



# Mid-Latitude All-sky-imager Network for Geophysical Observations 2

---

## Atmospheric Imaging

March 5<sup>th</sup>, 2015

A Major Qualifying Project (MQP) submitted to  
the Faculty of Worcester Polytechnic Institute in partial fulfilment of the requirements for the  
Degree in Bachelor of Science in Electrical and Computer Engineering submitted by

---

Danielle Riccardi

---

Luis M. Vinke

Advised by:

Prof. John Orr

Prof. David Finkel

Supervised by:

Asti Bhatt

Elizabeth Kendall

Sponsored by:

## **SRI International**

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>

# Acknowledgements

We would like to thank the following organizations and individuals for all their valuable help and involvement throughout our project: Professors John Orr and David Finkel, our advisors on this project, for all of their hard work, help, and guidance throughout this project. Asti Bhatt and Elizabeth Kendall of SRI International, our sponsors, for all of their in-depth knowledge and guidance on the subject matter, broadening of our horizons in the world of scientific research, and desire to do this project. Steven Chen, John Kelly, Mary McCready, Scott Seaton, and all the other SRI staff in our area for giving us great advice and help whenever we needed it, and for making our time here at SRI International an outstanding experience. Lastly, we would like to thank SRI International for sponsoring this project, and the city of Mountain View, CA for hospitality allowing us to do this project.

# Abstract

Disturbances that propagate through the ionospheric region in Earth's atmosphere, called ionospheric waves, were the focus of this project. SRI International has a network of nine imagers scattered throughout the United States that take pictures of the night sky. This project developed two systems that process these images. The first system integrated images from the imaging network and displayed the resulting mosaic over a geographical map projection. The second system enhanced, analyzed, and identified ionospheric wave features in each image. The team also provided SRI with a developer's manual. These tools will be useful in providing a better tool for SRI to understand ionospheric waves.

# Executive Summary

The Earth has a protective layer of gasses called the atmosphere. The atmosphere can be separated into five distinct layers, the Troposphere, Stratosphere, Mesosphere, Thermosphere, and Exosphere. A region that lies between the Thermosphere and the Mesosphere, the Ionosphere, has been a popular region of study by scientists. This layer helps scientists gain insight into properties such as atmospheric composition and flow. In the ionosphere, disturbances exist and are commonly referred to as ionospheric waves. These waves have sources ranging from Aurora Borealis and Airglow to tropospheric thunderstorms and many other things, both known and unknown. Studying ionospheric waves and determining their source mechanisms of these waves sheds light on the energy coupling among various atmospheric regions, which is an outstanding issue in the field of space sciences.

SRI International (SRI) is a company that has been researching these ionospheric waves over the past years. SRI is a nonprofit organization headquartered in Menlo Park, California. Founded in 1946, SRI International's commitment can be found in their mission statement: "apply science and technology for the good of society". This innovation center provides research, services, technology and licenses, advice, and more to government services and industry [1]. This American nonprofit is dedicated to Science, Technology, Engineering, and Mathematics (STEM) through continuous investments in research and development.

To study the ionospheric waves, SRI has installed a network of sky-imagers, which are cameras that take pictures of the sky, in the United States. These network consists of a total of nine cameras as seen in Figure 1. Each sky-imager covers an area of about 687km in radius, making it sufficient to cover and study almost the entire US.

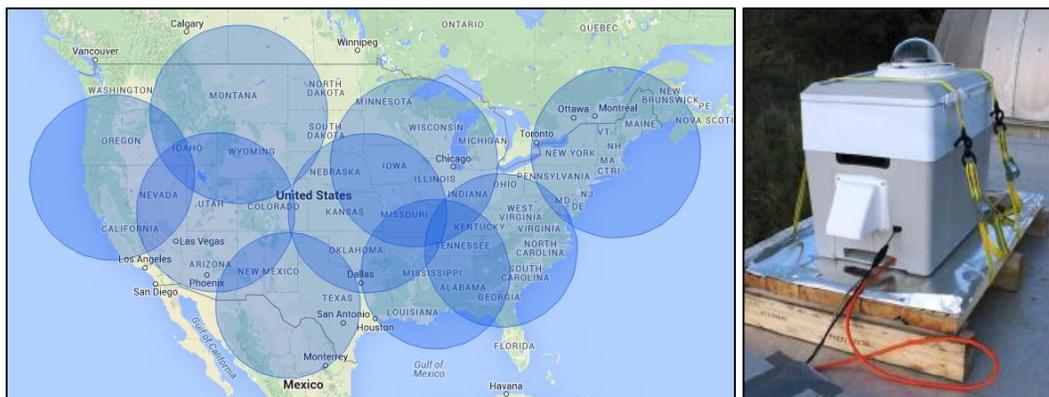


Figure 1. Sky Imagers (Right) installed across the United States (Left)

A previous WPI group developed a system that would perform digital image processing on the raw, binary images that the sky-imagers would capture. This system would unwarped the images, remove the stars, and uniformly space the picture in a square matrix. To better understand this process refer to the report: Mid-latitude All-sky Imager Network for Geophysical Observation [2]. An example of the original images that the cameras capture and the end result of this system developed by the previous WPI group can be seen below in Figure 2.

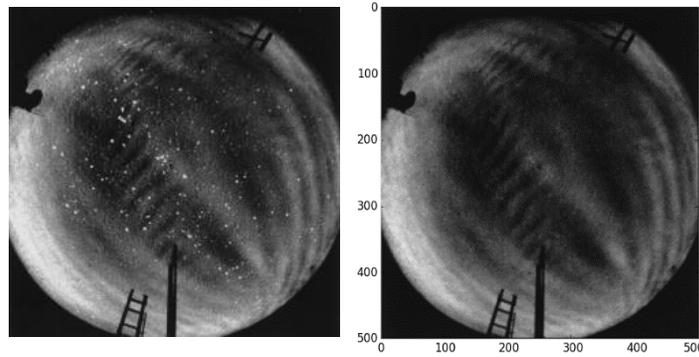


Figure 2. Digital Image Processing on raw images. Original image (left) and post-processed image (right)

After these images were processed, this project picked up where they left off. This project consisted of developing two systems that would further assist SRI research and understanding of ionospheric waves. The first system integrated images from the network of sky-imagers and created one large composite image of the sky displayed over the geographical projection of the United States. The second system enhanced and analyzed each image to identify features of these ionospheric waves.

### First System – Image Visualization

The first task was to create a large composite image. To do so, the regions that overlapped between sky-imagers had to be disregarded and the images had to be “stitched” together. The overlapping regions were identified in two different ways. The first approach was geometrically, using the radius of each image and calculating the intersection points. The second approach, knowing that every pixel can be related to a latitude and longitude, was to look for matching latitudes and longitudes between adjacent sites and disregard this overlapping region. In Figure 3 you can see the intersection points between each site, as noted by the red and blue dots.

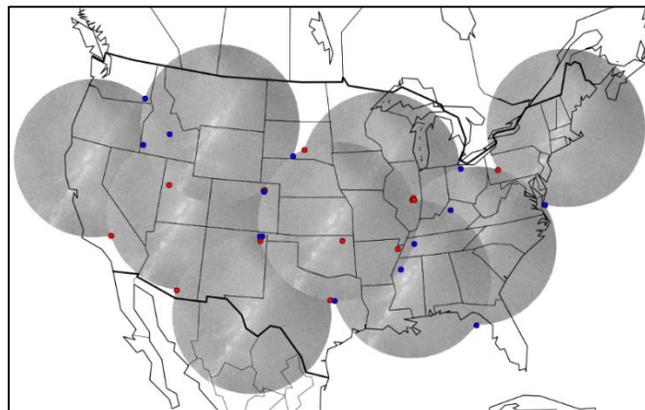


Figure 3. Intersection Points between sites

After identifying where each site overlapped, masks were created. These masks are used to hide, or cut, an image exactly in the overlapping area, yet still show the non-overlapping portion. Masks were created for every combination of adjacent locations. For example, if a site intersected with two other locations, then there were a total of eight different masks for that site; however, if a site intersected with only one other location, then there were a total of four different masks. Examples of what masks look like and what an image looks like after the masks were applied can be seen in Figure 4.

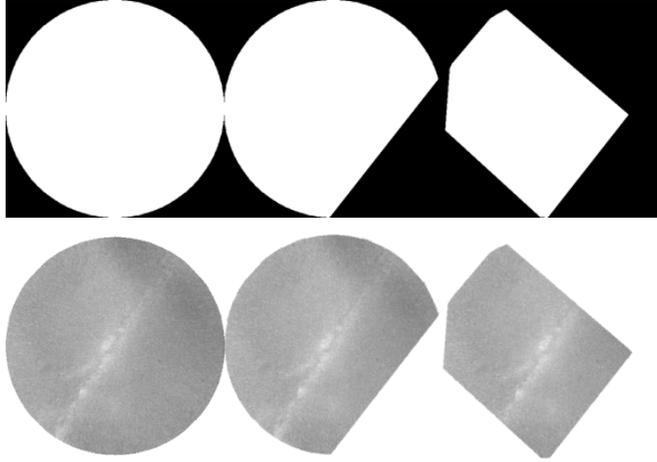


Figure 4. Example of masks (top), and images after masks were applied (bottom)

The purpose of masking images was to display a seamless, continuous map without the overlapping portions. This map and mask combination can also be used for real-time purposes to showcase the active sites, and mask images accordingly based on a site's status as they turn ON and OFF. Figure 5 shows an example of the entire US with masked images, and Figure 6 shows how the system can be used for real-time purposes. There are other applications that these system can be used for such as plotting thunderstorm data, cloud density, and more.

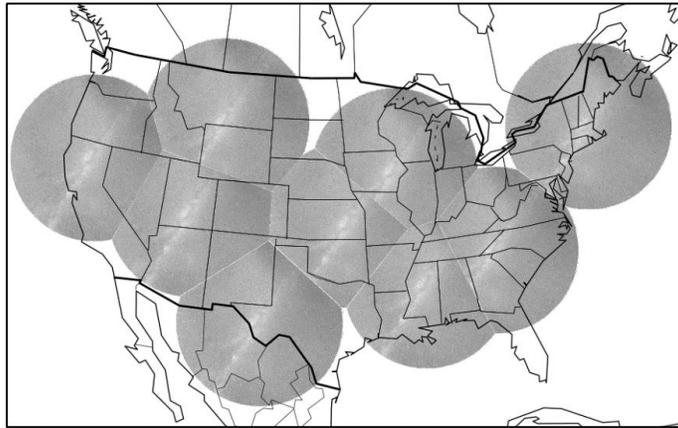


Figure 5. Example of map with masked images

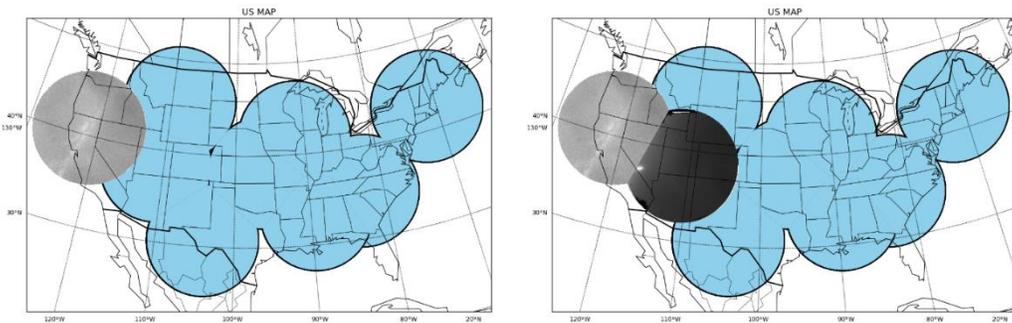


Figure 6. Example of map used for real-time purposes

## Second System – Image Analysis

The last goal of this project was to identify and analyze the ionospheric waves in each image. The first step was to enhance the images in order to identify the waves. The image enhancement process was performed using two different techniques. The first technique was image subtraction and the second technique was histogram equalization.

Image subtraction is an image enhancement technique that is used to determine the changes between two images [3]. These changes considered to be mostly uneven sections between images, which help identify if anything between the images moved, or identify different pixel intensities throughout the images. After performing the image subtraction and obtaining the difference, histogram equalization was performed. A histogram returns the pixel intensity distribution of an image. This latter technique is used to evenly distribute pixel intensities throughout the image; thus, improving the image contrast. This technique was performed over two sequential images at a time. A brief example of the enhancement process can be observed in Figure 7.

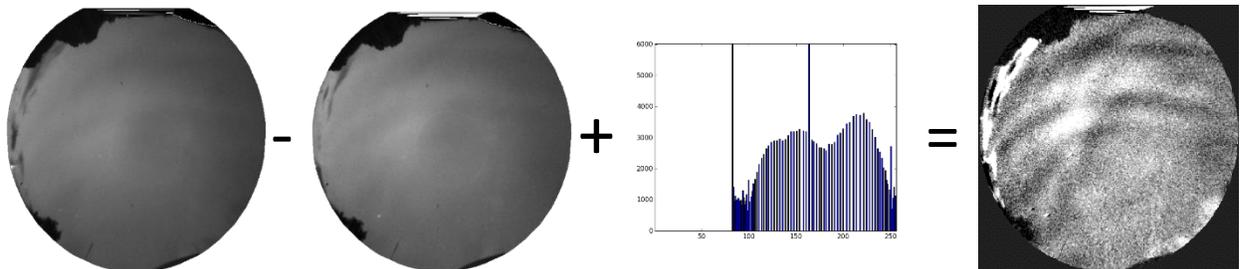


Figure 7. Image Enhancement Process

After obtaining these difference images, image analysis was performed to identify the direction of the wave and other features such as frequency, amplitude, wavelength, and speed. The first task was to automatically determine the direction of the wave. To do so, samples from each image were taken that are 500 pixels long and one pixel wide. These samples can be thought of as a vertical, horizontal, and diagonal lines that were drawn across the image and returned the pixel intensity from within each line. Refer to Figure 8 for an example of the samples taken.

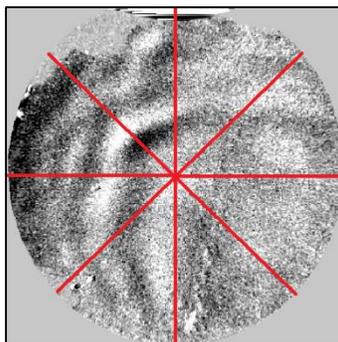


Figure 8. Taking image samples to identify the direction of the wave

These samples were taken for two consecutive images, and the data from both images was then compared. The samples were plotted to observe the obtained data, as seen in Figure 9. The blue noisy signal is the original pixel intensities for each sample. The red wave is the same data after being

processed through a Fast Fourier Transform (FFT) and low-pass filter. A FFT was done to reduce the noise and obtain a wave that was much easier to work with. The cleaned samples from the two images were then compared to identify the movement of the wave. Figure 9 shows the comparison of samples from the two image plotted on the same graph. The blue line represents image 1, while the green line represents image 2. The local maxima and minima of each wave were then compared against each other. The movement between the images represented the wave movement. Therefore, based on the sample direction, you could identify where the wave moved for each sample. After analyzing each sample across both images, magnitudes of change were calculated and summed to create vectors for each sample direction. Summing up those vectors resulted in the final magnitude and direction of the wave as observed in Figure 10 below, in which each color represents a different sample line.

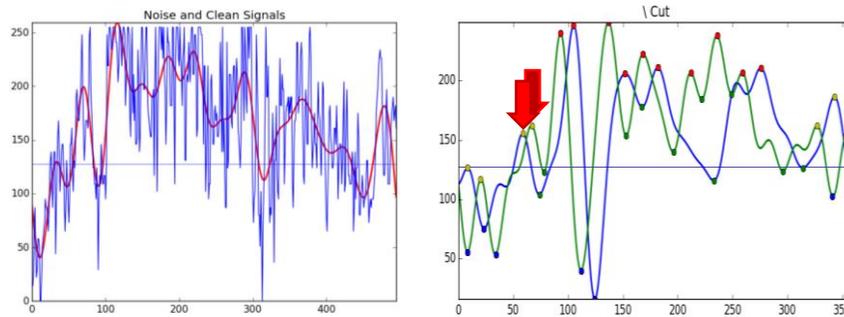


Figure 9. Direction analysis

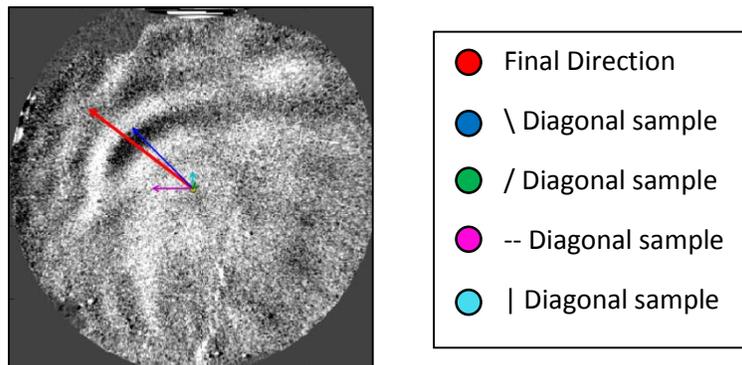


Figure 10. Wave movement through different samples

Once the wave direction was determined, each image was resampled along that vector sum line. This sample was then noise reduced, and the local maxima and minima were calculated. From there, wave features such as amplitude, frequency, velocity, period, and wavelength were calculated. These calculations were made through common measurement techniques and wave analysis equations.

These two systems mark the end of this project. The MANGO 2 team provided SRI with a large-scale categorized series of mosaic images of the wave activity and features that should be used for further research purposes. In addition, the team delivered a developer’s manual and all code pertaining to this project. The combination of these two systems developed during this project will hopefully assist SRI and future scientists in their research of ionospheric waves. Similarly, we expect the completion these systems to shed light on the energy coupling between various atmospheric regions, which is an outstanding issue in the field of space sciences, by determining the source mechanisms of these waves.

# Table of Contents

First System – Image Visualization.....	iv
Second System – Image Analysis .....	vi
1 INTRODUCTION.....	1
2 BACKGROUND.....	2
2.1 EARTH’S ATMOSPHERE .....	2
2.1.1 Main Layers of Earth’s Atmosphere.....	2
2.1.2 Ionosphere .....	3
2.1.3 Ionospheric Variations .....	5
2.1.4 Ionospheric Waves.....	6
2.1.5 Aurora and Airglow .....	6
2.2 SRI INTERNATIONAL’S ATMOSPHERIC WAVE RESEARCH .....	7
2.3 PRINCIPLES OF ATMOSPHERIC IMAGING.....	9
2.3.1 Projection Model.....	10
2.3.2 Source of Noise in Sky Imagers’ Images.....	11
2.3.3 Van Rhijn Effect and Atmospheric Extinction .....	11
2.3.4 Required Image Processing.....	13
2.3.5 Data Binning.....	14
2.3.6 Coordinate Mapping .....	14
2.3.7 Spatial Calibration .....	14
2.3.8 Star Removal .....	15
2.3.9 Projection onto a Uniformly Spaced Grid .....	15
3 METHODOLOGY AND IMPLEMENTATION.....	16
3.1 INTRODUCTION .....	16
3.2 VISUALIZATION OF IMAGES ON MAP .....	17
3.2.1 Creating a Map.....	18
3.2.2 Plotting Images on Map .....	20
3.2.3 “Stitching” Images Together .....	21
3.2.4 Decision-Making on Displaying Images.....	27
3.2.5 Displaying and Saving Images .....	27
3.3 IMAGE ENHANCEMENT AND ANALYSIS.....	28
3.3.1 Image Enhancement .....	28

3.3.2	Image Integration with Outside Data.....	28
3.3.3	Analysis of Image Wave Features .....	29
4	RESULTS.....	31
4.1	MAP MOSAIC IMAGES.....	31
4.1.1	Map Creation, Image Plotting, and Image “Stitching” .....	31
4.1.2	User Interface.....	34
4.1.3	Image Integration.....	35
4.2	ENHANCED AND ANALYSED IMAGES .....	35
4.2.1	Image Enhancement .....	35
4.2.2	Image Analysis.....	37
5	CONCLUSION AND RECOMMENDATIONS.....	40
5.1	Concluding Statements .....	40
5.2	RECOMMENDATIONS FOR FUTURE WORK.....	40
5.2.1	Map Mosaic Image System .....	40
5.2.2	Image Analysis Recommendations .....	40
	APPENDIX 1 – SRI: ABOUT THE SPONSOR.....	41
	APPENDIX 2 – VISUALIZATION OF IMAGES DISPLAY SYSTEM FLOWCHART .....	43

# List of Figures

Figure 1. Sky Imagers (Right) installed across the United States (Left) .....	iii
Figure 2. Digital Image Processing on raw images. Original image (left) and post-processed image (right) .....	iv
Figure 3. Intersection Points between sites.....	iv
Figure 4. Example of masks (top), and images after masks were applied (bottom) .....	v
Figure 5. Example of map with masked images.....	v
Figure 6. Example of map used for real-time purposes.....	v
Figure 7. Image Enhancement Process .....	vi
Figure 8. Taking image samples to identify the direction of the wave.....	vi
Figure 9. Direction analysis .....	vii
Figure 10. Wave movement through different samples.....	vii
Figure 11. Main layers of Earth's atmosphere with the Ionosphere and its electron density graphed next to it [11]. .....	3
Figure 12. Layers of the Ionosphere during the day and night [13]. .....	3
Figure 13. How the ionospheric layers refract different frequencies [20]. .....	4
Figure 14. Exaggerated look at how the height of the ionosphere changes between night and day [13]... 5	
Figure 15. Many colors of an aurora [19] (left). Airglow expressing three different colors [16] (right).....	7
Figure 16. Sky-imagers across the continental US.....	8
Figure 17. Sky Imager at SRI International.....	9
Figure 18. A Zuiko Circular Fisheye 8mm lens [21]. .....	10
Figure 19 – Central Perspective Projection Model (left) vs. Fisheye Lens Projection Model (right) [22]... 10	
Figure 20. Difference between a regular lens (left) and a fisheye lens (right) [23]. .....	11
Figure 21. Fisheye Lens Image Capturing Display .....	11
Figure 22. Van Rhijn Effect and Atmospheric Extinction on the Earth .....	12
Figure 23. Original Picture before image processing [20]. .....	13
Figure 24. Different Coordinate Conversions [27]. .....	14
Figure 25. Spatial Calibration, before (left) and after (right).....	14
Figure 26. Star Detection and Removal Algorithm, before (left) and after (right) [20].....	15
Figure 27. Image on a Uniformly Spaced Grid [20]. .....	15
Figure 28. Example of unwarped image on left and unwarped image on right. ....	17
Figure 29. Examples of map projections. GALL projection (left), and GEOS projection (right) .....	19
Figure 30. Continental map of US in LCC projection, created in python by using the matplotlib.basemap toolkit.....	19
Figure 31. Example of square image, red dots are extent locations, and blue (false color) show where the picture is transparent.....	20
Figure 32. Obtaining intersection points of two overlapping circles [31] .....	21
Figure 33. Example of 2D rotation [32].....	22
Figure 34. Iterating through boundaries of circles example.....	23
Figure 35. Example of image's pixel coordinates. Note how the origin is at the top left corner of the image.....	24
Figure 36. Origin of Map and Origin of Image .....	24

Figure 37. Resulting example of masking an image.....	25
Figure 38. Examples of different possible combination of masks. ....	26
Figure 39. Representation of how masking an image was done. ....	26
Figure 40. Example of pixel samples taken on each image. One line top to bottom, one left to right, one diagonally, and one counter diagonally.....	29
Figure 41. Clean Waves from Two Images, extrema highlighted as red, yellow, blue and green dots. Red arrows indicate two of many extrema to be compared. ....	30
Figure 42. Images plotted on LCC projection.....	31
Figure 43. Map of the US and atmospheric images, with intersection points between images shown in red and blue. ....	32
Figure 44. Example of different masks for the site located at Bridger, MT. It intersects with three locations.....	32
Figure 45. Example of masked images.....	33
Figure 46. Example of resulting mosaic with masked images .....	33
Figure 47. Example of real-time map as sites turn on and off. Blue circles are sites that are off. ....	34
Figure 48. Different options for map display module. All features (left); No map, transparent image (middle); no background, transparent image (right). ....	34
Figure 49. Plotted difference images over NASA AIRS research data.....	35
Figure 50. Example of Image Enhancement. Two consecutive images are on the left which are being subtracted .....	36
Figure 51. Example of histogram equalization, original histogram (left) and equalized histogram (right) .....	36
Figure 52. Resulting Difference Image .....	36
Figure 53. Original, noisy signal (blue) before FFT, and clean signal (red) after for each sample. ....	37
Figure 54. Calculated extrema (dots) for two images (blue and green lines) for each sample. ....	38
Figure 55. Vectors for each sample direction, bold red vector is automatically calculated wave direction. Image on right is zoom of vectors as seen in image on left.....	38
Figure 56. Resample for one of two images (left), results of image analysis and calculation for two consecutive images (right). ....	39
Figure 57. Automatic wave direction detection software not perfectly finding the wave direction. The wave is flowing in a much more NW direction than the bold red vector sum line would indicate (left). Calculated results, however, are not too far from expected values. ....	39
Figure 58. SRI's Five Disciplines of Innovation in association with their business model.....	42

# List of Tables

Table 1. SRI International's Sky Imagers' Locations.....	7
--	---

### 1 INTRODUCTION

The universe is a vast and mysterious place. Astronomers, astrophysicists, and many other scientists have done much to study it, but there still remains an immense realm of uncharted territory. The same can be said of regions of Earth's atmosphere. Scientists have been able to study much of it, but many areas of our atmosphere remain somewhat unknown. Proper knowledge of space around planets, and especially our Earth, can help scientists understand the universe. The ionospheric region of Earth's atmosphere is especially difficult to study, in part due to its high altitude. However, scientists are trying to gain insight into this region by studying atmospheric disturbances that propagate through the ionosphere, commonly referred to as ionospheric waves.

The upper atmosphere is a challenging region to make measurements, so these ionospheric waves are studied by scientists to gain insights into properties such as atmospheric composition and flow. Waves that perturb the ionosphere come in a variety of scales and propagate in multiple directions. These waves have sources ranging from Aurora Borealis and Airglow to tropospheric thunderstorms. Determining source mechanisms of these waves sheds light on the energy coupling among various atmospheric regions, which is an outstanding issue in the field of space sciences.

Atmospheric waves have been a topic of study for SRI International for several years. SRI International is a nonprofit organization headquartered in Menlo Park, California. Founded in 1946, SRI International's commitment can be found in their mission statement: "Apply science and technology for the good of society". This innovation center provides research, services, technology and licenses, advice and more to government services and industry [1]. To study these atmospheric waves, SRI International has built a sensitive all-sky imaging system capable of observing upper atmospheric waves at night for research purposes. This system consists of nine sky-imagers installed across the continental US.

A previous WPI group, "Mid-latitude All-sky-imager Network for Geophysical Observation" (MANGO), worked with SRI to design a system capable of taking all-sky images of upper atmospheric airglows and low-latitude auroras. MANGO developed a system, using the Python programming language, which takes raw images and outputs calibrated, unwarped images with stars removed. These output images were created by applying algorithms that correct distortions and eliminate background noise in the images. However, this previous project did not account for the creation of a large-scale mosaic image and the properties of the corresponding wave activity.

The goal of this project was to assist SRI International in studying the ionosphere by creating an image visualization program for images taken by their all-sky imagers scattered throughout the US, and a program that will automatically detect and calculate features of the ionospheric waves. The proposed project built on the previous project and had two main components. The first component was to develop image "stitching" code for the all-sky images and overlay the resulting mosaic on a geographic map projection. The second component was to develop an algorithm that automatically enhanced images and identified wave features in those images. The completion of this project led to a large-scale categorized series of mosaic image of the wave activity and features that should be used for further research purposes. In addition, the team delivered a developer's manual and all code pertaining to this project. The expected result of the project is to shed light on the energy coupling between various atmospheric regions, which is an outstanding issue in the field of space sciences, by determining the source mechanisms of these waves.

## 2 BACKGROUND

In the following sections, we present the information necessary to understand the context of our project. This information is separated into two main areas of thought. The first is about the Earth's atmosphere and ionosphere. The second main area talks about SRI's efforts to understand these ionospheric waves and the imaging hardware and software used to take and process photos of the atmosphere.

### 2.1 EARTH'S ATMOSPHERE

Surrounding the land and water of the Earth is a protective layer of gasses called the atmosphere. The atmosphere can be separated into five distinct layers, the Troposphere, Stratosphere, Mesosphere, Thermosphere, and Exosphere. These layers help keep the temperatures on the surface of the Earth fairly regulated and protect the Earth from harmful space radiation, so as to maintain proper living conditions on Earth's surface [4]. The atmosphere is most dense nearest to the surface of the Earth and thins out as altitude increases [5]. The Ionospheric layer itself is not considered to be a part of the five distinct layers, and will be discussed later in section 2.1.2.

#### 2.1.1 Main Layers of Earth's Atmosphere

The Troposphere is the layer that is most close to the surface of the Earth. It is the densest layer of the atmosphere, and contains almost all of the atmosphere's water and dust [5]. The troposphere is heated from the bottom by the sun's rays reflecting off the Earth and into the sky. Therefore, the temperature of the troposphere decreases as altitude increases. It is also the layer in which contains the most weather events [6].

The Stratosphere is the second layer from the Earth's surface. It is where most of the atmosphere's ozone can be found. Due to this abundance of ozone, harmful radiation from the sun is absorbed and the atmosphere is heated [5]. Air within the stratosphere contains very little water vapor, temperature increases with altitude, and the air is very thin, resulting in very little weather activity, clouds, or turbulence. This makes the stratosphere ideal for jet aircrafts and weather balloons to reach their maximum operational altitudes [7].

The Mesosphere is the middle layer of Earth's atmosphere. Unlike the stratosphere, the temperature of the mesosphere decreases as altitude increases, with the top of the mesosphere being the coldest part of the Earth's overall atmosphere [8]. Meteors from space burn up at this layer. This layer is harder to study because it is difficult for scientific instruments to reach this altitude. The air here is also so thin that gas molecules and atoms rarely collide. Noctilucent clouds and special types of lightning events called sprites, blue jets, and ELVES appear in the mesosphere [8].

The Thermosphere is the layer that is second closest to space. Temperatures at this layer fluctuate depending on the time of day, but typically range from 500°C to 2,000°C [9]. The air density in this layer is so low that it is also considered part of outer space, with space shuttles and the International Space Station orbiting Earth residing within this layer [5]. Auroras also occur at this layer due to charged particles colliding with atoms and molecules in the thermosphere and emitting photons of light [5]. Most of the x-ray and UV radiation from the Sun is also absorbed in this layer [9].

## BACKGROUND

The atmospheric layer that is farthest from the Earth is the Exosphere. This is considered the region where atoms and molecules escape into space. This is also where the air is most thin [10]. It is composed of incredibly widely dispersed particles of hydrogen and helium [5].

### 2.1.2 Ionosphere

The ionosphere is the region of the earth's atmosphere in which this project is studying. It is made up of charged particles that have been ionized by solar and cosmic radiation [11]. The high energy of the radiation excites the atoms in that region so much that they are stripped of at least one electron, resulting in a sea, or a plasma, of positively charged particles and loose electrons. The negative free electrons and positive ions attract each other, but are too energetic to stay together as an electrically neutral molecule [2]. Those charged particles are called ions. As seen in Figure 11, the ionosphere lies within the thermosphere and exosphere, and can have varying electron densities depending on the altitude.

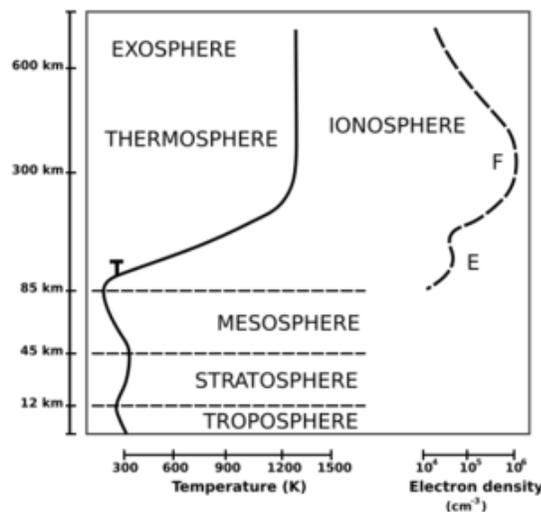


Figure 11. Main layers of Earth's atmosphere with the Ionosphere and its electron density graphed next to it [11].

The ionosphere is broken into three main layers: D, E, F, with F subsequently broken into F<sub>1</sub> and F<sub>2</sub>. As seen in Figure 12, the number of layers fluctuate in relation to the time of day, with the F<sub>1</sub> and F<sub>2</sub> regions combining into the F region at night, and the D region disappearing altogether [12].

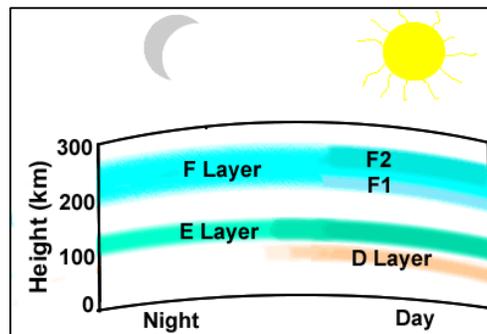


Figure 12. Layers of the Ionosphere during the day and night [13].

## BACKGROUND

Among other things, the ionosphere is used to propagate radio waves to distant places on Earth, and between Earth and satellites orbiting Earth in the thermosphere [11]. The electrons in the ionosphere are responsible for the refraction and reflection of high frequency (HF) radio waves [2]. Each layer of the ionosphere absorbs or reflects signals of different frequencies, with higher frequencies passing through higher layers, as seen in Figure 13.

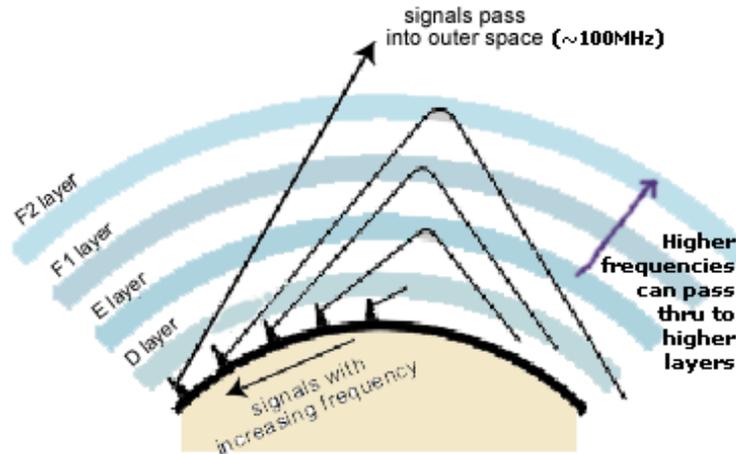


Figure 13. How the ionospheric layers refract different frequencies [20].

The D region is the lowest region in the ionosphere. Unlike the E and F regions, the D region almost completely disappears at night. The free electrons in the D region recombine with oxygen ions to form electrically neutral oxygen molecules at night, which renders the layer much more neutrally charged [12]. This allows radio transmissions to pass through the D region and into the strongly reflective E and F layers. The aforementioned radio waves, however, differ from the ionospheric waves in which our project is studying. Ionospheric waves will be discussed in section 2.1.4. The reflective capabilities of the D region in the daytime are still very weak, but can occur. However, the strength of the radio waves are reduced, thus causing the noticeable reduction in the range of daytime radio transmissions [12]. The E region is directly above the D region. It persists at night, but its ionization is considerably reduced. The F region is the layer farthest from the Earth. It has the greatest concentration of free electrons and its degree of ionization still persists from day to night with little change. In the daytime it has two distinguishable layers, F<sub>1</sub> and F<sub>2</sub>. F<sub>2</sub> is the larger and more highly ionized of the two, and is located above F<sub>1</sub>. The two layers merge at about the level of the F<sub>2</sub> layer at night [12].

During the daytime, the sun's rays are the main cause of ionization in the ionosphere. At nighttime, however, much of the ionization is due to either cosmic radiation from space, or there are leftover ions from the daytime. Between the transition from day to night, the time it takes for the Sun to ionize the ionosphere once its rays hit is nearly instantaneous [11].

The two principal processes that occur in the ionosphere are photoionization and diffusion. Photoionization is the ionization, or conversion, of electrically neutral atoms to electrically charged atoms, caused by light emissions, and makes up the electrical activity of the ionosphere. Diffusion occurs when these ions and electrons produced at high altitudes can freely disperse downward into the atmosphere, guided by Earth's electromagnetic field [2].

## BACKGROUND

There are many disturbances that can occur within the ionosphere as well, due largely in part to the high amount of electrons present and the influence of the Earth's magnetic field. Disturbances such as x-rays, polar cap absorption, geomagnetic effects, aurora, airglow, and lightning can all occur [2].

### 2.1.3 Ionospheric Variations

As previously mentioned, the ionosphere can be affected by many different factors. Variations in the ionosphere can have a large impact on satellite communication. Typical ionosphere variations include daily variations, solar cycle and flare variations, and latitude variations.

Daily variations, as mentioned in Section 2.1.2, occur and are due to the change in solar radiation. Figure 14 shows an exaggerated indication of how the height of the ionosphere from the Earth's surface changes between day and night. Higher frequencies are more likely to be affected during the day, while lower frequencies are more likely to be affected at night [2].

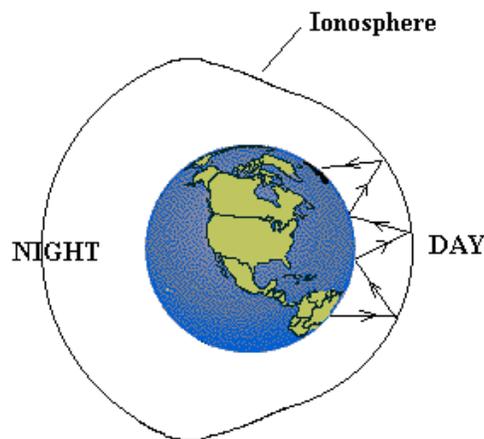


Figure 14. Exaggerated look at how the height of the ionosphere changes between night and day [13].

Variations from the solar cycle are caused by the periodic rise and fall activities of the Sun. This mainly affects high frequency devices. Things such as solar flares and the 11-year-annual time of greatest production in the Sun's solar radiation increases the amount of free electrons in the ionosphere. The higher amount of electrons in turn means that higher frequencies can successfully reflect and propagate. During the time of least solar radiation at the 11-year solar cycle, lower frequencies are better reflected to Earth [2]. During solar flares, x-ray energy and solar radiation hitting Earth's atmosphere increase the ionization of all layers, including the layer closest to the center of the earth, the D layer. Solar flares ionize the D layer so much that it is strong enough to reflect radio waves at lower altitudes. Once the solar flare is over, the electrons in the D region quickly recombine and signal strengths return to normal [11].

The last major cause of ionospheric variation is latitude variation. This is related to a site's specific latitude, and helps to determine the angle of contact between the solar radiation and the Earth. As daylight increases through the day, solar radiation hits the ionosphere in a more head-on position, and therefore solar radiation is much less impactful as the latitude increases [2].

## BACKGROUND

### 2.1.4 Ionospheric Waves

Ionospheric waves, also known as travelling ionospheric disturbances (TIDs), are disturbances in the electron density of the ionospheric region. These disturbances are known to propagate across large distances (thousands of kilometers) [14]. More specifically, these are ionospheric manifestations of gravity waves propagating in the neutrally charged atmosphere. Gravity waves, in turn, are waves generated in a fluid when the force of gravity and/or buoyancy try to restore equilibrium [15]. Things such as jet stream shear, wind flowing over mountains, and thunderstorms in the lower atmosphere can cause gravity waves that propagate up to the thermosphere and ionosphere [16]. This ripple effect from gravity waves on the ionosphere can be comparable to a ripple in a pond, where a disturbance such as wind flow or a strong thunderstorm creates a ripple in the ionosphere that propagates outward from the center of the disturbance. Ionospheric waves have a tendency to propagate in a northeastern to southwestern direction, however, smaller waves have been seen to propagate in other directions. [ASTI BHATT]. Ionospheric waves are one of the more common ionospheric phenomena that contribute to disruptions in ionospheric measurements of things such as the ionosphere's total electron content (TEC) and frequency measurements of the ionospheric waves. Solar eclipses, geomagnetic storms, and the day to night separation line known are some mechanisms responsible for observed TIDs. TIDs and their characteristics have mainly been studied in the Northern Hemispheric regions, mainly due to the data limitations in the in the Southern Hemispheric regions, however studies in that region have been done as well [17].

There are two different classifications for TIDs, medium scale and large scale. They are organized into these classifications based on characteristics such as period, velocity, source, and special distribution. Medium scale TIDs (MSTIDs) have periods of less than one hour and are believed to originate from many sources such as seismic events and shears in the jet streams. Large scale TIDs (LSTIDs) have periods that are over an hour and typically originate at high latitude regions. However, during intense geomagnetic storms, they can propagate to mid- or low-latitudes, or even the opposite hemisphere. The main source of LSTIDs are thought to be geomagnetic disturbances [17].

### 2.1.5 Aurora and Airglow

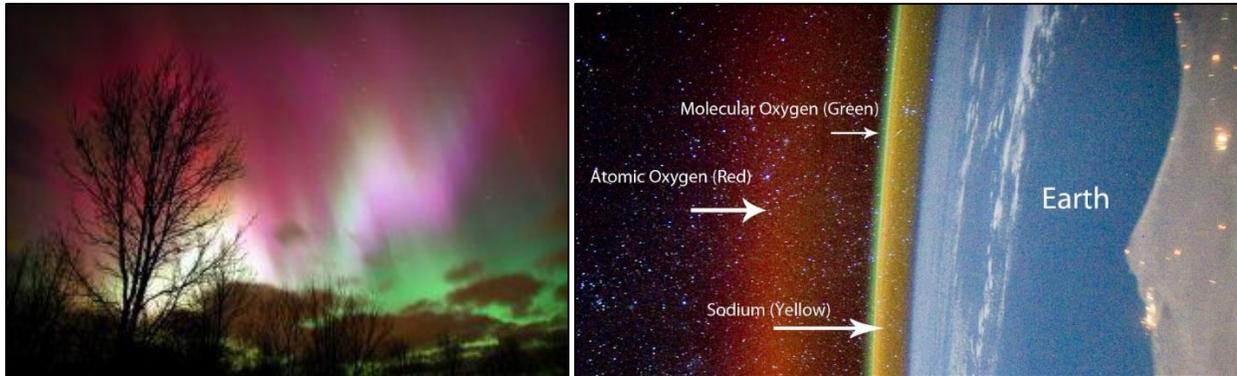
Auroras are luminous phenomenon that occur within the ionosphere and can be a manifestation of violent space weather. Plasma sent by the Sun causes disruptions in Earth's magnetic field, which then causes the magnetic field to reconfigure. Auroras are then formed when energetic electrons and protons of the solar wind interact with atoms of Earth's upper atmosphere. This interaction causes the formation of free electrons and ions in the upper ionosphere. Since much of the plasma in the solar winds are carried along magnetic fields, the auroras take the form of curtains, arcs, bands, and patches [2].

Auroras can have various colors such red, yellow, blue, violet, and green, due to the different wavelengths of light radiation being emitted by free ions, as seen in Figure 15. Altitude also plays a part in the colors of the auroras. Blues and violets are seen at low altitudes, green in middle altitudes, and red at higher altitudes [2].

Airglow is a thin glowing band that can be seen around the edge of the Earth, as seen in Figure 15. Airglow is the result of atoms, molecules, and ions that get excited by UV radiation from the Sun and then release the energy as both visible and infrared light when they return to their "normal" state [18]. This phenomenon occurs similarly to that of auroras. Airglows manifest themselves along the same

## BACKGROUND

color spectrum as auroras, with visible colors such as blues, greens, and yellows [18]. Airglow is brightest during daylight hours, however, the light of the sun makes it invisible during the day [16]. Scientists can use airglow and auroras to understand special distribution of charged particles, as well as to characterize their dynamics and can be used to help study ionospheric waves [18].



*Figure 15. Many colors of an aurora [19] (left). Airglow expressing three different colors [16] (right).*

### 2.2 SRI INTERNATIONAL’S ATMOSPHERIC WAVE RESEARCH

SRI International has been researching waves in auroras and airglows in the atmosphere for the past three years. A previous WPI group performed a project with SRI International and presented a design of an all-sky imager network system. This project was titled “Mid-latitude All-sky-imager Network for Geophysical Observation” (MANGO). As the project’s report explains the system MANGO designed was capable of taking all-sky images of upper atmospheric airglows and low-latitude auroras, and applying image processing algorithms to correct distortions and eliminate background noise. The image processing algorithms were coded in Python.

To achieve the most optimal hemispherical images, SRI International has set a network of nine identical sky imagers (explained in the next section) across the continental United States. These imagers will each take pictures at the same time and the images will then be analyzed and stitched together. Doing so, allows for optimal analysis of the images taken and best results. The sky imagers’ locations are stated in Table 1 and can be seen on a map in Figure 16.

The locations are as follows in Latitude, Longitude (-180 to 180):

*Table 1. SRI International’s Sky Imagers’ Locations*

Location	Latitude	Longitude	Location
1	45.298377,	-108.918643	Bridger, MT
2	41.888844,	-91.498607	Eastern Iowa Observatory, IA
3	42.611913,	-71.484704	Millstone Hill Observatory, MA
4	35.201148,	-82.874855	Pisgah Astronomical Observatory, NC
5	33.292046,	-89.386532	Rainwater Observatory, French Camp, MS
6	38.12713,	-96.138625	Madison, KS
7	30.6714,	-104.0225	McDonald Observatory, TX
8	38.185269,	-111.179774	Capitol Reef Field Station, UT
9	40.8,	-121.47	Hat Creek Observatory, CA

## BACKGROUND

Each circle here has the radius of 687 km. This is assuming 250 km altitude and 140 degree viewing angle looking straight up, as explained by Asti Bhatt, the project’s advisor.

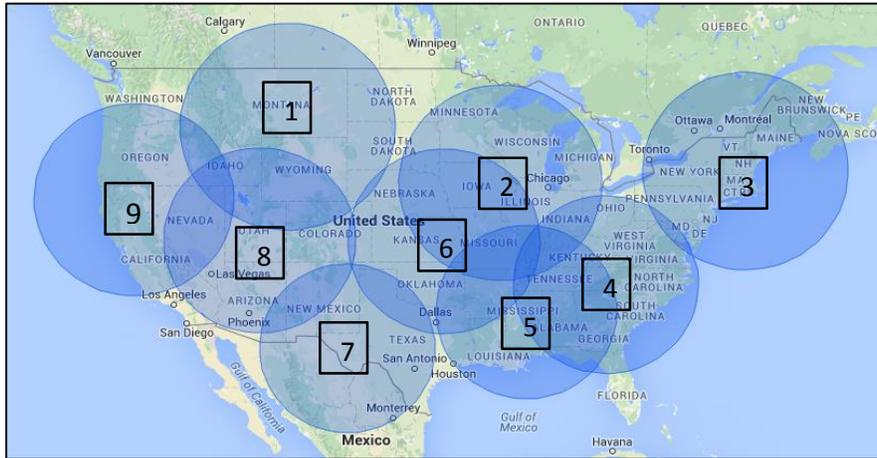


Figure 16. Sky-imagers across the continental US

The real-time observation system designed by MANGO and SRI consists of three subsystems: a data acquisition system, an image processing system, and a visualization system. The data acquisition system is responsible for image acquisition and remote system information acquisition. The image acquisition was achieved using CCD cameras with Windows drivers and software support for Windows-based platforms. The program created is able to monitor the camera enclosure’s ambient temperature, the power status of the computer controlling the camera, and its disk space. The next step is the image processing system. This subsystem rectified the distortions found in the images and mitigate the consequences of the Van Rhijn effect and atmospheric extinction. The image processing techniques performed to achieve optimal resolution of the images is spatial calibration, Van Rhijn and atmospheric extinction correction, and star removal. The last subsystem is the visualization system. During this stage the images processed are displayed. This step was achieved by designing a website to which students, scientists, and SRI International’s researchers have access.

Through this project, MANGO facilitated data acquisition on the consequences of geophysical phenomena on contemporary communication systems. This project led to the development of one of the first astronomical image processing programs using Python. Additionally, it advanced SRI International’s goal to learn more about mid-latitude airglow and auroras. Their study is currently used to further research macroscopic-scale auroras and airglows over the continental United States and characterize the dynamics of charged particles in the ionosphere [2].

This project will be a continuation of the MANGO project, and as such it will continue where the previous MQP group’s project was left off. This means that we will keep improving upon the existing platform and code already written. The star removal, coordinate mapping, spatial calibration, and uniformly spaced grid projection algorithms have already been programmed by MANGO and are explained in sections 2.4 and 2.5. Therefore, this team will continue by writing the image “stitching” algorithm, and writing code that will enhance and identify the ionospheric wave features in the images.

## BACKGROUND

### 2.3 PRINCIPLES OF ATMOSPHERIC IMAGING

As discussed previously, airglows, auroras, ionospheric waves, and other ionospheric variations have many adverse effects on satellite communication systems. To study these ionospheric waves, scientists use imaging technologies to gather data by taking and using atmospheric photographs. When images are captured, airglow emissions exhibit considerable spatial and temporal fluctuations over large geographic regions. Atmospheric imaging, therefore must be able to capture large areas in each image that are of high quality.

These atmospheric images are captured by what is referred to as a sky-imager. Each sky imager consists of a special camera and software that performs image processing techniques on the pictures taken. These techniques are discussed later in the chapter. Due to the application of the imagers, the cameras need to be light sensitive, low noise, and high resolution. Figure 17, below, shows an example of what a sky-imager with a fisheye lens looks like. This sky-imager is an example of the nine used by SRI International.



*Figure 17. Sky Imager at SRI International*

The sky imager used by SRI relies on CCD image sensors. CCDs are the preferred type of sensors for high-end imaging applications. Because of its superior image quality and flexibility, it is mostly used for digital photography, high performance industrial imaging, and scientific and medical applications. Regarding atmospheric imaging, high resolution, solid-state (CCD) imagers are used to provide detailed information of the occurrence frequency of the waves, their horizontal wavelengths, and their apparent phase velocities. As explained by Garcia et al, in their applied optics article published in 1997, “the CCD detectors in the sky-imagers are capable of viewing wave structure in the mesospheric airglow emissions over a large geographic area (up to 1,000,000 km<sup>2</sup> at 96-km altitude)” [20].

## BACKGROUND

### 2.3.1 Projection Model

Given that the sky imager takes hemispherical pictures, a very large field of view (FOV) and depth of field needs to be captured. An accessory that allows capturing images with such requirements is an ultra-wide angle lens. An instrument that is widely used for the measurement and study of the hemisphere is the fisheye lens. This unique lens, as shown in the figure below, allows the imaging of a large sector of the surrounding space by a single photo.



Figure 18. A Zuiko Circular Fisheye 8mm lens [21].

The geometry of images taken with a fisheye lens does not comply with the central perspective of projection as seen in regular camera pictures. The central perspective projection model assumes that “the angle of incidence of the ray from an object point is equal to the angle between the ray and the optical axis within the image space” [22]. Ellen Schwalbe explains in her article “Geometric Modelling and Calibration of Fisheye Lens Camera Systems” how the fisheye model deviates from the conventional system by introducing spherical coordinates [22]. While wide-angle rectilinear lenses capture angles of view of up to 100° degrees, fisheye lenses capture up to 180° degrees. Spherical coordinates used in fisheye lenses introduce the sense of depth into images. An example of the difference of projection between the central perspective and the fisheye lens model can be seen in Figure 19, below. The central perspective model on the left captures the area, above the red line, between the two outer black lines (about 100° degrees); the fisheye lens on the right captures everything ranging from the left outer black line to the right outer red line on the horizontal axis (180° degrees).

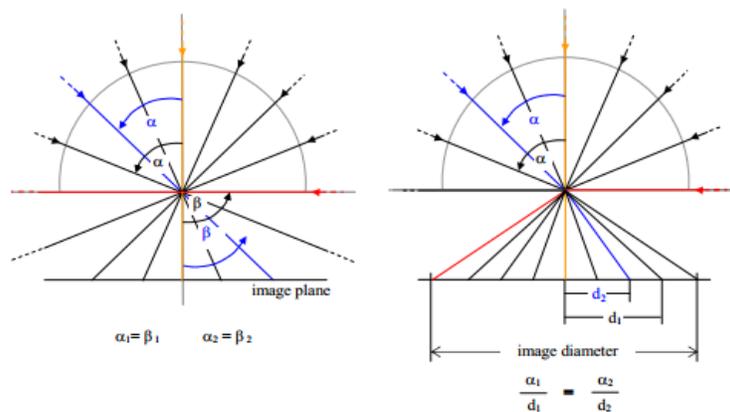


Figure 19 – Central Perspective Projection Model (left) vs. Fisheye Lens Projection Model (right) [22].

## BACKGROUND

As you can observe in Figure 20 below, the standard image differs from the fisheye lens image. This is an example of how fisheye lens pictures separate themselves from the standard rectangular image and projection model. It is important to note the difference of the fisheye lens projection model as the team will be analyzing pictures taken by this type of lens. Therefore, for future analysis of the images taken, spherical coordinates will need to be taken into consideration. This means that post-processing techniques of these images are required.



Figure 20. Difference between a regular lens (left) and a fisheye lens (right) [23].

### 2.3.2 Source of Noise in Sky Imagers' Images

Noise is unwanted variations in the pixel values that cause the image to deviate from its original scene. The noise presents itself in multiple ways such as “graininess” in darker backgrounds, faint horizontal or vertical lines, gradients between darker and lighter regions, or even low contrast in certain parts of the image. CCD images contain noise that can be caused by various sources. Some typical sources of noise in CCD images can be pixel graphics, electron conversion, and read-noise [24].

Some possible solutions to correcting noise can be taking a large sample of images, longer exposure time, or by combining multiple frames. However, the methods used by the previous team were standard image processing techniques, i.e. data binning, spatial calibration, star removal, etc.

### 2.3.3 Van Rhijn Effect and Atmospheric Extinction

Due to sky imagers using fisheye lenses, that have about  $180^\circ$  FOV, objects located closer to the optical axis, which is the center of the lens, are projected closer to the center of the image; and objects located further to the optical axis, are projected closer to the edges of the image frame. Figure 21 below depicts how images taken by fisheye lenses adhere further objects to the edges, while closer objects are closer to the center.

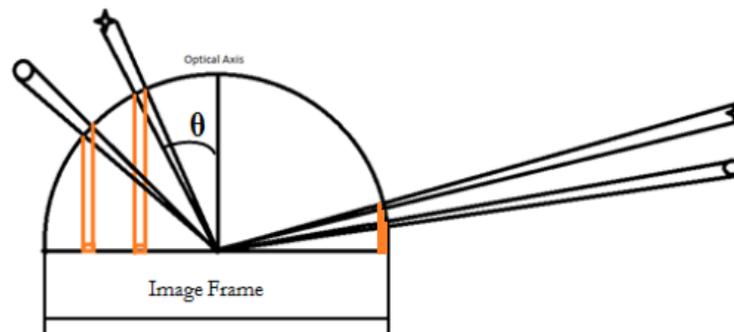


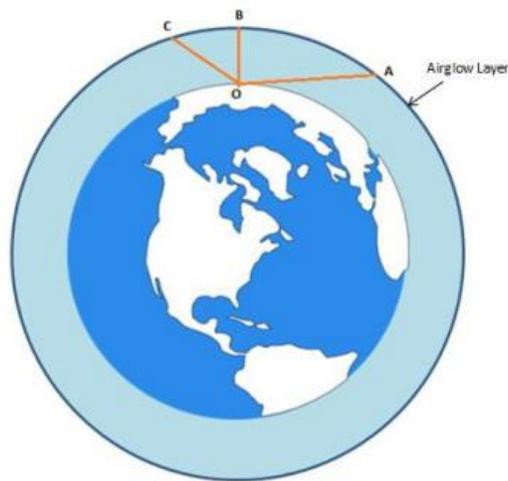
Figure 21. Fisheye Lens Image Capturing Display

## BACKGROUND

As mentioned, CCD images introduce noise into the sky imager's photographs. Other sources of noise are known as the Van Rhijn Effect and Atmospheric Extinction. These two are common optical noises in auroral and airglow images. These two effects are dependent on the distance of the object from the sky imager. An example of these effects is that objects closer to the optical axis can be shown as brighter than objects that are further from the optical axis.

Garcia et al. describe the Van Rhijn Effect as a line-of-sight enhancement of the airglow signal resulting from an increase in the optical path length when viewed at low elevations [25]. As shown in the result of this effect on the images is the concentration of objects at low elevations onto a few of the pixels. This concentration causes the pixels to be superimposed on outer pixels of all-sky images, hence brighter than the rest. In simple terms, objects further away from the center of the sky-imager will be represented over few pixels; in comparison of objects closer to the center and represented in a wider arrange of pixels [25].

On the contrary, the effects of the Atmospheric Extinction will cause pixels to be darker by reducing their true illumination. A further explanation of the uneven illumination distribution can be attributed to the light emission of each object. Light emissions will undergo attenuation due to increased air mass until it is captured by the sky imager. This means that objects at lower levels will travel more distance to the sky imager location. Therefore, the intensity received from further objects will not be as strong as that from closer objects. An example can be observed in Figure 22 below. Object B is located at the zenith, directly above the sky imager, therefore its light emissions will travel a less distance than objects A and C.



*Figure 22. Van Rhijn Effect and Atmospheric Extinction on the Earth*

Overall, the Van Rhijn Effect and Atmospheric Extinction oppose each other. As said, one causes pixels to be brighter while the other causes pixels to be darker. These are some of the problems with fisheye lenses and atmospheric imaging. The Van Rhijn effects explains the accumulation of color over a few pixels, while the Atmospheric Extinction explains the illumination strength received by the camera and reflected in the pixels.

## BACKGROUND

### 2.3.4 Required Image Processing

The process of image processing is the manipulation of an input to produce a desired output. This input is an image, such as a photograph or a video frame. After the image processing is applied, the output will be an image, characteristics of the image, or parameters related to the image. The previous MQP used digital processing techniques to process the images taken from the network of all-sky imagers.

Digital image processing techniques, as explained by MathWorks, “is the use of computer algorithms to create, process, communicate, and display digital images” [26]. The use of this type of image processing has various applications such as removing noise, adjusting contrast, detecting edges, compressing, changing colors, recognizing patterns, and rotating images. Digital image processing is successfully applied through the use of different mathematical expressions and algorithms. Some of these mathematical expressions can be primitive geometric equations, as for example equations of circles, lines, curves, squares, etc.; other equations can be quadratic equations to even more complex expressions like filters.

Due to the level of noise present in the pictures, other systematic errors, and the lack of geographical coordinate systems, several image processing techniques will need to be performed. These processing techniques required for image quality correction allow optimal analysis of the pictures [20]. For example, Figure 23 shows an original picture before image processing techniques are performed. Through these techniques the features, frequency, scale size, and direction of atmospheric waves can be determined. These image processing techniques required are explained below.

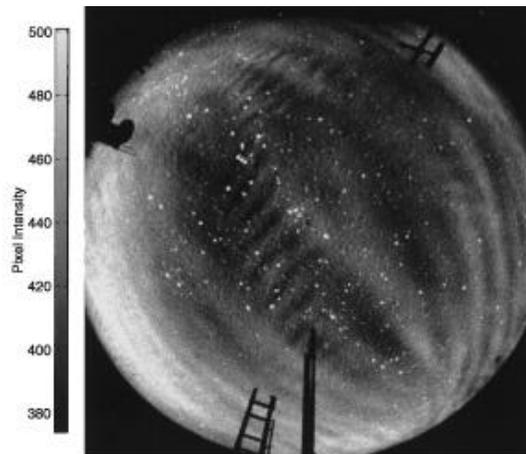


Figure 23. Original Picture before image processing [20].

## BACKGROUND

### 2.3.5 Data Binning

Data binning is a pre-processing technique that limits the read-out noise improving the signal-to-noise ratio. This improvement is achieved by reading a cluster of pixels as one super-pixel. Nonetheless, although this technique improves the signal-to-noise ratio, it worsens the resolution. For example, an original picture of 1024x1024 pixels, during the data binning process and using a 2x2 measurement, will read each 2x2 cluster of pixels as 1 super pixel. As a result the new image resolution will be 512x512 pixels, and each pixel will have an aggregate value of 2x2 pixels.

### 2.3.6 Coordinate Mapping

This technique relates distances between pixels in the image to physical distances in the airglow layer. Coordinate mapping is used to transform data from the original all-sky format to geographic coordinates, and vice versa. There are four types of coordinates that help organize the image. These are the original image, standard, azimuth-elevation, and geographic coordinates. The original image taken by the sky imager depicts an array of data. Each pixel represents a point in a data array. Every  $(i,j)$  point in the original image corresponds to a point  $(f,g)$  in the standard coordinates, which relates to  $(az,el)$  transformation that maps to a point  $(x,y)$  in the geographic coordinate. Figure 24, below, depicts the different coordinate conversions possible and its relation. It is possible to map between coordinate systems because they are all invertible, meaning  $(i,j) \Leftrightarrow (f,g) \Leftrightarrow (az,el) \Leftrightarrow (x,y)$  [27]

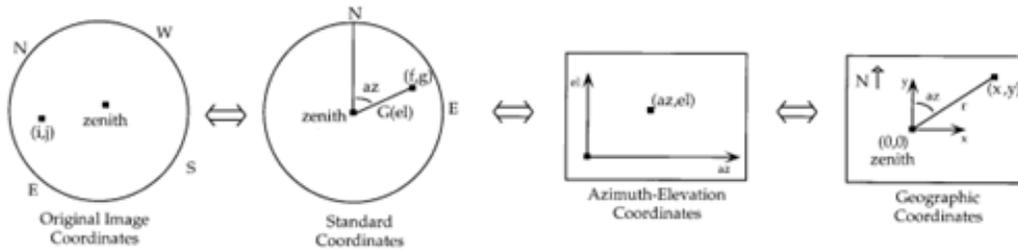


Figure 24. Different Coordinate Conversions [27].

### 2.3.7 Spatial Calibration

The axes on an image taken are not oriented along a geographic orientation, so a reference point is used for spatial calibration. Doing so allows the researcher to align the image with the respective coordinates (north, south, east, and west). Spatial calibration is performed by rotating the image, flipping it top-down or left-right, and scaling the x- and y- planes. For example, stars or even known points of the sky-imagers can be used as reference points in the sky for the spatial calibration step. Spatial calibration can be seen in Figure 25

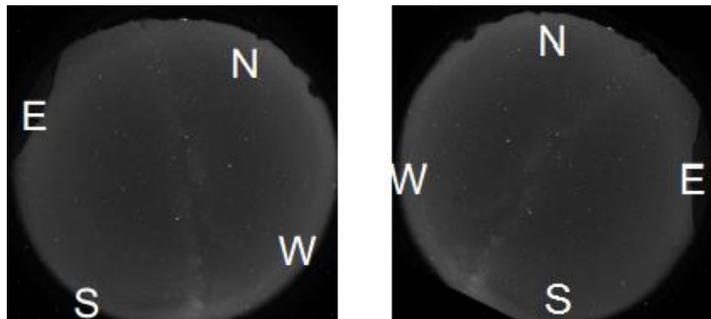


Figure 25. Spatial Calibration, before (left) and after (right)

## BACKGROUND

### 2.3.8 Star Removal

The purpose of the All-Sky Imager study is to analyze atmospheric waves. As a result, stars are not needed and can be removed for a better analysis of the waves in the pictures. Removing stars can be achieved using algorithms. For example, an algorithm can analyze each row and column of pixels looking for high intensity dots above a certain threshold. If an intensity difference is found in a pixel, it is then compared to the surrounding pixels to ensure it is not increasing brightness but an outlier from the mean. If it is identified as such, it is then replaced by a linear fit of the surrounding pixels. Thus, the star is removed. This process can be seen in Figure 26.

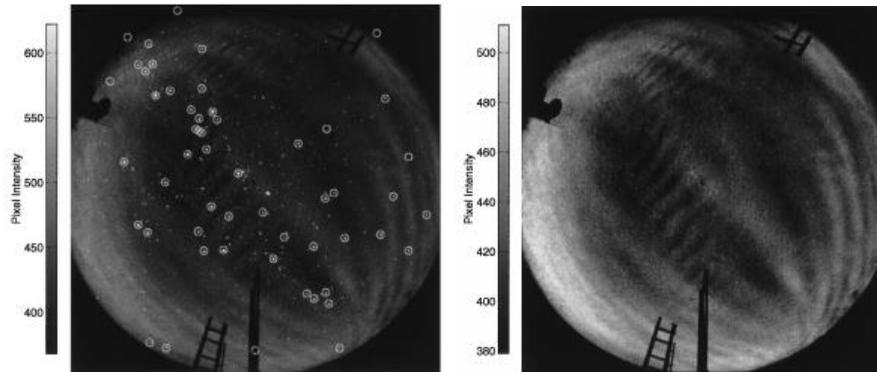


Figure 26. Star Detection and Removal Algorithm, before (left) and after (right) [20].

### 2.3.9 Projection onto a Uniformly Spaced Grid

After an image has been calibrated and the stars have been removed, a transformation is required to project the hemispherical view onto a uniformly spaced geographical grid. A uniformly spaced geographical grid is simply an image with the same amount of pixels in both the x and y planes. In simple terms, said grid will be an  $N \times N$  image, in the case of this study as it was defined by the previous MQP, the image will be spaced into a  $500 \times 500$  grid. To do so, unwarping is performed. This term refers to the movement of pixels in order to straighten out the image. An example of unwarping can be the selection of a specific area of an image, and mapping it with a certain pixel resolution onto a geographic grid for future analysis. The result of this projection can be seen in Figure 27.

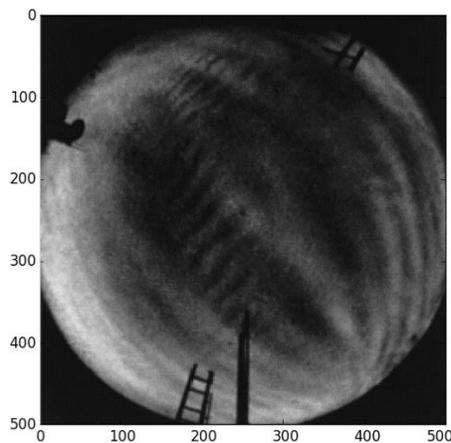


Figure 27. Image on a Uniformly Spaced Grid [20].

### 3 METHODOLOGY AND IMPLEMENTATION

This Major Qualifying Project (MQP) was a continuation of the previous MQP, “MANGO”. As previously mentioned, that team created code that processed binary images and produced unwarped images of the atmosphere, and taken from an all-sky imager. This MQP used the 500x500 uniformly spaced images that the previous MQP group created and further performed image processing. In the following sections, the work performed by this MQP group is explained. This section is separated into an overall introduction presenting the goals of the project, along with the respective objectives and tasks that were performed to successfully achieve the goals. Additional information regarding the decisions made during the design process and what had to be understood and performed is also explained.

#### 3.1 INTRODUCTION

This project had two goals.

1. Assisted SRI in developing a system that created and displayed a mosaic of images from a network of eight identical sky imagers installed across the continental United States in a geographical map projection.
2. Created a second system that identified the wave feature, frequency, scale size, and direction of ionospheric waves by analyzing images taken by SRI’s sky imagers.

To achieve the first goal the group completed the following objectives with its respective tasks:

1. Created a geographic map in a desired projection and scaled it to the continental United States using Python.
2. Displayed the images in the geographic map such that they are correctly plotted with respect to site’s latitude and longitude coordinates and scaled to the projection of the map.
3. Displayed a mosaic of images in the US map in a seamless, continuous manner, disregarding overlapping sections and accounted for each all-sky imager’s status (online/offline, resolution, etc.).
  - a. Developed an algorithm that identified the quality of each image and sorted the images to be displayed in order of quality.
  - b. Wrote an automated system that calculates intersection points of images from adjacent sites based on the map projection used.
  - c. Based on the intersection points calculated, created image masks that could self-generate based on images whose sites overlapped and the status of the adjacent sky imager’s location status (online/offline).
  - d. Utilized the masks created to “stitch” images together.

To achieve the second goal the group completed the following objectives through these tasks:

1. Enhanced images taken to highlight and identify wave regions
  - a. Researched different image enhancement methods
  - b. Created a system that enhanced images by taking the difference between two images, used histogram equalization, and enhancing the overall image.
2. Developed an algorithm that identified wave features.
  - a. Analyzed consecutive images to determine wave direction
  - b. Quantitatively measured waves

## METHODOLOGY

3. Developed an analysis routine that categorized the identified features by their frequency, direction and scale-size.
  - a. Wrote code that assigned categories to each image by comparing the results obtained with a set of parameters defined previously

The completion of these tasks and objectives led us to achieve three deliverables.

1. A series of images of a geographic map of the continental United States with the resulting time-sequence of “stitched-images” mosaics from a network of nine identical sky imagers.
2. A complete system implementing several algorithms that enhanced images and identified wave features, frequency, scale size, and direction by analyzing the images received from sky imagers.
3. A report that documented the design and implementation of the image processing system together with user’s and developer’s manuals.

For the completion of the objectives a Gantt chart was used. This schedule tracked our progress while at SRI International and helped us identify major tasks and milestones.

### 3.2 VISUALIZATION OF IMAGES ON MAP

As a continuation of the previous WPI group’s project, this project was able to begin with code that could already process raw images. These images, as explained in the background were originally received in a binary file format (i.e. .153 extension), and then several correction programs were performed to account for star removals, flat fielding, and more. Figure 28, below, showcases an example of an unwarped image (left) and a processed image (right) that is used to be displayed on a map.

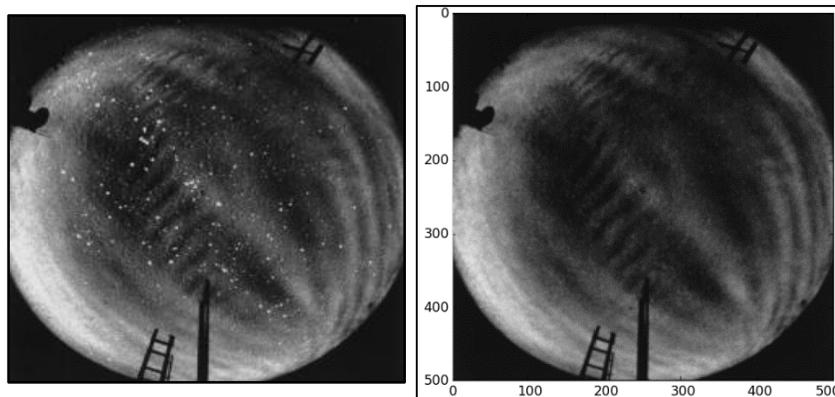


Figure 28. Example of unwarped image on left and warped image on right.

In order to keep track of the attributes of each site, we first created a comma separated value (.csv) file to store information for each location. Each location’s list contains the following information: the full name of the site, the abbreviation, and the site’s center in latitude and longitude.

The csv file information is used to create a list of lists in the main code. This list of lists first contains a list for each site, and each site contains the same information as the .csv file in addition to the time of the current image being processed, the file path of the image, and the last mask that was used. The reason for this list of lists is that it enables us to update it as the code runs while maintaining the original .csv file for future use.

## METHODOLOGY

The images are stored in separate folders defined by site and date. The folders labeling format is the abbreviation of the site and the month, day, and year the images were taken. For example, if the folder is “CFEB1915”, it translates to images taken on February 19<sup>th</sup> of 2015 by the sky imager located in Capitol Reef Field Station. Moreover, each folder has every image taken on that day. Each image is labeled using a predefined system. The system for naming the image files is as follows: “N112233A.png”, where N stands for the abbreviation of the location where the image was taken, and 112233 stands for the time in UTC of when the image was taken (“hhmmss”). For example, if a file has the name “H112233”, we know that H stands for Hat Creek Observatory, and the image was taken at 11 hours with 22 minutes and 33 seconds. This information is used for two things; to identify the location to which the image belonged; and the time in which it was taken. The time was necessary because it is used to check if a threshold time of five minutes had passed since the last image was taken.

Our program iterates through every folder and first converts the raw image files into .png files using the previous MQP’s system. Then, after receiving the processed images, our program proceeded to update each site’s location, check for the defined parameters given, and then display the images for each site on the US Map in accordance with the decision process programmed by the team. A flowchart of this system can be seen APPENDIX 2 – VISUALIZATION OF IMAGES DISPLAY SYSTEM FLOWCHART.

### 3.2.1 Creating a Map

In order to create a series of maps displaying the mosaic of images from each all-sky imager location, the first step in the process was to create the map. Map projections are an attempt to display a three-dimensional object onto a two-dimensional surface, such as displaying the spherical Earth onto a flat piece of paper or computer screen. Python’s Basemap Toolkit was used, as it was simple to implement and provided 30 different map projections.

We first had to decide which map projection to use. Map projections can either preserve the area of map features, or the shape of map features [28]. Figure 29 shows two projections, Gall Stereographic Projection on the left, and Geostationary Projection on the right. These projections preserve area, but not shape. The main map projection we chose was Lambert Conformal Conic Projection (LCC), seen in Figure 30. This projection was chosen because it preserved the shapes of figures plotted on it, while minimally distorting the area [29]. This project plotted circular images onto the map, so it was imperative that the shape was preserved. It was also a good projection to use when the zone to be displayed had a great east-west extent, as is the case with the map of the continental United States [30].

## METHODOLOGY

On the figure below you can observe different examples of map projections. The figure on the left, GALL projection, showcases area preservation, while the figure on the right, GEOS projection, showcases shape preservation.

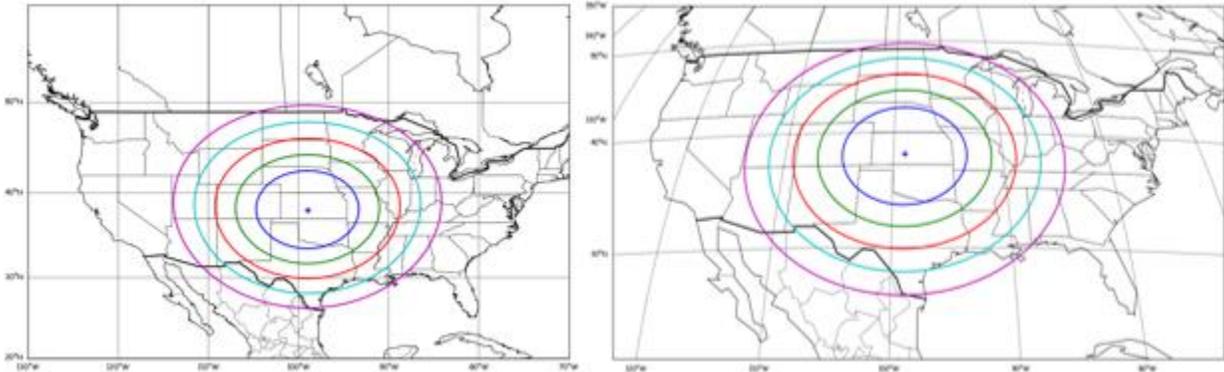


Figure 29. Examples of map projections. GALL projection (left), and GEOS projection (right)

As previously mentioned, this project used Python's matplotlib.basemap toolkit to create the underlying map. This toolkit not only created the map, it was also very useful for plotting 2D data onto the map. It offered the functionality to transform coordinates from x, y (m) to latitude and longitude. Through basemap one could plot images, contours, points, rivers, boundaries, and more. In Python, specific coordinates were given to center the map projection on the continental United States. Figure 30 portrays the resulting map created in Python using Lambert Conformal Conic (LCC) projection. This map was chosen as it preserves both, area and shape.

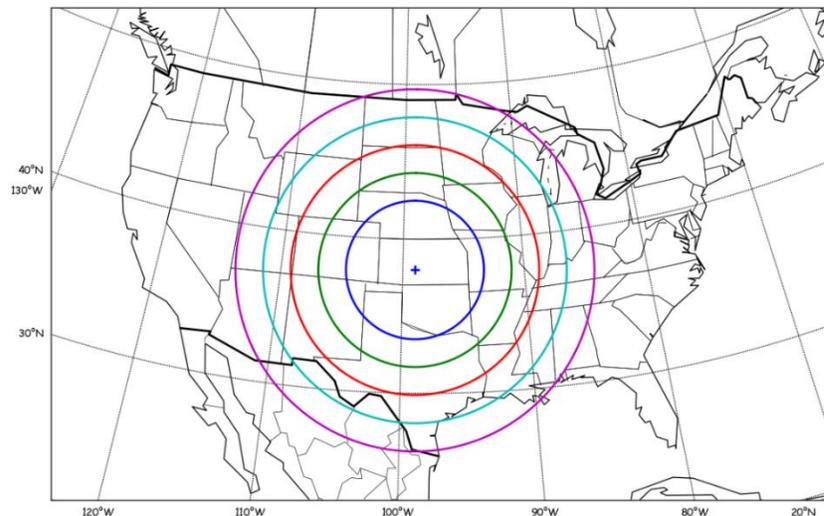


Figure 30. Continental map of US in LCC projection, created in python by using the matplotlib.basemap toolkit.

## METHODOLOGY

### 3.2.2 Plotting Images on Map

After the map had been created, the processed images from each site had to be displayed on the map in their respective locations. To do so, the team had to account for each sky imager's location and display the image according to its respective location and in accordance with the map projection. To do so, each image was scaled and projected by obtaining its extent locations. These extent locations for the images at each site were calculated by the previous MQP group through their program. Each sky imager has a defined viewing angle of  $140^\circ$  looking straight up from the center of the site, assuming an altitude of 250km. The location of each sky imager is also known in latitude and longitude. The program written by the previous MQP group could then calculate the radius of each sky imager, which related each pixel located to a specific distance from the zenith, which is the center of image, and calculate the latitude and longitude of each pixel based on its distance from the known latitude and longitude of the center. In the case of these sky imagers, the field of view was equal to 72 degrees, which relates to a radius of 687km for each image of the atmosphere. From the radius, the latitude and longitude coordinates for the northeast (top-right) and southwest (bottom left) corner of each image were identified, which relate to the extent locations. For more detailed explanation of how these extent locations were obtained, refer to the Mid-latitude All-sky-imager Network for Geophysical Observation report from Fabrice Fotso Kengne, Rohit Mundra, and Maria Alexandra Rangel [2].

These corners specified the scale of the image. While the displayed images themselves looked circular, the actual saved data was a square 500x500 image, with the background of the circular image saving as transparent. This can be seen in Figure 31. The Basemap toolkit was then able to extend, or print, the image from top right corner to bottom left corner based on where each corner was located on the map.

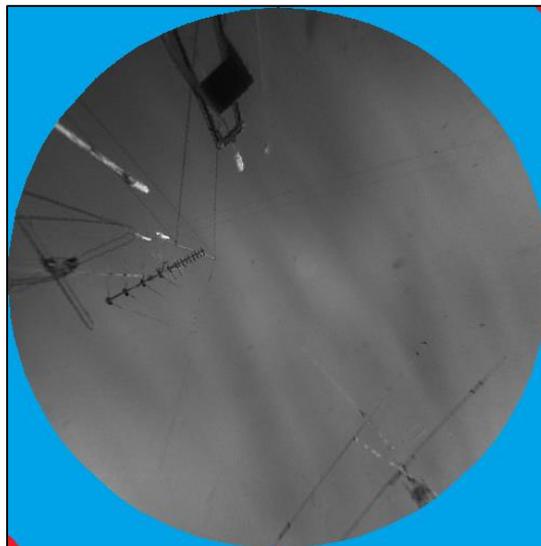


Figure 31. Example of square image, red dots are extent locations, and blue (false color) show where the picture is transparent.

As previously mentioned, the extent locations were received in latitude and longitude (degrees). In order to be placed on the map projection generated by Basemap, they had to be converted into meters relative to the map of the United States. Basemap had a built-in function which could do the conversion from latitude and longitude to the x and y coordinates in meters for the corresponding map projection.

## METHODOLOGY

### 3.2.3 “Stitching” Images Together

As seen in Figure 17, some of the areas covered by each adjacent sky imager overlap. To properly display the images on the map, these overlapping regions had to be accounted for to produce the best display of the image mosaic. We developed a method to account for the overlapping regions: slice the images where they overlapped and have part of each image displayed in the area that was covered by both images. In order to implement this method, we had to calculate the points of intersection between two site locations, convert pixel coordinates to map coordinates, and then overlap or slice the images. This upcoming section explains how each image was successfully “stitched” or cut so as to combine the images.

#### 3.2.3.1 Calculating Intersection Points of Adjacent Locations

First and foremost, the team had to calculate the intersection points of each sky-imager with another. Two different approaches were used to ensure the intersection points identified were correct.

#### APPROACH 1: GEOMETRICALLY

The first approach was performed mathematically through the means of algebra and geometry. Figure 32 presents a good example of how to approach a problem of two intersecting circles to find their respective intersection points. Below you can observe two circles with centers P1 and P2 located at  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively. The goal was to find the intersection points, IP1 and IP2.

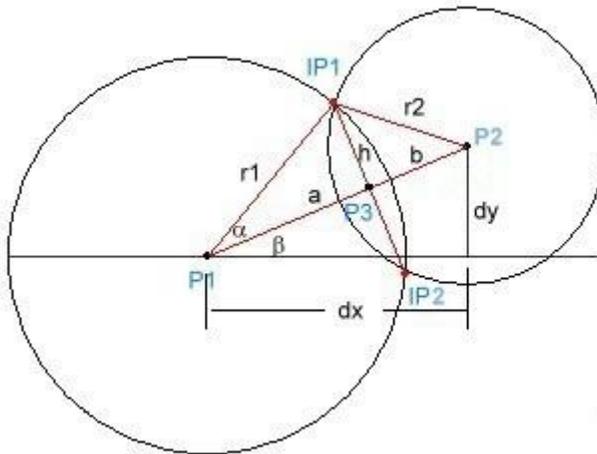


Figure 32. Obtaining intersection points of two overlapping circles [31]

The distance between each center was calculated by subtracting the x and y locations. This returned the offset distances from each point “P”.

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$

Next, through the Pythagorean Theorem we obtained the diagonal,  $a + b$ , or the total distance between point P1 and point P2. If dx is side 1 of the triangle, and dy is the height, the Pythagorean Theorem can be applied. This results in:

$$diagonal = \sqrt{dx^2 + dy^2}.$$

## METHODOLOGY

By applying the Law of Cosines, which relates the length of the sides of a triangle to the cosine of one of its angles, the length of “a” can be obtained. This length is the distance from center point P1 to the midpoint where the two circles intersect, assuming you draw a straight line through the intersections. The Law of Cosines states:

$$\cos(a) = \frac{r1^2 + diagonal^2 - r2^2}{2 * r1 * diagonal}$$

Solving for a by dividing both sides by r1 produces:

$$a = \frac{r1^2 - r2^2 + diagonal^2}{2 * diagonal}$$

Keep in mind that the radius of each sky-imager is the same, therefore the final equation yields:

$$a = \frac{diagonal^2}{2 * diagonal}$$

Now that we have  $a$  we want to rotate point P3 down so that it lies in the same axis as point P1. This allows point P3 to have a location  $(a, h)$  relative to point P1's  $(x1, y1)$ . This rotation will be equal to shifting point P3 by angle  $\beta$ . To rotate a point through a given angle, the following derived equations can be used. Figure 33 provides a visual representation of what 2D rotation would look like.

$$x' = x * \cos(\beta) - y * \sin(\beta) \text{ and } y' = y * \cos(\beta) + x * \sin(\beta)$$

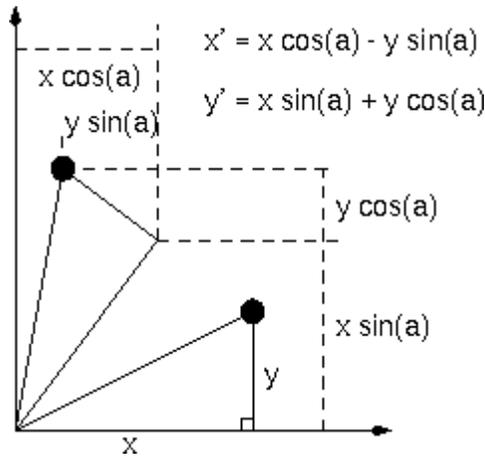


Figure 33. Example of 2D rotation [32]

The values of  $\cos(\beta)$  and  $\sin(\beta)$  can be calculated from the previously defined triangles formed by sides  $dx$ ,  $dy$ , and its diagonal. Hence,  $\cos(\beta) = dx/diagonal$  and  $\sin(\beta) = dy/diagonal$ . One last aspect to consider is because the rotation being performed is not from the origin  $(0,0)$  but actually from point P1  $(x1, y1)$ , the final coordinates need to be shifted back to their original state by adding back  $x1$  and  $y1$  to the rotated values. Therefore, the final intersection point's equations can be noted below.

$$IP1x = x1 + \frac{a * dx - h * dy}{diagonal} \text{ and } IP1y = y1 + \frac{h * dx + a * dy}{diagonal}$$

$$IP2x = x1 + \frac{a * dx + h * dy}{diagonal} \text{ and } IP2y = y1 + \frac{-h * dx + a * dy}{diagonal}$$

**APPROACH 2: FINDING MATCHING LATITUDES AND LONGITUDES**

The second approach looked for matching pixel coordinates (latitude and longitude) between two adjacent site images. Each location had corresponding files that contained the latitude and longitude locations of each pixel for that site’s images. Each file contained a circle the same size as the actual image (500 x 500 pixels), but instead of containing pixel values it contains the site’s coordinate information.

The comparison of a location’s longitudes and latitudes was done in pairs of two locations. Even if three overlaps happened, only two adjacent locations at a time were analyzed. Each file was read using python. The program proceeded to iterate through the circle’s boundaries. The reason to iterate through just the boundary was because the overlapping regions had several matching coordinates, but we only desired the intersection points at the boundaries. While each cell in the file was being read, the program searched for matches between the two files. After matching latitudes and longitudes were found, the program returned coordinates of the two exact points of intersection between the two sites. As shown in Figure 34, the dark blue colored area contained every pixel where each circle overlapped, but we only wished to obtain the two yellow intersection points IP1 and IP2. Therefore, by defining the starting point to be opposite sides and following the path as highlighted in the arrows along the red boundary, the code would eventually find and return the intersected points.

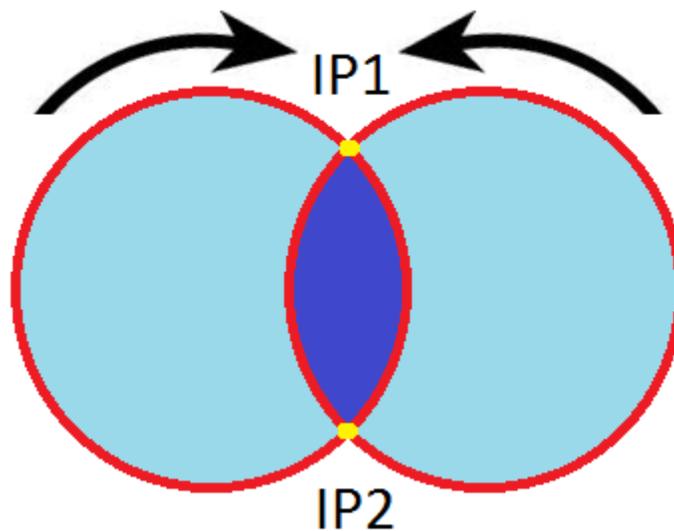


Figure 34. Iterating through boundaries of circles example

*3.2.3.2 Converting Map Coordinates To Pixel Coordinates*

After the intersection points were identified, these coordinates had to be converted from latitude and longitude coordinates from the whole map into pixel coordinates corresponding to each individual image. Pixel coordinates are the locations of each pixel on the map generated by python and given in meters. Each image, as seen in Figure 35, had dimensions of 500 by 500 pixels, as it was set by the image-processing program. Knowing the image’s boundary size allowed us to scale the map coordinates appropriately for each image. This was done so that we could translate the intersections of each image from map coordinates to individual locations on each image’s 500 by 500 grid.

## METHODOLOGY

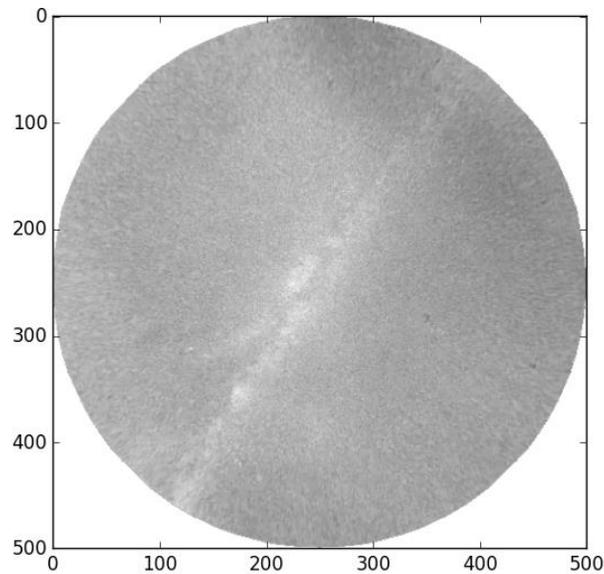


Figure 35. Example of image's pixel coordinates. Note how the origin is at the top left corner of the image.

Knowing the center location and the extent locations of each image, we can calculate the width and height of each image. In this manner, the total distance from the left boundary, or southwest extent location to the right boundary, or northeast extent location was obtained in the map's pixel coordinates. This calculation was done to account for the fact that no site lies at the origin of the plotted US map. As a result, the height and width of each image, was calculated using the equations below.

$$\begin{aligned}\text{deltaX} &= |\text{RightMostXvalue} - \text{LeftMostXvalue}| \\ \text{deltaY} &= |\text{TopMostYvalue} - \text{BottomMostYvalue}|\end{aligned}$$

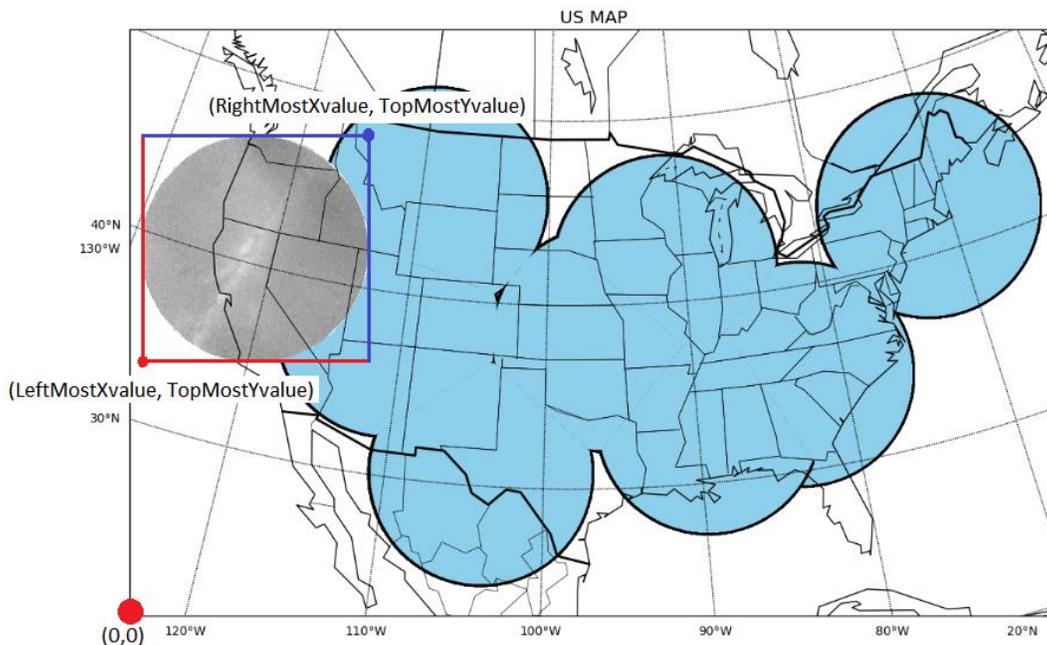


Figure 36. Origin of Map and Origin of Image

## METHODOLOGY

A ratio was then taken to keep the map coordinates proportional to the image's pixel coordinates. This ratio was then the intersection point in the image, IP x or y, over the total width of the image (500 pixels) equal to the intersection point calculated in the map over the total width of the image in meters. This resulted in what is shown below.

$$\frac{\text{Pixels IPx}}{500} = \frac{\text{Map's IPx}}{\text{deltaX}} \quad \text{and} \quad \frac{\text{Pixels IPy}}{500} = \frac{\text{Map's IPy}}{\text{deltaY}}$$

Lastly, given the map's origin, as shown in Figure 36, which is in the lower left corner, the Y values calculated had to be flipped. The reason is because, as seen in Figure 35, the origin is in the upper-left corner. Therefore, by subtracting the rationalized result by 500, we could obtain the flipped Y value. From the equations above, by solving for the Pixel's IPx and IPy and taking into account the difference in origin between the map and the image, we obtained the following:

$$IP1x = \frac{X1*500}{\text{deltaX}} \quad \text{and} \quad IP1y = 500 - \left(\frac{Y1*500}{\text{deltaY}}\right)$$

$$IP2x = \frac{X2*500}{\text{deltaX}} \quad \text{and} \quad IP2y = 500 - \left(\frac{Y2*500}{\text{deltaY}}\right)$$

### 3.2.3.3 Masking the Images

Image masks are just black and white images, which were then loaded and used to mask, or hide certain aspects of other images. In this application we used masks to hide certain sections of the images taken at each site. The sections of the image that were not hidden would correspond to the white areas of the mask, and the sections of the image that became transparent would correspond to the black areas of the mask. An example of image masking can be seen in Figure 37.

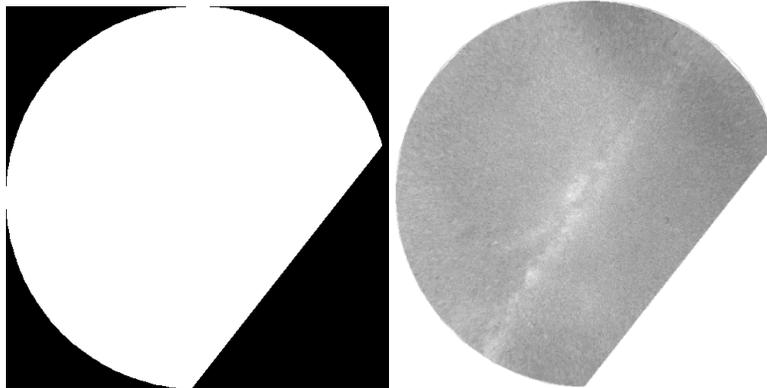
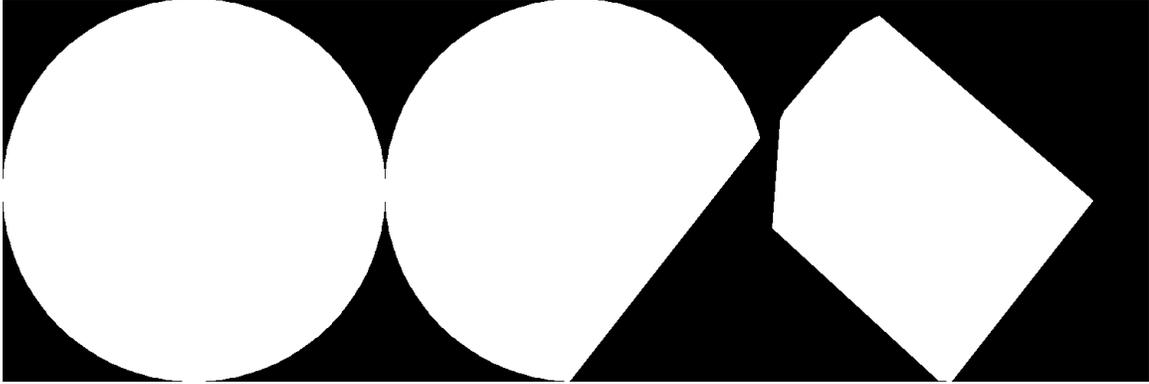


Figure 37. Resulting example of masking an image

In order to generate the masks for each site, we had to take into account the number of other sites that were adjacent to and intersected with the site in question, and the numbers of combinations that could be made in the event that some or all of the intersecting sites could be off (no longer sending images). The number of adjacent sites dictated the total number of masks and mask combinations to be generated. A mask would be generated if it fell under one of the three following criteria: no intersection occurred (left image), one adjacent site was active (middle image), or two or more sites were active (right image). Examples of these masks can be seen in Figure 38.

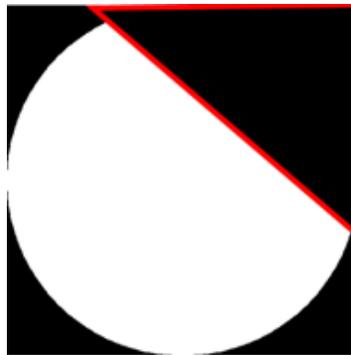
## METHODOLOGY



*Figure 38. Examples of different possible combination of masks.*

We first created a mask of the site as if it were not intersecting with any other site. We then generated masks for the site in question as it intersected with one other adjacent site, and did this for every site that it was adjacent to, and saved those masks. From there we were able to use the single-intersection masks to create more complex combinations where more than one site intersected with the site in question. All of these masks were then automatically named in accordance with the sites in the masks, and saved in folders corresponding to the name of the site in question. Ideally, masks would be generated once for each desired map projection, saved, and regenerated only if a new site was added or a different map projection was desired but had not yet been previously generated.

In order for a mask to be generated, the points of intersection for that site and another were needed. These points were then taken and used to calculate the vertices of the triangle to be drawn and used to create the mask that will cut the image (see Figure 39). In order to figure out which corner to place the triangle, the equation of the line created by the two points of intersection was needed. This was calculated by using basic line equation algebra. From there we determined where to place the triangle based on the line's slope and where the line fell in relation to the center of the image.



*Figure 39. Representation of how masking an image was done.*

## METHODOLOGY

### 3.2.4 Decision-Making on Displaying Images

After successfully creating masks, the next step in the overall system was determining how to display images, and which images to display based on the status of a site.

#### 3.2.4.1 *Determining the Status of Each Sky-Imager*

The system would take images from the folders that have the post-processed images. The program developed would then organize the list of images by time, and take the first and last image's time. This was used to determine when each location would turn on. The program would then get out of all the folders being processed, which location has the earliest time, and start with that file. This first location would be considered ON, while the others OFF. The system would then start processing each image in the folder until a location's image's time would match the start time of another location, then the system would process both images at the same time and consider both of those sites ON. Similarly if a folder reached during the process the last image in its folder, the site would afterwards be considered as OFF. The status of each site was used to determine the right mask and display in the mosaic map the images appropriately.

A crucial part of the program was to reflect the status of a site through the image mosaic. If the site was considered offline, the program would display a predefined error image rather than the last actual image used. Therefore, if a location was indeed off, the image directory path was never found, or the image was wrongly processed, the image displayed at that location would be an error image, which was typically a blue circle with a black border. A site would be considered off if no newer images were being received for that site, or if the folder reached the last image in it and that image was already processed.

#### 3.2.4.2 *Choosing The Right Masks according To Each Site's Status*

To display the images on the map, the program would slice each image via masks to create a more seamless image mosaic. This technique required knowledge of the status of each site and which sites intersected with others.

As previously mentioned, each site was considered off until an image was received from that location. Any sites that were considered on and actively displaying images had to account for all other sites with which it intersected. If the other site was on, then the two images would be sliced and share the space in which they overlapped. However, if the intersecting site location was off, the active site would not be sliced and would be able to display its full image. This step gets more complex with multiple intersecting sites as previously masked images had to adapt to the newly adjacent activated site, and thus display the right mask. As a site was turned off, the program would auto-adjust so as to display the correct masked-image and adapt to the newly turned off site.

### 3.2.5 Displaying and Saving Images

The last two steps of the process were to print each image onto the map projection and display the resulting mosaic, and lastly save the mosaic as a .png file.

For displaying the images, each image, after the corresponding mask had been applied, was then plotted on the map projection by locating each pixel to a specific latitude and longitude. This step was done for every image that was online at the moment, and those that were offline were disregarded. Hence, the resulting mosaic displays every image online with its respective mask at its respective location. Additionally, each mosaic would have a title that contains the month, date, and year the image

## METHODOLOGY

was captured, followed by the location and the current time that particular image was taken, for every image that is displayed in the mosaic.

For saving the mosaic images, the labeling used was the month, date, and year the image was captured, followed by the abbreviations of the sites that were on, organized alphabetically, and the number of the image. For example, if an image was taken on the first day of the month of February of 2015, and the mosaic displays only Hat Creek Observatory and Capitol Reef Field Station, then the file would be named FEB0115CH01.png, FEB0115CH02.png, FEB0115CH03.png, etc.

### 3.3 IMAGE ENHANCEMENT AND ANALYSIS

The image enhancement and analysis portion of our project consisted of three main parts: the enhancement of processed images, the integration of our enhanced images with outside data, and the analysis of wave features in the images.

#### 3.3.1 Image Enhancement

The first part of our image enhancement procedure was to enhance the images, two at a time, such that the waves in the pictures became more pronounced. This was done by first taking the difference between two consecutive images, and then performing histogram equalization.

Image differencing is an image enhancement technique that is used to determine the changes between two images [3]. We had to figure out which image was already more enhanced, and subtract the less enhanced image from the more enhanced image. This was done by comparing pixel intensities between the two images and choosing the image that had a higher intensity. This transformation combined two images into one image that we would then further enhance. We then performed histogram equalization on the resulting image so that we could enhance the contrast of the image and better see the overall waves. These resulting “difference images” were then saved to the folder corresponding to its site.

#### 3.3.2 Image Integration with Outside Data

Knowing that these images would also be used to compare with outside data, we had to make an interface that would integrate data from the NASA AIRS project and these images.

In order to view the difference image data over data from NASA, we first had to make our difference images semi-transparent. This was done by creating image masks that, when applied to an image, create a resulting image that is semi-transparent. The process of creating these masks was done by first loading the previously generated black and white masks, or generating them if they did not already exist. Then it modified the image such that all white areas became a shade of gray, corresponding to the level of transparency desired (more/darker gray = more transparent). These were then saved in another folder so that they could be used in other programs.

The NASA data we were working with was saved as in a netCDF3 format, which is a format that is commonly used to save weather, atmospheric, and oceanographic data. Our interface first opened that file and extracted the data. It then created a map of the continental United States using Basemap, and plotted the NASA data on top of it and added a colorbar on the side of the figure. This image was then saved. The interface then continued by loading the recently saved image of the NASA AIRS data as well as the desired all-sky imager difference image mosaic that had the same data and time as the NASA data. Currently the image and the NASA data are chosen manually to correspond in time. The mosaic

## METHODOLOGY

difference image was generated by our image enhancement code, plotted on a map, and then saved without the map in the background. This was done so that the enhanced mosaic image would be geographically placed in the correct location, but would otherwise have no background and be semi-transparent. The difference image mosaic was then plotted on top of the NASA data, and then saved.

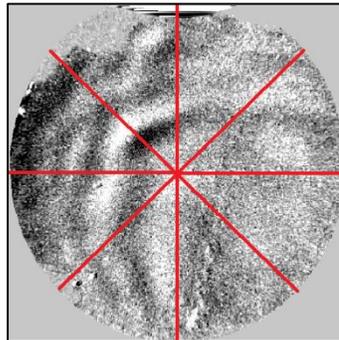
### 3.3.3 Analysis of Image Wave Features

The process of analyzing image wave features first began with automatically detecting the direction of the wave, and from there we could analyze the wave and determine properties such as speed, frequency, amplitude, wavelength, and period.

#### 3.3.3.1 Detect Wave Direction

The first major component of analyzing the waves in each difference image was to find its direction. Finding the direction of the waves manually would be too tedious and undesirable. Therefore, we developed a program that would detect the overall wave direction based on two images. This program would first take pixel samples of the two images, then compare the samples, find the magnitude of change for between the samples of the two images, and lastly determine the wave direction based on those magnitudes.

The two images had four samples each, as can be seen by the red lines in Figure 40. Each sample was a line 500 pixels long, and one pixel thick, resulting in four arrays per image. These arrays contained pixel intensities along each sample.



*Figure 40. Example of pixel samples taken on each image. One line top to bottom, one left to right, one diagonally, and one counter diagonally.*

From there we plotted the samples on graphs, however, there was a lot of noise due to the image enhancement. We used a fast Fourier transform (FFT) with a low pass filter to clean up the signal. This was performed on each sample, for both images, and they were then plotted together to show how the samples changed between the two images.

We then located the local maxima and minima for the cleaned waves by using a function from python's numpy toolkit. Next we compared the local maxima and minima of each corresponding sample between the two images and calculated the magnitude of change. You can see an example of the local maxima and minima that we found in Figure 41. We had to be aware that not all local extrema from one image sample would directly correspond to the extrema from the other image sample. We made thresholds that compared the extrema from image one to image two in terms of distance away from each other in both the x and y axis. If an extremum from image two was either too far away in the x-axis

## METHODOLOGY

from the extremum in image one, or had a drastically lower pixel intensity (y-axis) from the extremum in image one, it was discarded and the program moved on to check the next local extremum in image two. This comparison and totaling was done separately for both the local minima and maxima. These magnitudes were then totaled together to form the magnitudes of change for each sample. From there we summed up the vectors for each sample and were able to use the vector sum to find the overall wave direction for each image.

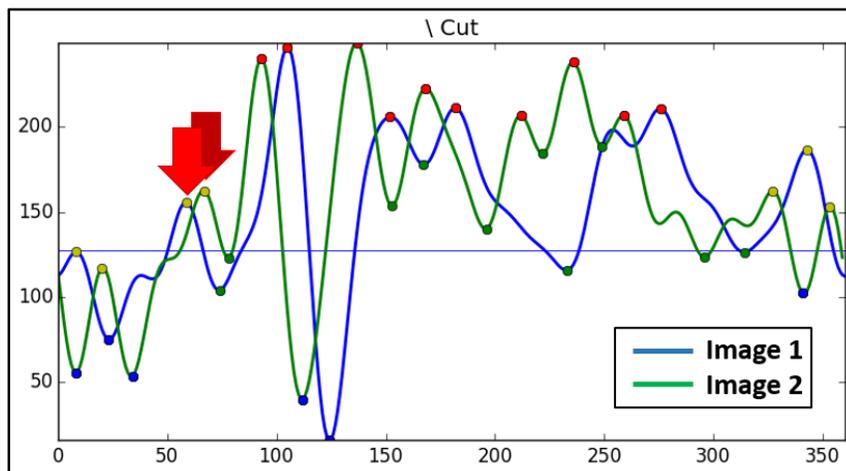


Figure 41. Clean Waves from Two Images, extrema highlighted as red, yellow, blue and green dots. Red arrows indicate two of many extrema to be compared.

Once the wave direction was calculated, the images were resampled along the path of the calculated vector. This would provide optimal wave data for further analysis. Examples of the total magnitude obtained and direction of the wave can be seen in the results section.

### 3.3.3.2 Determine Wave Properties

Wave properties for the images that were processed two at a time were determined from both the data gathered from the resampling along the calculated wave direction, and the timestamps of the two images. The properties that we calculated for each image were: timeframe, average wavelength, velocity, average amplitude, frequency, and period. Timeframe and velocity were the same for both images, and all other values were calculated on a per-image basis between the two images.

Timeframe was calculated by using the titles of each image, which are set up in the format HH:MM:SS UTC, parsing them into seconds, and then subtracting the earlier time from the later time. The average wavelength was calculated by finding the pixel distance between each local maxima, and then finding the average. The velocity was calculated by finding the length of the vector in pixels along which the new sample was taken, then using the following equation:

$$v = \text{vectorLen} \times \frac{\text{PIX\_TO\_KM}}{1000} \times \frac{1}{\text{timeFrame}} \quad \text{where } \text{PIX\_TO\_KM} = \frac{2 \times (687 \text{ km})}{500 \text{ pixels}}$$

The average amplitude was calculated by finding the height above the middle pixel intensity (127.5) each local maxima was, and then averaging those values together. The frequency and period were calculated by these two equations:

$$f = \frac{v}{\lambda}; \quad p = \frac{1}{f}$$

## 4 RESULTS

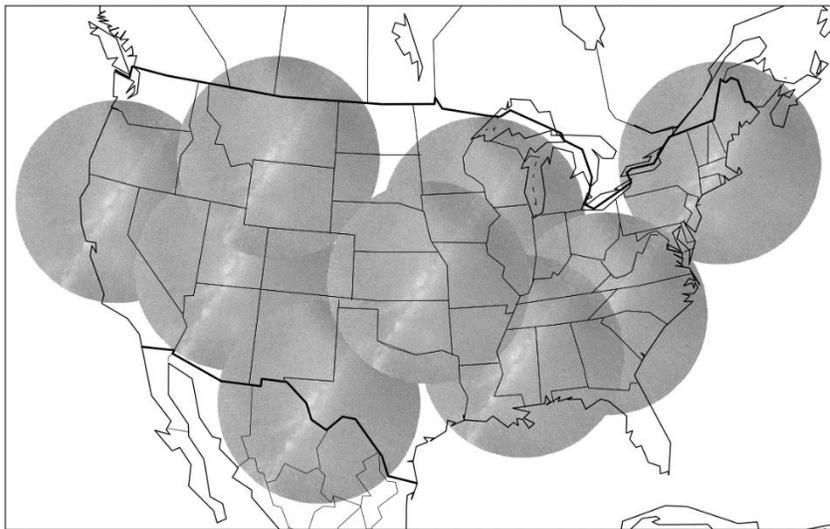
In this section we present our results on the topics of: presentation of the images from the all-sky imagers on a map, editing of those images to create a seamless image mosaic, the user interface associated with that module, the enhancement of the all-sky images, the integration of those images with our mosaic plotting code and data from other organizations, and analysis of the ionospheric wave features in those images.

### 4.1 MAP MOSAIC IMAGES

In this section we describe the results of our program that can plot images from the network of all-sky imagers in the correct location on a US map, how image masks were generated, how they were applied to the plotted images, and how the user interface with this module operates.

#### 4.1.1 Map Creation, Image Plotting, and Image “Stitching”

As mentioned in the methodology, the very first step in the process of displaying and saving mosaic images was determining the most adequate map projection. The projection chosen was the Lambert Conformal Conic projection, as it best preserved shape and mostly preserved area. On this map projection, an image was plotted at each all-sky imager location. The image used for each site on display in Figure 42 was a placeholder for test purposes.



*Figure 42. Images plotted on LCC projection*

You can observe in the figure above how each image intersects at certain points with adjacent sites. The program that was written automatically identified the intersection points and returned these values as it can be seen in Figure 43.

## RESULTS

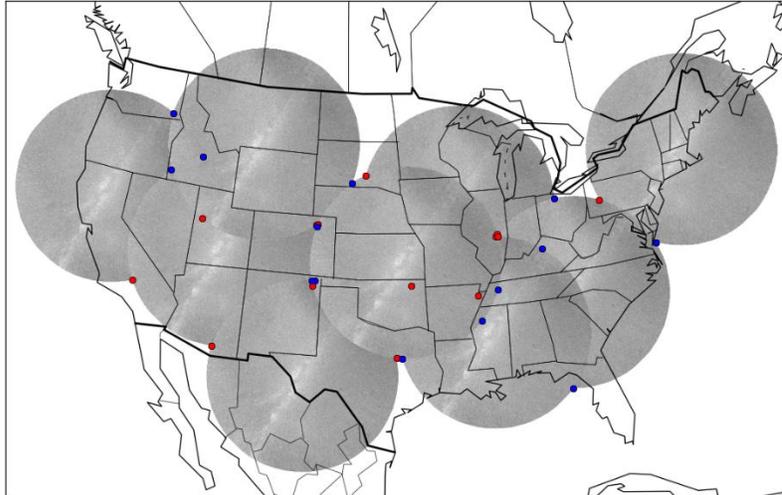


Figure 43. Map of the US and atmospheric images, with intersection points between images shown in red and blue.

The values obtained were in map coordinates, which the program then converted into pixel coordinates for each image. Using these image coordinates masks were created. Different combinations of masks were created depending on the number of locations with which a site intersected. In Figure 44 you can observe an example of each types of possible masks for a site that intersects with three other sites, in this case Bridger, MT (B) intersecting with Capitol Reef Field Station, UT (C), Madison, KS (D), and Hat Creek, CA (H). The first letter of each file represents the current site, and the abbreviation after the “to” are the sites with which it intersects.

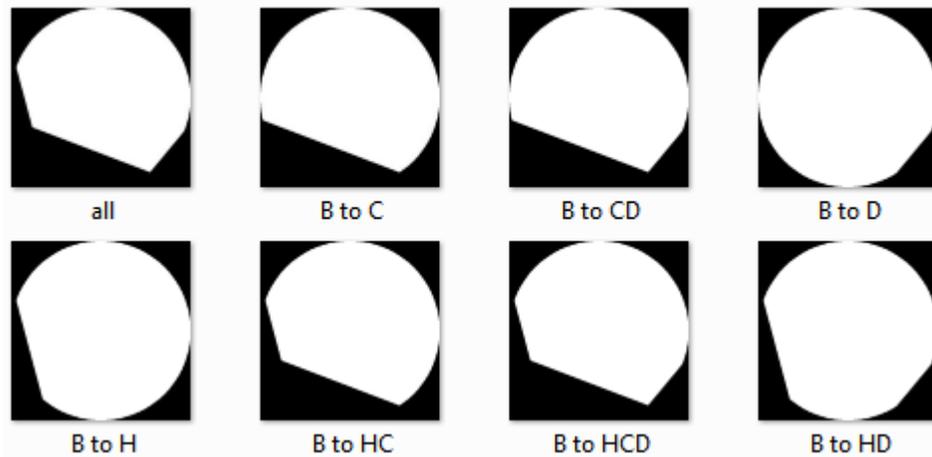


Figure 44. Example of different masks for the site located at Bridger, MT. It intersects with three locations.

These masks were then applied to the all-sky images to “cut” the images that were to be displayed in accordance with the adjacent sites that were on and it intersected. An example of the resulting masked image can be seen in Figure 45.

## RESULTS

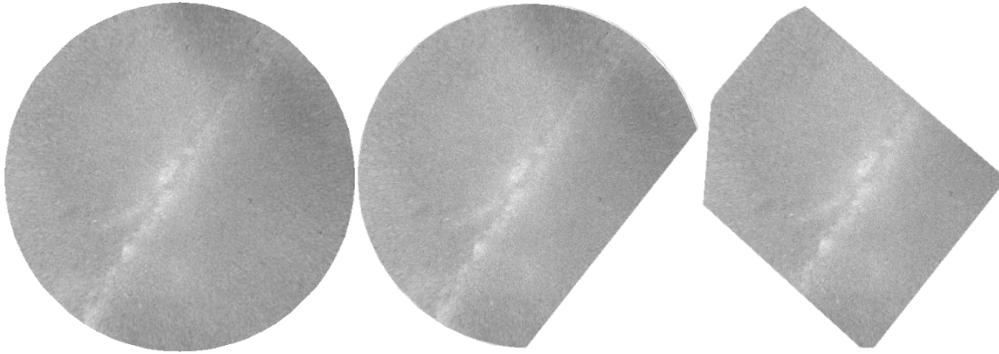


Figure 45. Example of masked images

Finally, after the system determines which adjacent locations are ON and OFF, and it masked the images accordingly, the system proceeded to save the resulting mosaic image in a specified folder. An example of the resulting mosaic can be seen in Figure 46 below.

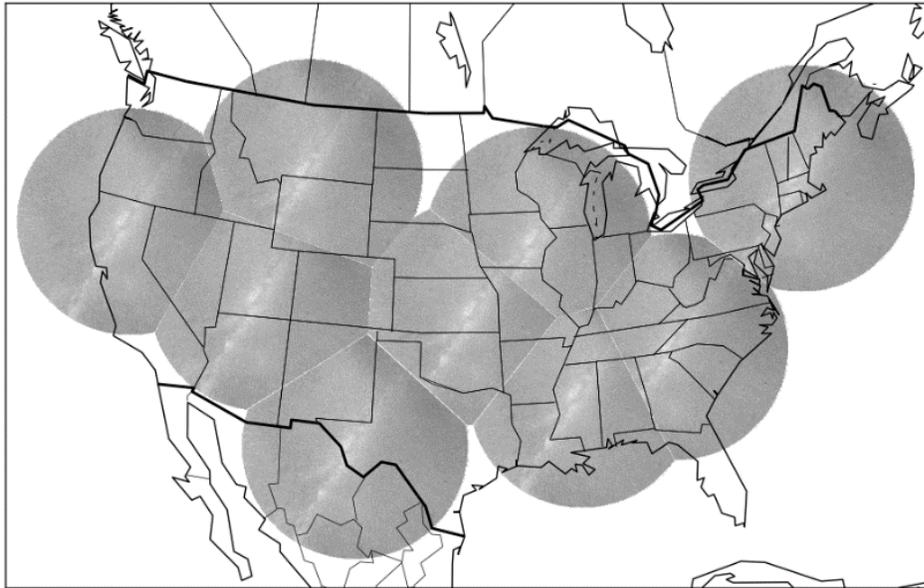


Figure 46. Example of resulting mosaic with masked images

Masking the images have multiple applications. One example can be the one provided below. If the system is to be running in real-time, the map would adapt as images would turn ON and OFF and display the appropriate images with the appropriate masks. The following sequence of images in Figure 47 simulate what a real-time system would show. The first map has only one site ON. As a second image comes in, the map would adapt and mask both images accordingly. Then, a third location turns on and the map adapts and displays the image for that location. The pattern repeats itself as locations turn ON and OFF.

## RESULTS

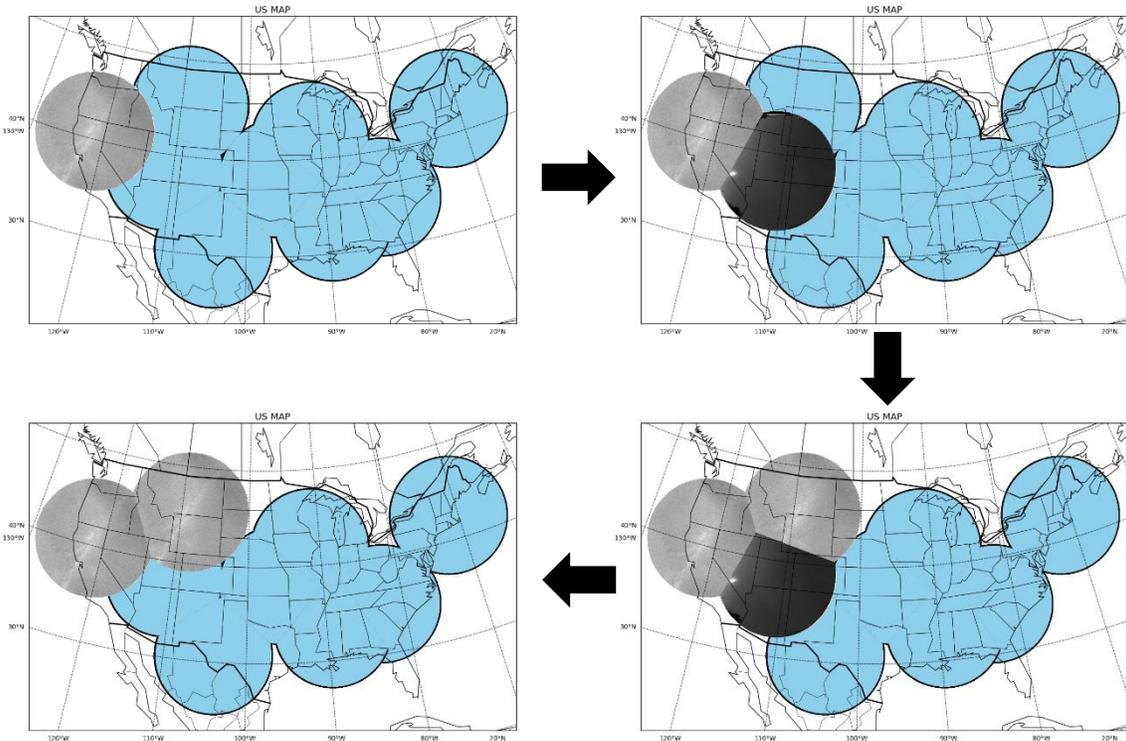


Figure 47. Example of real-time map as sites turn on and off. Blue circles are sites that are off.

### 4.1.2 User Interface

The system also has the capability for customization under many options. These options are: the map projection, whether to have the map show in the background, whether the all-sky images are transparent or not, the visibility of the latitudes and longitude lines, and the visibility of the title of the image. Some examples of this customization can be seen in Figure 48.

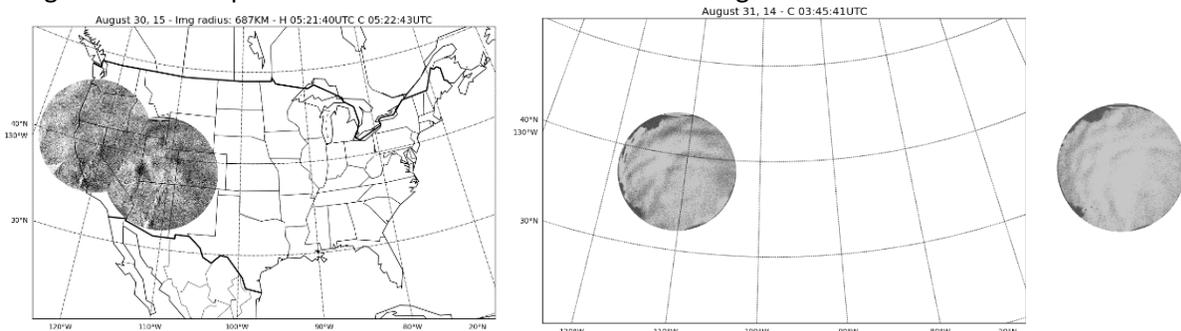


Figure 48. Different options for map display module. All features (left); No map, transparent image (middle); no background, transparent image (right).

## RESULTS

### 4.1.3 Image Integration

As mentioned, these images, in combination with the modules that display images on the map, can be used in conjunction with data from other sources. In our case we used data from the NASA AIRS project. We plotted the images over their different research data, such as thunderstorm data and cloud density data. Figure 49 below shows a python-generated map with one of the difference images plot over thunderstorm data captured by NASA. Ideally, the mages would correlate with the NASA AIRS data in terms of time and date, however we did not know all of the information for the NASA data, so the semi-transparent image in Figure 49 was arbitrarily chosen. Its location, however, is accurate. Mosaic images can also be used in the same application.

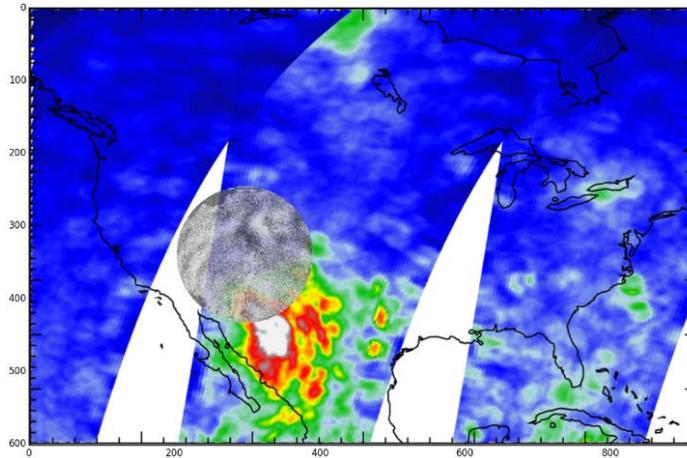


Figure 49. Plotted difference images over NASA AIRS research data

## 4.2 ENHANCED AND ANALYSED IMAGES

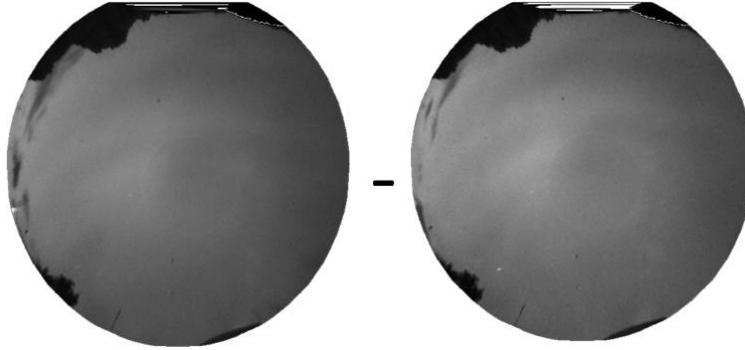
In this section we describe the results of enhancing the all-sky images, using those enhanced images for different applications, and analyzing the enhanced all-sky images.

### 4.2.1 Image Enhancement

Image enhancement was performed on images taken by the all-sky imagers, and it was done two consecutive images at a time. This was performed over each site in which we had available images with which to work. The resulting images were saved in the folder corresponding to the site in which they were taken.

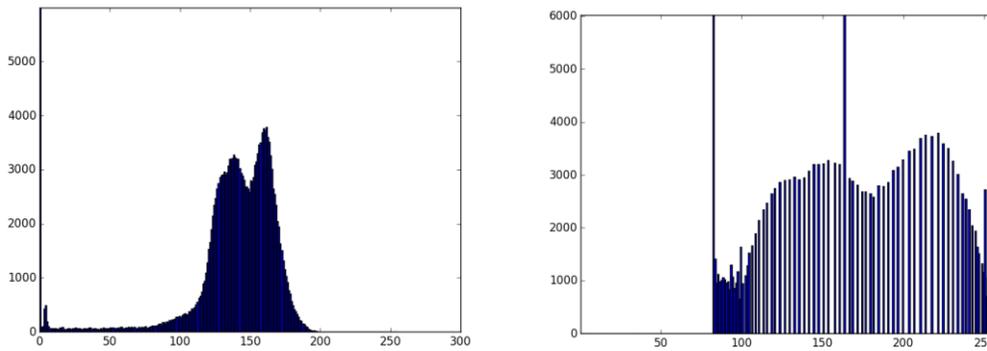
The first step was to perform image subtraction of two images taken one after the other, this process can be seen in Figure 50.

## RESULTS



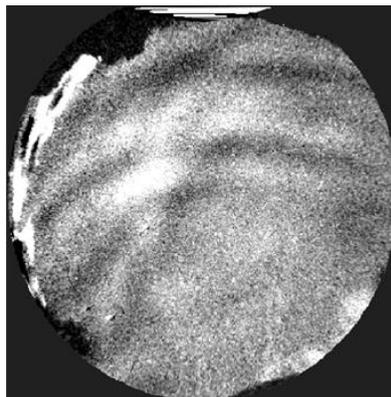
*Figure 50. Example of Image Enhancement. Two consecutive images are on the left which are being subtracted*

After performing the image subtraction, the histogram of this difference was obtained and then histogram equalization was performed. The result of histogram equalization can be observed below in Figure 51.



*Figure 51. Example of histogram equalization, original histogram (left) and equalized histogram (right)*

The resulting difference image can be seen in Figure 52. These difference images were then used in other applications and for image analysis due to the clear enhancement of the visual waves.



*Figure 52. Resulting Difference Image*

## RESULTS

### 4.2.2 Image Analysis

During the image analysis section of our program, we had to reduce the noise for each sample, sum up the vectors to find the wave direction, and then resample the image to calculate different attributes of the wave.

The fast Fourier transform was used to automatically reduce the noise in the pixel intensities. Even with the varying levels of pixel intensities and changes, it seemed to work well. An example of this work can be seen in Figure 53

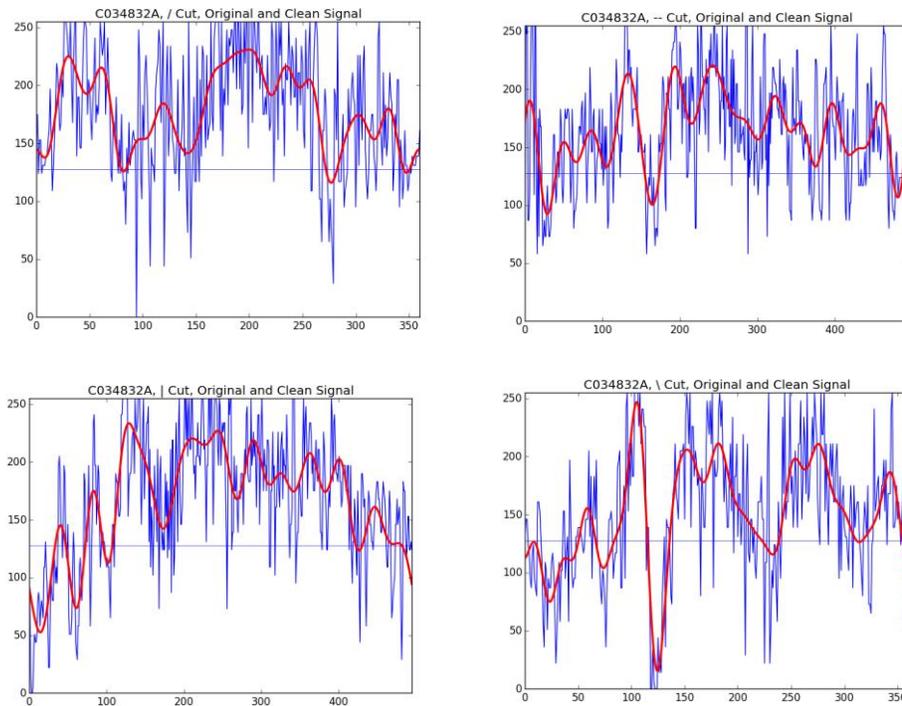


Figure 53. Original, noisy signal (blue) before FFT, and clean signal (red) after for each sample.

We next had to automatically calculate local minima and maxima, as well as filter for the most extreme of that list. The result of this can be seen in Figure 54.

## RESULTS

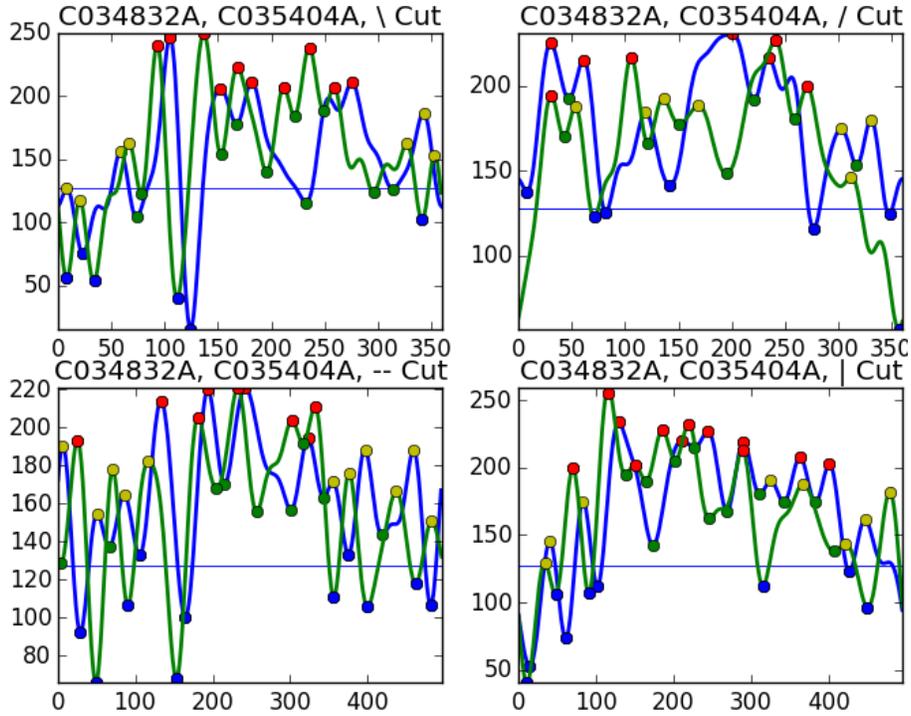


Figure 54. Calculated extrema (dots) for two images (blue and green lines) for each sample.

From this extrema data we calculated the magnitudes in each sample direction, plotted it, and calculated the vector sum. This can be seen in Figure 55. While this wave detection software is not perfect, it does seem to locate the overall wave direction pretty well. However, this program does not account for smaller waves along the image, it only looks for the overall wave going across the images.

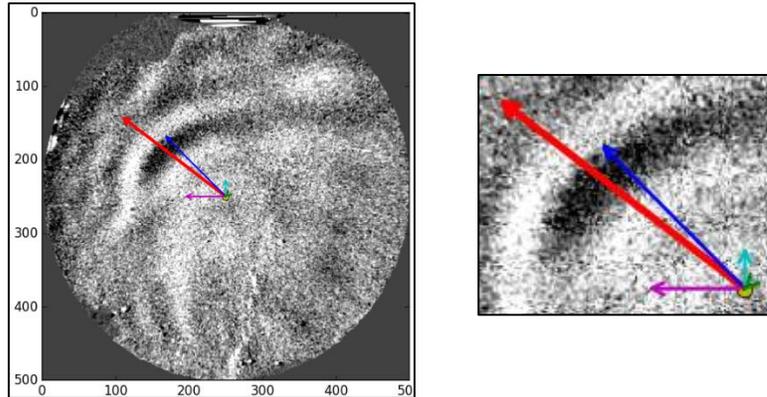


Figure 55. Vectors for each sample direction, bold red vector is automatically calculated wave direction. Image on right is zoom of vectors as seen in image on left.

We next resampled along the new vector direction, and then calculated wave features. Some results can be seen in Figure 56.

## RESULTS

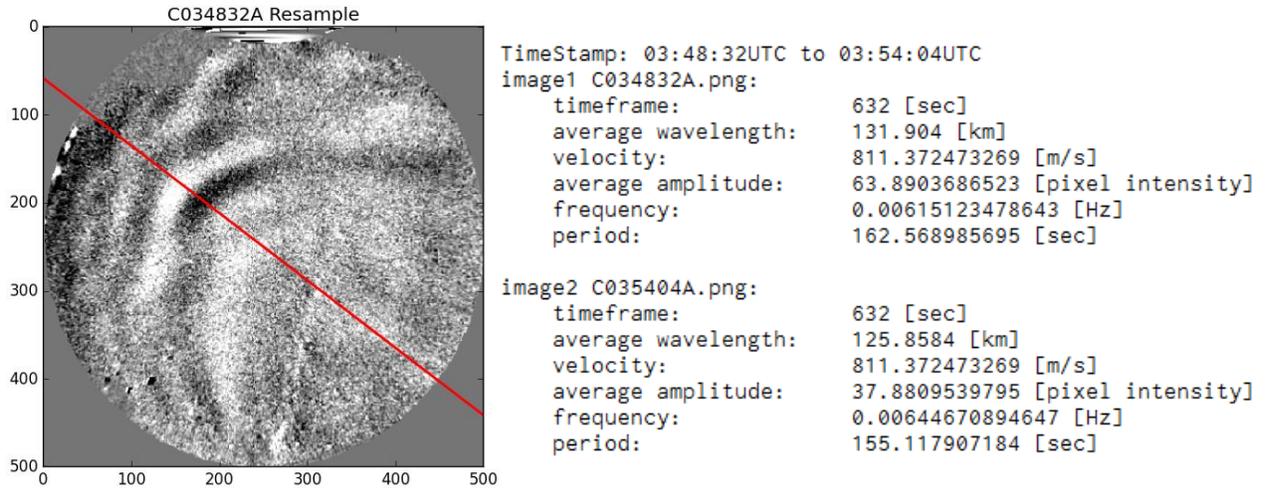


Figure 56. Resample for one of two images (left), results of image analysis and calculation for two consecutive images (right).

Upon conferring with our sponsor, Asti Bhatt, we have determined that these derived values are mostly within the range of expected and acceptable values.

However, we have also found that this software is not perfect, as we did not have enough time to fully address the calculations portion of this system as much as we had wanted. Figure 57 shows that our automatically chosen wave direction is not entirely on point. The wave direction was calculated somewhat, but not perfectly. Similarly, the calculated values are not too far-fetched, but are not entirely accurate either. Some of the calculations appear reasonable, such as the average wavelength, however other measurements, such as velocity, are erroneous. Future work would be needed to obtain better results. Although these results are slightly off, this system is a good proof of concept for automatically detecting wave features, and if improved, it will serve as a highly useful tool.

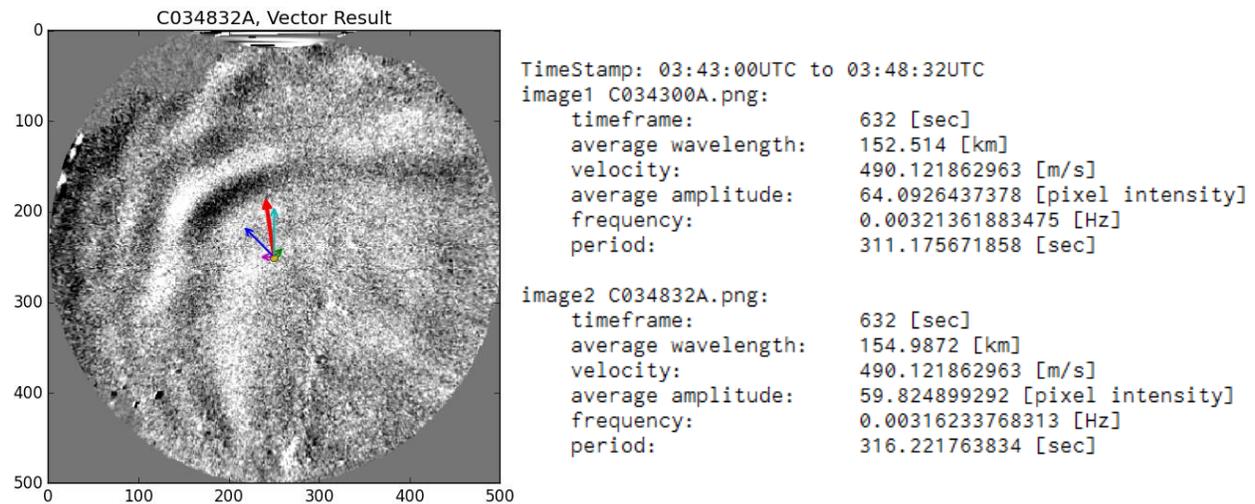


Figure 57. Automatic wave direction detection software not perfectly finding the wave direction. The wave is flowing in a much more NW direction than the bold red vector sum line would indicate (left). Calculated results, however, are not too far from expected values.

### 5 CONCLUSION AND RECOMMENDATIONS

In this section we present concluding statements and recommendations for future work to enhance the accuracy, speed, and overall implementation of our programs.

#### 5.1 Concluding Statements

The ionosphere, the upper atmosphere, from about 85km to 600km has charged particles. Because of the nature of these particles the Earth's atmosphere is susceptible to turbulence triggered by enhanced flux of energetic electrons emitted by the Sun. This turbulence causes the emission of photons in the Earth's upper atmosphere which creates a natural light to be seen, referred to as an Aurora. Airglow is another upper atmospheric phenomenon which is caused by solar radiation. Solar radiation energizes the atmospheric atoms and molecules during the day which relax later at night, thereby emitting photons.

A clear understanding of these phenomena is important to scientists because solar storms can disrupt satellite telecommunications and radio transmissions as a result of large disturbances in the upper Atmosphere. There are other effects these ionospheric waves have on communication systems. To study these phenomena, scientists capture atmospheric images using sky-imagers. Image acquisition and processing techniques are then performed on the pictures. These techniques, such as image stitching, star removal, are decisive in the understanding of ionospheric effects.

#### 5.2 RECOMMENDATIONS FOR FUTURE WORK

After analyzing our results, we have created some recommendations based on our two main deliverables, the map mosaic image system, and the image analysis system.

##### 5.2.1 Map Mosaic Image System

As with most any program, the system can certainly be improved in terms of speed and readability. However, the main recommendations we have for this system are to add more features for customization, such as specifying a location over which to plot the images, or a timeframe to plot the images, and to enhance the automation of the code. Currently a few loops had to be hard coded to allow for at most nine sites. If more sites are to be added, the user may have to go into the code and manually add more to it to accommodate for the new site.

##### 5.2.2 Image Analysis Recommendations

Although the wave direction analysis tool has been working fairly effectively, it does not always pick up the wave direction perfectly. Better or more rigorous methods can be looked into in order to improve the accuracy. Methods in comparing local maxima and minima, choosing which local maxima and minima are related between the images, and methods of preserving the overall pixel intensities without removing valuable data when smoothing the original pixel intensity signal may be investigated. The calculations after resampling the images must also be addressed to improve the accuracy of the results.

## APPENDIX 1 – SRI: ABOUT THE SPONSOR

SRI International is a nonprofit organization headquartered in Menlo Park, California. Founded in 1946, SRI International’s commitment can be found in their mission statement: “apply science and technology for the good of society”. This innovation center provides research, services, technology and licenses, advice, and more to government services and industry [1]. The American nonprofit is dedicated to Science, Technology, Engineering, and Mathematics (STEM) through continuous investments in research and development.

SRI offers a unique business model by conducting sponsored R&D. As explained in the organization’s website, “their focus is always on meeting client and market needs to create and deliver new value”. The revenue generated by SRI is reinvested to improve the organization’s capabilities, facilities, and staff. SRI defines innovation as “the creation and delivery of new customer value in the marketplace with a sustainable business model for the enterprise producing it”. The nonprofit organization has had approximately \$4 billion in sponsored R&D in the last decade, received over 4,000 patents, and has spun off 60 companies [33]. A flowchart of SRI International’s business model can be observed in Figure 58.

With over 2,300 employees worldwide the corporation is organized into seven R&D divisions. SRI has five disciplines that form the core of their systematic approach to innovation. These five disciplines guide their business model and lead to tangible results. For a visual explanation of the five disciplines refer to Figure 58 [34].

1. Important Customer and Market Needs: SRI ensures to focus on customer needs rather than focusing purely on research topics. This leads to positive impact for their clients, partners, end users, and the marketplace.
2. Value Creation: SRI’s methodic approach – Needs, Approach, Benefits, and Competition – helps them quickly define, create, and communicate high customer value. This approach applies to every project SRI works on and allows them to help their clients articulate their needs, define their approach, analyze the benefits, and quantify their approach.
3. Innovation Champions: a passionate advocate guides each project to deliver a successful project.
4. Innovation Teams: each team at SRI is multidisciplinary. Ranging from engineers and scientists to marketers, SRI has brought the best collaboration of people to meet their clients’ needs.
5. Organizational Alignment: SRI’s teams align with client and partner needs, not only SRI’s interests. Through best practices SRI will continue to improve their business by providing the best results to their clients.

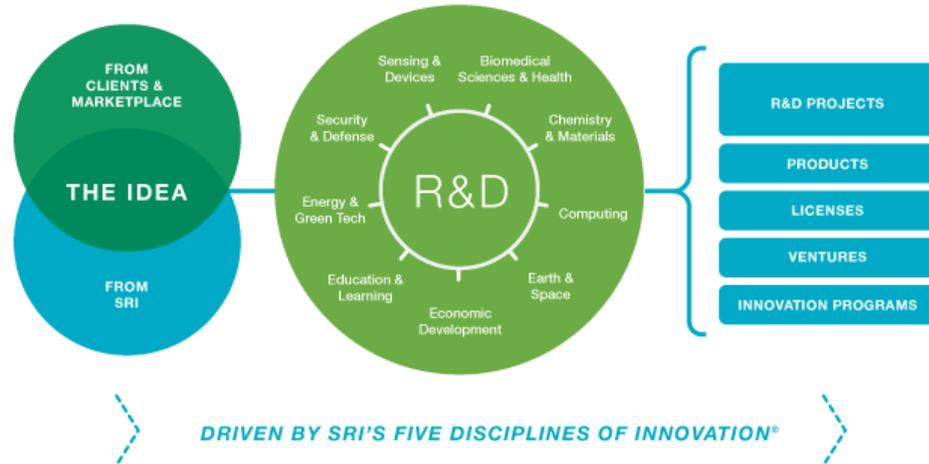


Figure 58. SRI's Five Disciplines of Innovation in association with their business model.

Due to SRI International's organization and success, the corporation has become a pioneer in the sponsored R&D industry. They have worked with multiple companies around the world and have helped launch over 60 spin-off companies as stated before. Some of the companies that have born from SRI International are E-Trade, Symantec, American Microsystems, and more. Additionally, some of SRI's most recognized patents and projects are: computer algorithms for Google, the computer mouse, the Internet, high-definition television, and Siri [34].

## APPENDIX 2 – VISUALIZATION OF IMAGES DISPLAY SYSTEM FLOWCHART

