22: Proteins on the Surface of a HeLa Cell: CLAHE**

Figure 22-25: Display the original and the final image after the last process of Contrast limited adaptive histogram equalization is utilized. The histograms are also present to display the dynamic range in which finer details are being observed. When images are animated dynamics are widely enhanced.

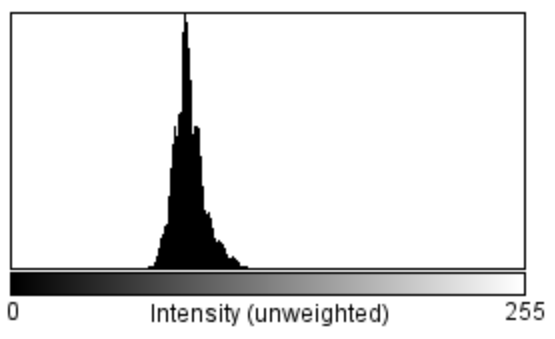
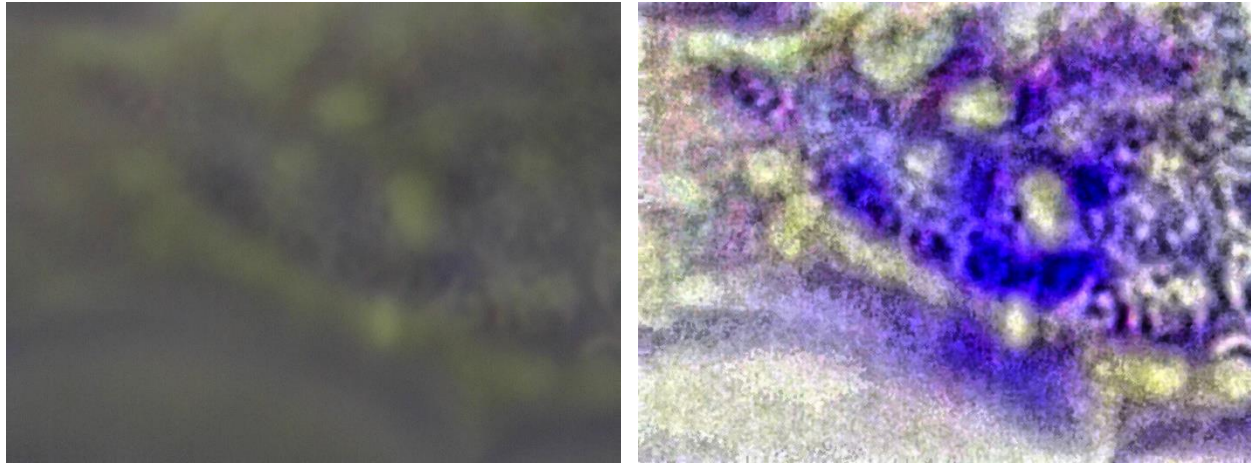


Count: 1027140      Min: 56  
 Mean: 119.401      Max: 203  
 StdDev: 28.201      Mode: 94 (17727)

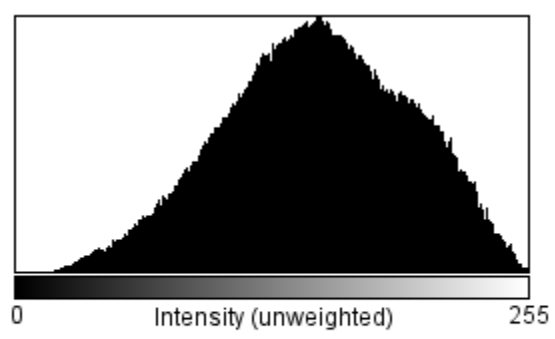


Count: 1027140      Min: 29  
 Mean: 156.108      Max: 254  
 StdDev: 41.297      Mode: 149 (9619)

**Figure 23: Into the Nucleus of a HeLa Cell: CLAHE**

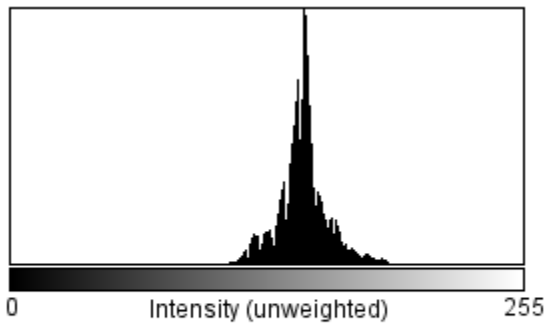
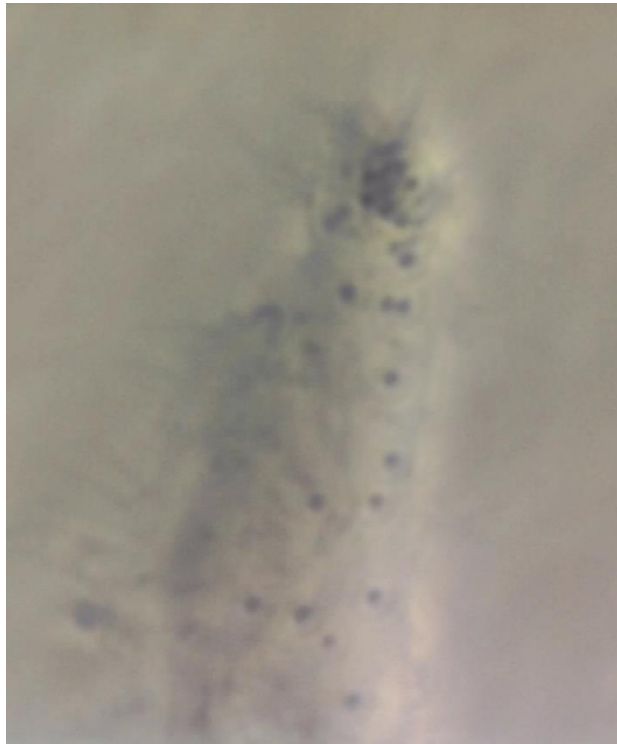


Count: 1228800      Min: 58  
Mean: 88.118        Max: 128  
StdDev: 7.738        Mode: 86 (89441)

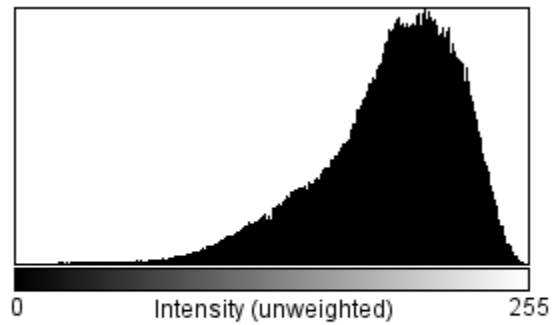


Count: 1228800      Min: 5  
Mean: 149.551        Max: 255  
StdDev: 46.223        Mode: 151 (10423)

**Figure 24: E.coli Grazing on the Surface of a Cell: CLAHE**



Count: 758108      Min: 86  
Mean: 145.464      Max: 198  
StdDev: 12.798      Mode: 146 (50838)



Count: 758108      Min: 5  
Mean: 185.428      Max: 255  
StdDev: 36.448      Mode: 204 (9916)

**Figure 25: Viral Particles on the Surface of a Cell: CLAHE**

## CHAPTER IV

### DISCUSSION

#### **Video Analysis**

As this is a paper report on the process developed for enhancement, it is difficult to illustrate how enhancement assists in being able to observe the dynamics. Contact Dr. Juan Guevara or Dr. Natalia Guevara for copies of the videos processed. Visualization of cellular dynamics and the ability to differentiate between two close objects are greatly enhanced. Utilizing 3D filters such as Gaussian and median caused objects to disappear when they moved rapidly and not enough frames were being used. Fully decompressed 15-30 fps videos are too large to process as well as showcase at conferences and presentations. The nearly 35-minute video containing proteins moving on the surface of a HeLa cell contains 31,598 frames and is 144.6 gigabytes when uncompressed. Utilizing FFMPEG and extracting every nth frame allows for the dynamics to be viewed and processed with much less resources required. This allows for any personal pc to process and visualize video micrographs. Using ImageJ funded by NIH makes the procedures viable and financially efficient to any lab with internet access. Various dynamics were enhanced utilizing the method. Viral particles were seen interacting on the surface of a cell. Proteins moving and interacting on the surface of a HeLa cell. Dynamics within the nucleus of a HeLa cell as well as E.coli grazing on the surface of the cell

## **Capture Method**

Various capture methods were discussed in this paper that will reduce the number of artifacts from the initial capture. FFMPEG is a versatile free tool microscopists should learn as data is now being stored digitally. It can convert, capture, and extract video and their frames. It grants the scientist more options than commercial software allows as well as lists all available formats and color spaces. The ability to choose the pixel format allows less color bleeding as well as crisp images.

## **Histogram Stretch**

Histogram stretch is a common and simple transformation. It allows for all possible bins to be utilized as well as enhance color as well since the images are in the RGB color space. This enhances contrast greatly in images that have low dynamic range. It is performed first as it is a linear transformation and allows the next step to have all bins available to be utilized in the background subtraction. It remaps pixels by taking the minimum and maximum and stretched them to 0 and 255 respectively.

## **Rolling Ball Background Subtraction**

The rolling ball background subtraction algorithm is utilized second. This allows for the subtraction of a color gradient while keeping details. This subtraction does not eliminate edges which are important when trying to track protein dynamics. It is important to remind that this algorithm on a stack of images creates a new background for each image to be subtracted. This allows for the changes in the background to constantly be considered. Color banding also appears in some images due to the gradient subtracted. This is taken care of by filters and is only visible in lower contrast images and zoomed in regions. This subtraction also shifts the

histogram previously stretch as data is subtracted from the original image. This procedure is reversible only if each background subtracted is saved.

### **Filters**

Median and Gaussian filters were primary studied to reduce light speckle noise as well as sensor noise. Gaussian filters were shown to decrease edges while a median filter maintains edges better. Both filters are smoothing filters and details are expected to be lost. The images displayed of the pixel changes after the filter is utilized are greatly enhanced and edges are kept intact while light speckle and sensor noise is greatly reduced.

### **CLAHE**

Contrast limited adaptive equalization algorithm is utilized last as it further enhances edges and contrast. As discussed preciously, it equalizes histograms by regions. These regions are chosen such that it differentiates between various objects which enhance the ability to track dynamics. A histogram stretch is at times utilized prior to this step-in order to increase the enhancement. CLAHE is utilized last as it can enhance noise and is best performed after utilizing a filter.



## CHAPTER V

### CONCLUSION

This process is created to limit computational resources required as well as financial resources. It is developed to enhance previously gathered data as well as an alternate way to capture new data using CMOS sensors. This process equipped with the inexpensive nanoscope previously developed at the University of Texas Rio Grande Valley will expand the possibilities of biological discovery with few financial and computational barriers. With the development of technological advances, imaging science will soon become the forefront of novel discoveries. The generation, collection, modification, and visualization of images is crucial in observing the physical and biological world the human eye cannot see. With these modifications, dynamics are better visualized and allow for previously discarded data to be viable for quantitative analysis.

## REFERENCES

- Romo, Jaime E., Jr. *Optical Method for High Magnification Imaging and Video Recording of Live Cells at Sub-Micron Resolution*, The University of Texas Rio Grande Valley, Ann Arbor, 2016.
- Rajesh kumar Rai , Puran Gour , Balvant Singh, “Underwater Image Segmentation using CLAHE Enhancement and Thresholding”, *International Journal of Emerging Technology and Advanced Engineering* (ISSN 2250-2459, Volume 2, Issue 1, January (2012), pp. 118-123.
- C. Zeiss, “Axiovert 25 / 25 C / 25 CFL Inverted Microscope Operating Manual” Carl Zeiss Microscopy. (1999)
- Dino-Lite Digital Microscopes “AM7013MZT” Dino-Lite Digital Microscopes. Accessed (2017.)
- Dino-Lite Digital Microscopes “AM4515T8” Dino-Lite Digital Microscopes. Accessed (2017).
- Schindelin, J.; Arganda-Carreras, I. & Frise, E. et al. (2012), "Fiji: an open-source platform for biological-image analysis", *Nature methods* 9(7): 676-682, PMID 22743772, doi:10.1038/nmeth.2019
- FFmpeg Developers. (2017). ffmpeg tool (Version be1d324) [Software]. Available from <http://ffmpeg.org/>
- Khattab D, Ebied HM, Hussein AS, Tolba MF. Color image segmentation based on different color space models using automatic GrabCut. *ScientificWorldJournal*. 2014;2014:126025
- Kruger, Volker. Image Processing. Lecture: Sistemi intelligenti naturali ed artificiali. Oct. 2010. Retrieved 2017. [PowerPoint]
- Devereux, V.G. Limiting of YUV Digital Video Signals. Research Department, Engineering Division. The British Broadcasting Corporation. Dec. 1987
- Podpora M., Korbas, G., Kawala-Janik A. (2014), “YUV vs RGB – Choosing a Color Space for Human-Machine Interaction”, *Federated Conference on Computer Science and Information Systems* 3:29-34. DOI: 10.15439/2014F206
- File:Colorcomp.jpg. (2018, June 6). *Wikimedia Commons, the free media repository*. Retrieved 19:07, September 6, 2018 from <https://commons.wikimedia.org/w/index.php?title=File:Colorcomp.jpg&oldid=30490238>

- S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt (2001). "Vision and Video: Models and Applications". In Christian J. van den Branden Lambrecht. Vision models and applications to image and video processing. Springer. p. 209. ISBN 978-0-7923-7422-0.)
- File:Yuv420.svg. (2018, June 8). Wikimedia Commons, the free media repository. Retrieved 19:12, September 10, 2018 from <https://commons.wikimedia.org/w/index.php?title=File:Yuv420.svg&oldid=305143901>.
- Kerr D., (19 Jan 2012) Chrominance Subsampling in Digital Images. Chrominance Subsampling Issue 3.
- Ho, Anthony T. S.; Li, Shujun (20 May 2016). Handbook of Digital Forensics of Multimedia Data and Devices, Enhanced E-Book. John Wiley & Sons.
- Microsoft Corporation (June 1998). "WAVE and AVI Codec Registries - RFC 2361". IETF.
- Cromey DW. Avoiding twisted pixels: ethical guidelines for the appropriate use and manipulation of scientific digital images. *Sci EngEthics*. 2010;16:639–67.
- Sachs, J. (1996) "Digital Image Basics" Digital Light & Color.
- Chapter 12: Image Data Compression. University of Iowa 55:148 Digital Image Processing lecture. <http://user.engineering.uiowa.edu/~dip/lecture/DataCompression.html>
- Zuiderveld, Karel (1994), "Contrast limited adaptive histogram equalization", *Graphics gems IV*, Academic Press Professional, Inc., pp. 474–485
- Zimmerman, J. et.al. An evaluation of the effectiveness of adaptive histogram equalization for contrast enhancement. *IEEE transactions on medical imaging*, 7(4), 1988. 6
- R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Third Edition, 2008.
- S. M. Pizer, E. P. Amburn, J. D. Austin, et al.: Adaptive Histogram Equalization and Its Variations. *Computer Vision, Graphics, and Image Processing* 39 (1987) 355-368.
- Stanley Sternberg's article, "Biomedical Image Processing", *IEEE Computer*, January 1983.
- Castle, M and Keller, J "Rolling Ball Code" NIH Image Pascal version. University of Michigan Mental Health Research Institute.
- Schmid, M. (2007) "Sliding Paraboloid" ImageJ contributor. Version 10-Jan-2008
- B. A. Sheno, Introduction to Digital Signal Processing and Filter Design, John Wiley & Sons, 2005 ISBN 0471656380.
- Huang, Thomas S.; Yang, George J.; Tang, Gregory Y. (February 1979). "A fast two-dimensional median filtering algorithm" *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 27 (1): 13–18.
- Arias-Castro, Ery; Donoho, David L. (June 2009). "Does median filtering truly preserve edges better than linear filtering?". *Annals of Statistics*. 37 (3): 1172–2009.
- Arce, Gonzalo R. (2005). *Nonlinear Signal Processing: A Statistical Approach*. New Jersey, USA: Wiley. ISBN 0-471-67624-1.

Shapiro, L. G. & Stockman, G. C: "Computer Vision". Prentice Hall, 2001

## BIOGRAPHICAL SKETCH

Michael Izaguirre is a first-generation college graduate from San Benito, Texas. He graduated from San Benito High School in 2013. He then completed his bachelor's in engineering physics with a concentration in Bioengineering from the University of Texas Rio Grande Valley in 2016. Here he began research work with Dr. Juan Guevara Jr. and Dr. Natalia Guevara in the Biophysics Research Laboratory. He completed his Master of Science in Interdisciplinary Studies in 2019 where he studied biophysics, physics and chemistry while pursuing imaging science research. He held positions as a research assistant, teaching assistant and assistant to the laboratory supervisor within the physics department during his Master's study. Michael Izaguirre's permanent address is 4740 Hackberry Rd, San Benito, Tx. 78586.