**Multi-Sensor Image Fusion (MSIF)**

*Phu Kieu, Keenan Knaur*

*Project Advisor: Dr. Eun-Young Kang*

*Project Sponsor: Northrop-Grumman*

*Department of Computer Science*

*California State University, Los Angeles*

**Abstract**

**The goal of MSIF is to calculate the depth of objects in an image given a stereo pair. Due to the nature and availability of sensor parameters, the approach used in this paper is slightly different from the prevalent methods used now. Three methods that distinguish this are: image rectification, match finding, and depth computation.**

**Introduction**

The purpose of the project is to develop an algorithm to fuse two images of the same scene from two space sensors and extract some meaningful information from that fusion. The algorithm is implemented using the OpenCV library which is an open source computer vision library. The images used as input were data sets acquired from the GOES satellites operated by NOAA. A rectification of the images was the first step and an equirectangular projection was used to accomplish this. The images were then segmented to identify cloud pixels and then those pixels were grouped together to form cloud objects. Those identified objects were then put through one of two matching algorithms to come up with a disparity map. The disparity map was then used to calculate assumed altitude of the cloud objects over a certain range and the results are shown in a graph.

**1. Analysis of Input Data**

In this preliminary step, the data was thoroughly analyzed so the best method to achieve the desired results could be selected.

**1.1. Satellite Information**

The initial input images for the program were acquired from the National Oceanic and Atmospheric Administration (NOAA). These images were taken by their Geostationary Operational Environmental Satellites (GOES): specifically the GOES-11 (west) and GOES-12 (east) satellites. Both satellites are positioned 35,790 km above the surface of the earth with the west satellite at 135° W. longitude and the east satellite at 75° W. longitude. The image from the west satellite spans an area of 12°-60° N latitude/90°-175° W longitude. and the time it takes for the image to be scanned is 5 min, whereas the image from the east satellite spans an area of 20° S-66° N latitude/45°-120° W longitude and takes 15 minutes to scan the image. The actual area of overlap between the two images is an area of 12°-60° N latitude/90°-120° W longitude.

**1.2. Input Data Information**

Two separate data containers were acquired from their respective satellites. Each data container consists of 10 channels of information, the first of which is the images that were used as input to the program, and the rest which contain information such as surface reflectivity, aerosol signals, whether or not a pixel is good or bad data, and other types of information. The sixth channel is a cloud screen channel which indicates if a given pixel is a cloud or not with a value of 0 (cloud) or 1 (not a cloud). This channel was not directly used in the program but as an external measuring tool to test the accuracy of the cloud detection in the program. The images themselves are grayscale images of cloud cover above the surface of the earth. The grayscale is stored in raster order represents the Aerosol Optical Depth (AOD).

For each image, NOAA also provided separate files containing latitude and longitude information per pixel. The data was stored as floating point numbers in raster order and it was also used as input to the program for the image rectification/projection step.

**1.3. Challenges with the Data**

Because of the data used in the project, there were properties inherent to the data that presented some challenges. The first challenge was that the images were not in fact taken at the exact same time, but were taken over a period of time. One image took 15 minutes to scan, while the other image took 5 minutes to scan. During the scan time, it is entirely possible that the clouds shift positions and any disparity that is detected could be the result of cloud motion in addition to the disparity caused by stereo imaging. Another complication is the fact that clouds are only visible due to the sunlight that reflects off of them. It is also possible that disparity is caused by the fact that the satellites view the same cloud from different angles, and the shape of the cloud would appear to change depending on how the sunlight hits the clouds at those angles. These were somewhat accounted for during the altitude calculation step.

**2. Image Rectification**

The image rectification stage is a preprocessing step that allows the matching algorithms employed to function properly. The rectification is achieved by performing a map projection using the latitude/longitude. The equirectangular map projection is selected--as a result of the projection, latitude/longitude lines are parallel and equidistant from each other. The behavior of this projection is very simple to model and is one of the reasons it was selected.

**3. Image Segmentation and Object Extraction**

At this point in the program, the rectified images are put through an algorithm that identifies cloud pixels from the other pixels and then clusters the pixels together into cloud objects.

**3.1 K-Means Clustering**

K-means Clustering was used as a way to identify which pixels in the images actually belonged to the clouds. It is applied to the rectified images and is an iterative process where a random mean is selected in the beginning. With each added pixel, the mean is updated and the algorithm is rerun until all the pixels are separated into their respective bins. The program uses k = 3 because this was found to produce better results over k = 2. The pixels are clustered based on their intensity values and the result is three binary images, one of which contains most of the identified clouds. The other two images contain either pixels that are identified as land, or pixels belonging to objects that may be lower than the height of the clouds. The picture with the highest overall intensity values was identified as the picture containing the clouds since clouds will always have the highest intensity values. It should be noted that this is where the cloud mask channel was used as an external verification of the results. When compared with what NOAA identified as clouds, the program was about 85-90% accurate.

**3.2 Connected Component Analysis (CCA)**

The binary image identified as the image containing the clouds was used as input for this part of the program. The CCA takes the pixels and groups them together with neighboring pixels to form cloud objects. These objects were then used in the matching algorithms to find matches between them. During this stage objects of less than 500 pixels were ignored since small chunks of pixels identified as clouds are frequently not clouds.

**4. Matching Algorithms**

Matching algorithms are used to match a cloud object from one image, with a cloud object from the other image to find the best spot where it should match. Two algorithms were implemented and each calculates a disparity map which is then used for the altitude calculations.

**4.1 Mean-Squared Difference (MSD)**

The mean-squared difference is a commonly used method in block matching. The difference in application is that it operates on true variable size blocks, instead of a set of predefined dimensions. The bounding box of the cloud object is computed, and a block is constructed and used to compare over the search area. MSD takes the sum of the squared difference of the corresponding pixel values between the two equal size blocks.

**4.2 Shape Histogram**

The shape histogram matching algorithm calculates a feature vector for each cloud object that is representative of the shape histogram of that object. The feature vector is simply a count of the number of white pixels in each row, and then a count of the number of white pixels in each column. Each entry in the vector is that respective row or columns white pixel count. For example, if the following image were a cloud object:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 2 |
|  |  |  |  |  | 3 |
|  |  |  |  |  | 4 |
|  |  |  |  |  | 3 |
|  |  |  |  |  | 3 |
| 2 | 3 | 2 | 5 | 3 |  |

Figure 1. Feature vector of a binary image

The feature vector would be . This vector is calculated for each object in the respective images and is then used to match an object from one image to a cloud object in the second image. The best match occurs when the Euclidian distance between the two feature vectors is at a minimum.

Figure 2. Cloud and ground positions with respect to satellites

Satellite 1

Satellite 2

Ground (earth) Plane

(Perceived Cloud Disparity)

Ground Point

Cloud Centroid

**5. Cloud Altitude Calculations**

To extract some meaningful information, methods have been devised to calculate the altitude of the cloud given the disparity, and the velocity of a cloud given an altitude.

**5.1 Finding the Altitude**

Let us consider a 3D point (e.g. green or blue dot in the diagram) that are observed by the two satellites and project the point onto the ground plane. If the point is on the ground plane (green dot), then the two projected positions coincide. If the point is not on the ground plane (blue dot), the two projected positions do not coincide and they generate a disparity due to the altitude .

The two projected images generated from the two satellite images contain the ground intersect for each pixel. In these images, any Earth surface point appears at the same location. A cloud with altitude, however, produces a disparity.

When a cloud is matched between the two images and a disparity is observed, the altitude of the cloud can be computed using the two satellite positions , which positions are known (Cartesian coordinates will be used). are the 3D ground intersect positions of the centroid of some cloud with respect to the satellites.

Construct two lines connecting to and to . Compute the intersection point of these two lines. This will yield the 3D location of the cloud’s centroid, and the cloud’s height can be extracted. In most cases, the two lines may not intersect. To cope with this problem, we will compute the point where the distance between these two lines is at a minimum.

**5.1.1 Proposed Solution**

Let be the locations of the satellites, and be the location of the ground intersect of the cloud at its centroid coming from the respective satellite.

The vector pointing from the satellites to the corresponding ground intersect is

The line segment connecting these two points is then defined by

The traditional method for solving for the intersection of these two line segments is highly likely to not have a solution; therefore the shortest vector connecting the two will be computed. Before we can compute this, must be normalized.

Let

Now we define the shortest vector connecting the two line segments as

The midpoint of this vector is simply

This is the 3D position of the cloud’s centroid. The height of this cloud is then

Satellite 1

Satellite 2

Ground (earth) Plane

(Perceived Cloud Disparity)

True Altitude

Calculated Location

‘True’ Location

Figure 3. Deviation of cloud altitude due to motion

**5.2 Plotting the Altitude and Velocity of a Cloud**

With the location of the some cloud’s reference point, alter it so that its altitude will be while maintaining its position with respect to the earth’s latitude and longitude. The lines which connect the satellites to this new location can be constructed. With this, calculate the intersection point of these two lines with the earth. These are the ground intersects of the cloud caused only by parallax. The true disparity of the cloud is now known. Find the difference between the displacements of the perceived ground intersects with that of the true ground intersects. This is the displacement of the cloud—how much it moved during the time it took to capture the image.

**5.2.1 Proposed Solution**

Let the perceived location and arbitrary true height of some cloud’s reference point be denoted by and . can be shifted to altitude without changing its latitude and longitude position with respect to the earth. It is done by converting to spherical coordinates , setting , and then converting it back to Cartesian coordinates. Let us call this new point . We assume to be the true location of this reference point.

Construct the lines connecting the satellites to and on.

Now find where these two lines intersect the earth. Let us assume that the earth is a sphere with radius . The point on earth where it intersects is found by solving the system of equations of with constraints (earth’s average radius).

The solution is composed as follows:

This can be solved through a routine application of the quadratic function. This will yield two solutions—pick the lowest value of ; it is expected that the first intersection will result when is at a minimum. The solution is expected to exist and .

The two points obtained from this is the true ground intersects of the cloud’s reference point if the cloud did not move at all.

Now, let be the perceived and be the true ground intersects. We propose the following approximation

where is the perceived disparity, and is the cloud motion. The distance between two ground points will be computed using the Haversine formula.

This distance is translated into the distance relative to the true altitude of the cloud, using the relation for a circle.

The translation is simply

The data set currently being worked on has a time difference of about minutes. We will divide by to convert this figure to cloud speed. With this, we can plot cloud altitude vs. speed over the feasible ranges of altitude.

Figure 4. Altitude vs. Velocity plot

**6. Data Visualization/Results**

With the solutions to the altitude problems solved, a plot is created to observe the correlation between altitude and velocity. Their relationship appears to be linear, with a fairly steep slope. The actual altitude of the cloud can be narrowed using the fact that clouds do not generally travel faster than about 200 km/h. With this, it can be concluded that its altitude lies between 6 and 13.5 km/h.

**Conclusion**

MSIF as it stands works pretty well. It is able to detect cloud objects with 85-90% accuracy and is able to find matches of cloud objects between the two images.

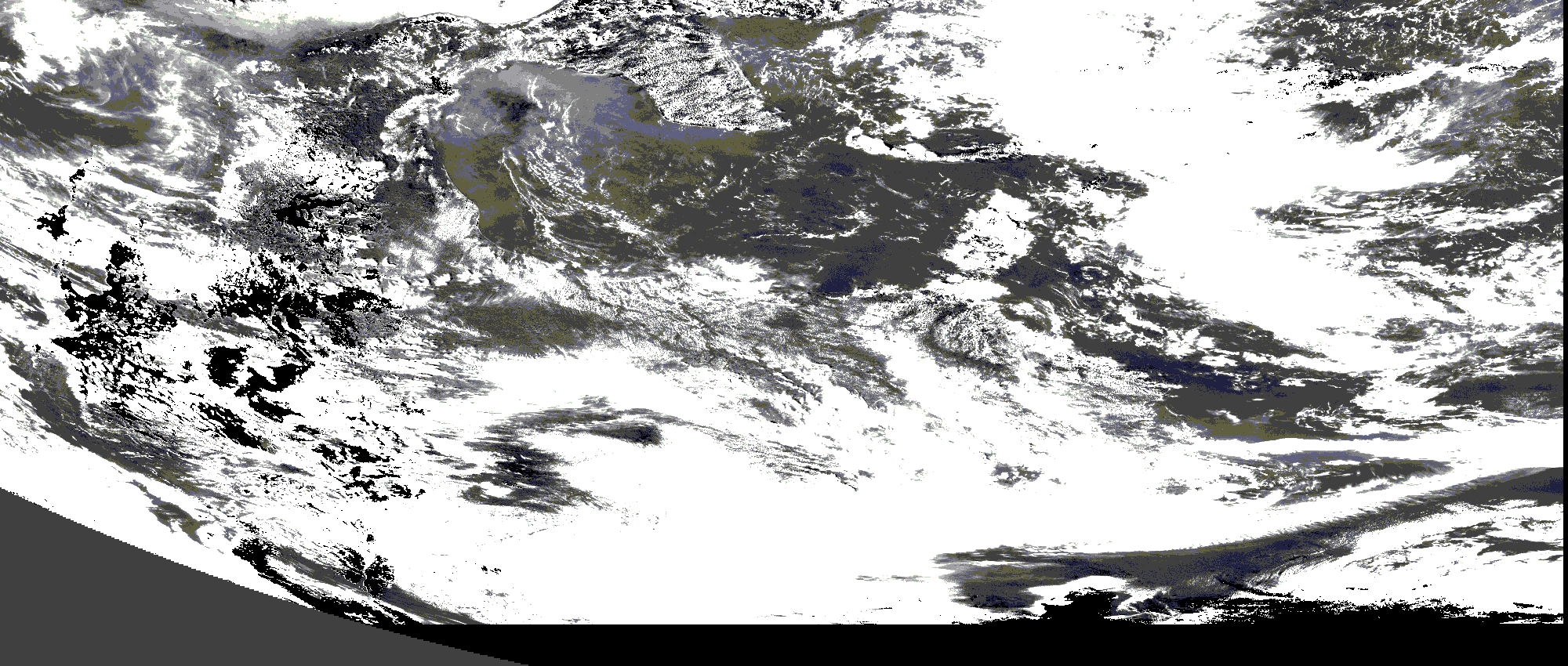
Using these matches it fuses the two images into a disparity map which is then used in the cloud altitude calculations.

It should be noted that the computation for finding velocity is fairly flawed. It operates with the assumption that any movement of the cloud will directly contribute to the disparity. However, it is possible for cloud movement to decrease the observed disparity of a cloud, and if the movement is orthogonal to the disparity vector, then the calculated velocity is higher than it should.

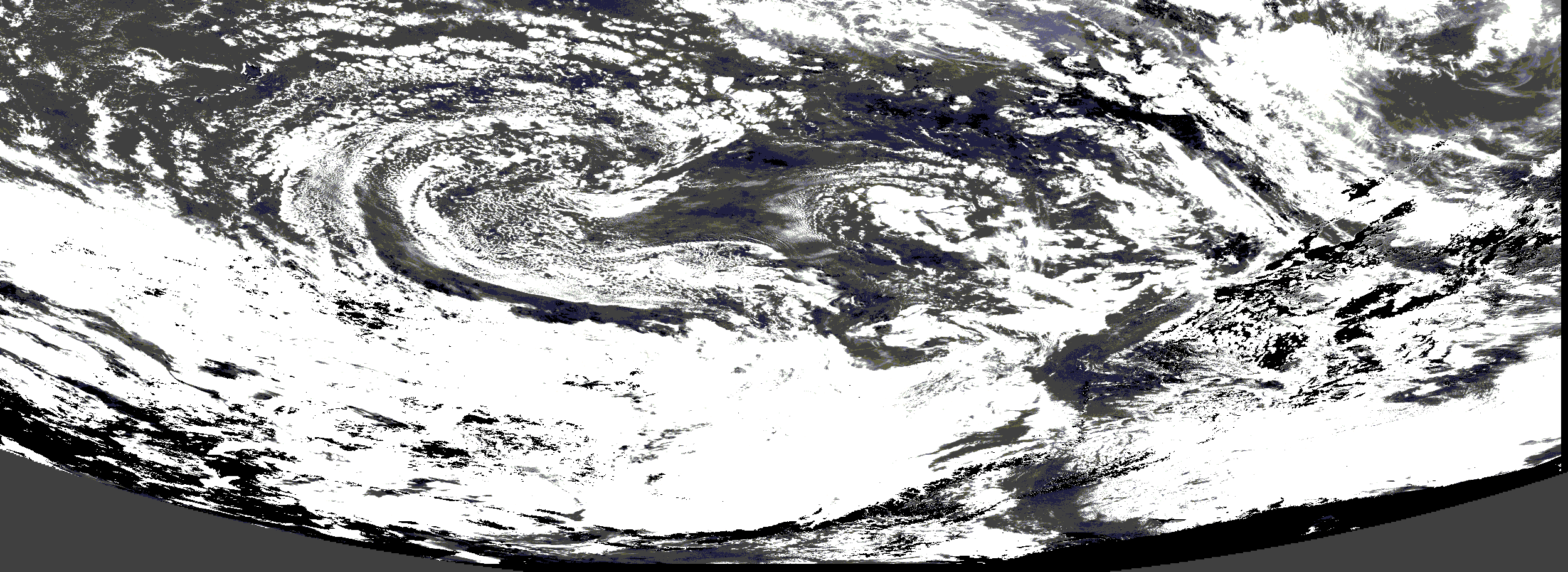
Another source of inaccuracy lies with the map projection. A projection which preserves distances is ideal, but is not used. By doing so, the disparities of the clouds may be more accurate, as well as the centroid. With more time and further refinement, it would be possible to make the program even more accurate.

Despite these inaccuracies, MSIF is able to take complex cloud images from the GOES sensors, and translate those images into some meaningful information.

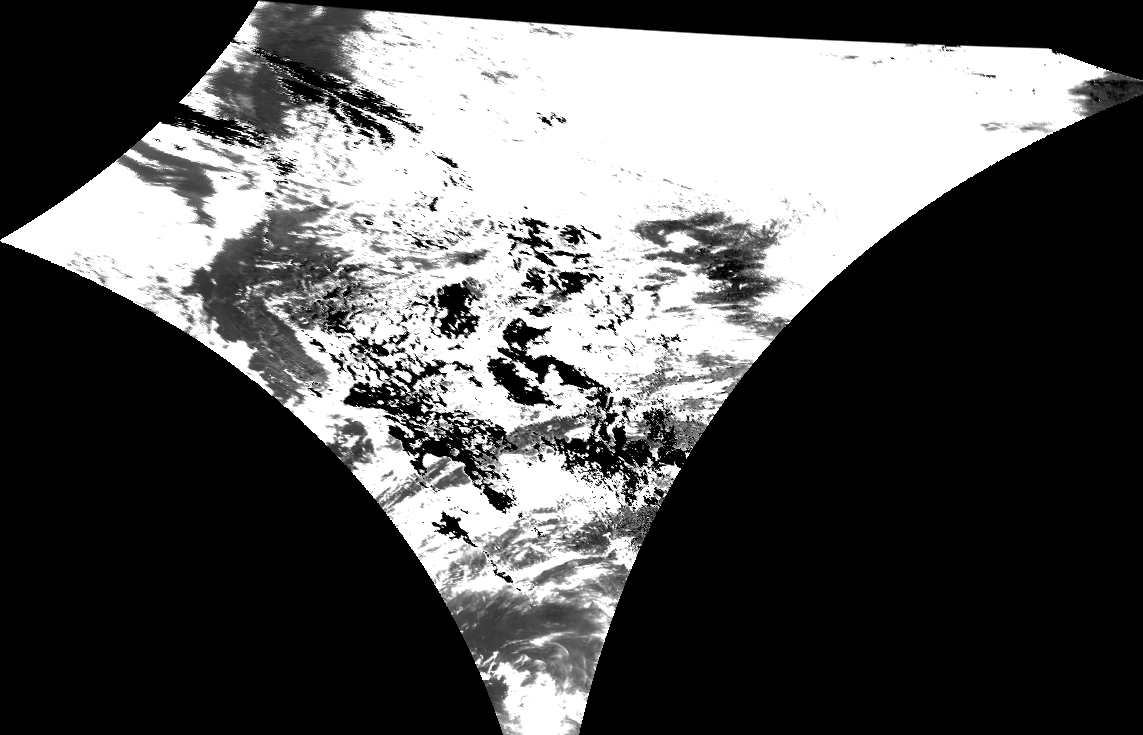
**Appendix**



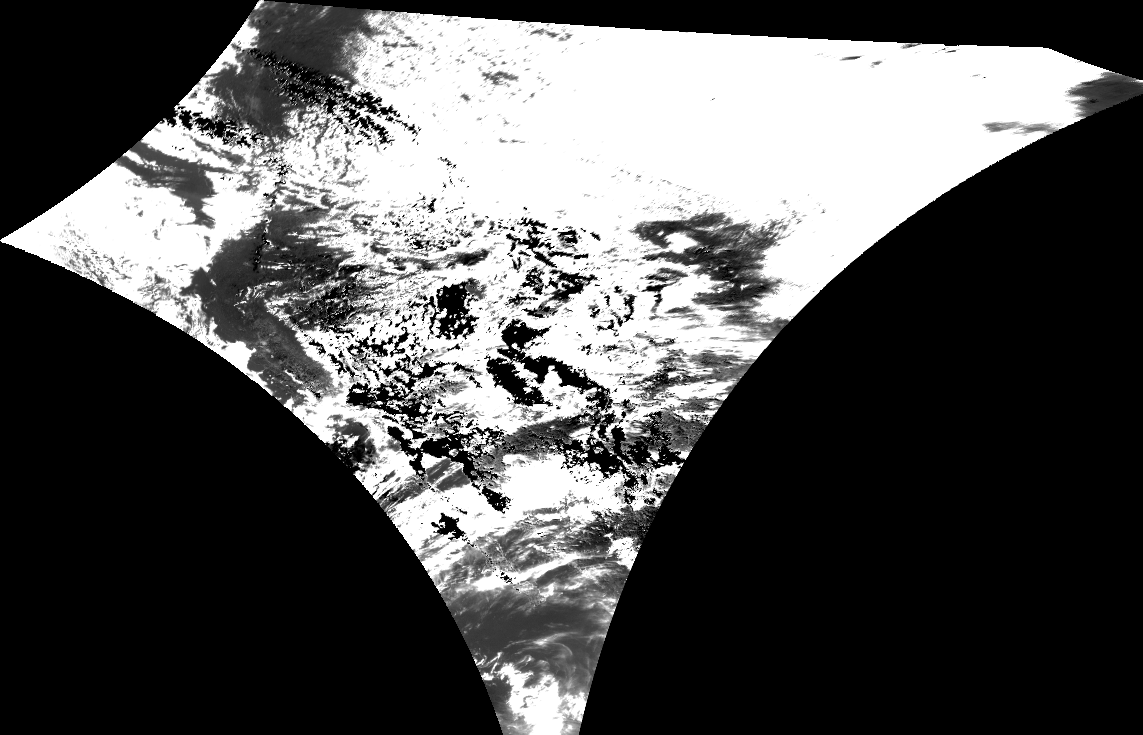
West Image: 2500x912

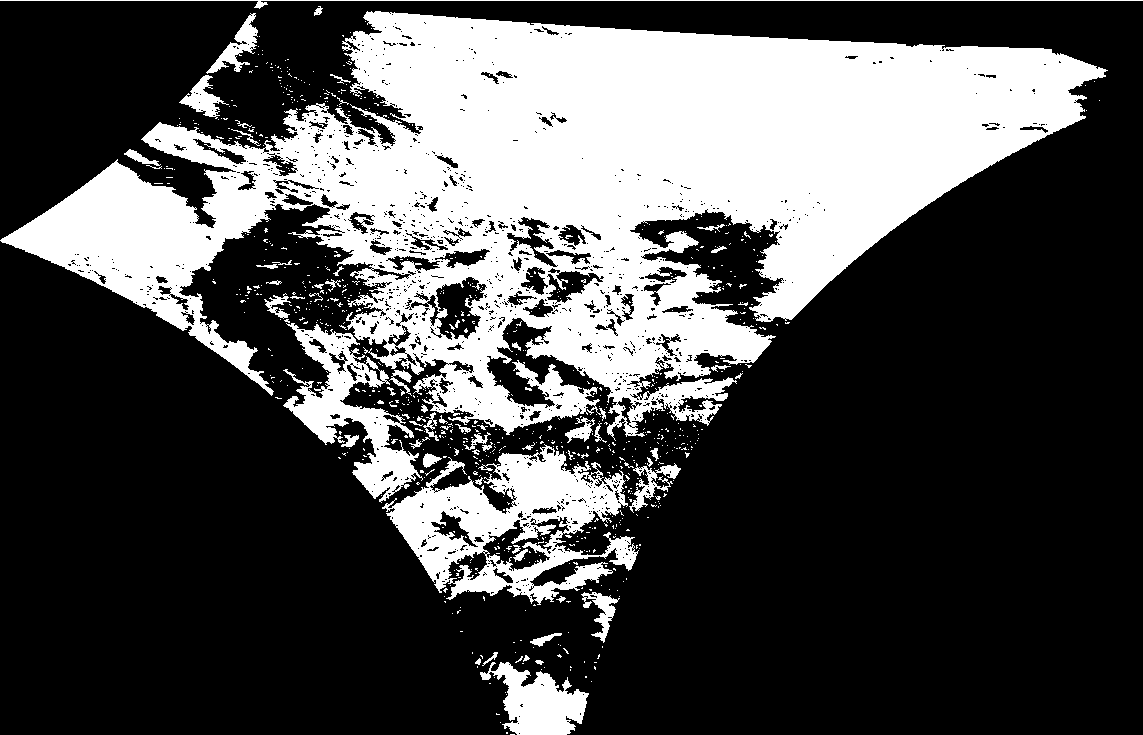


East Image: 2000x850

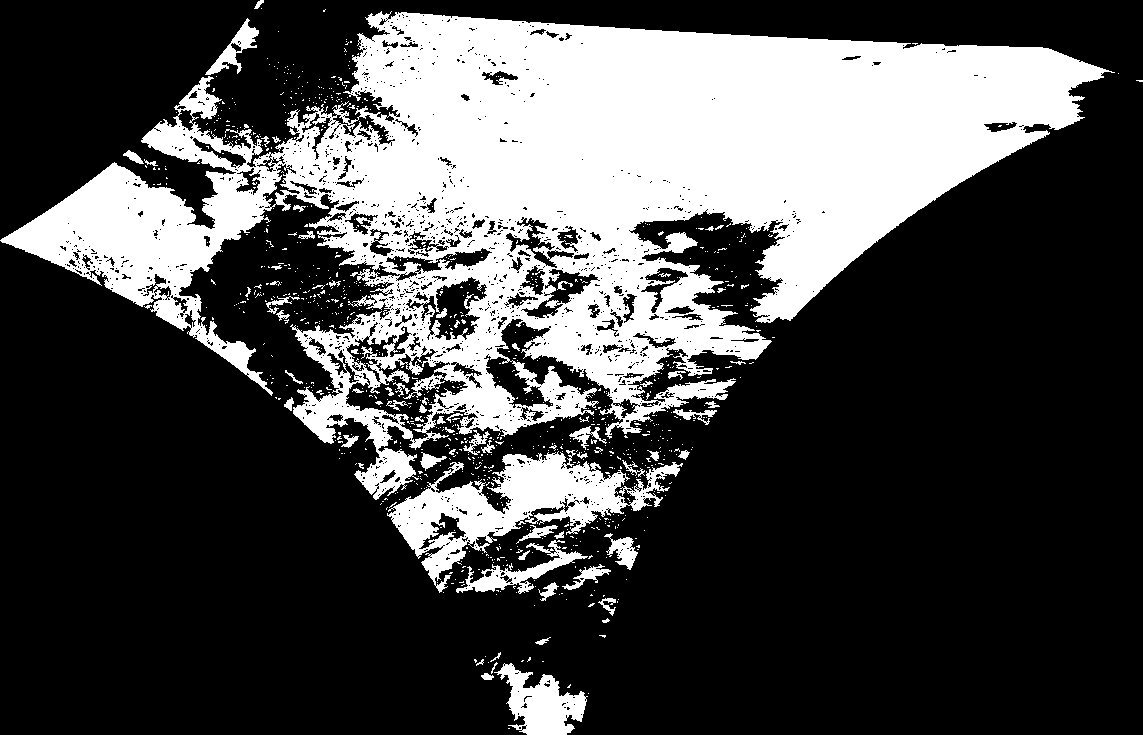


Equirectangular Projection: West 1143x735

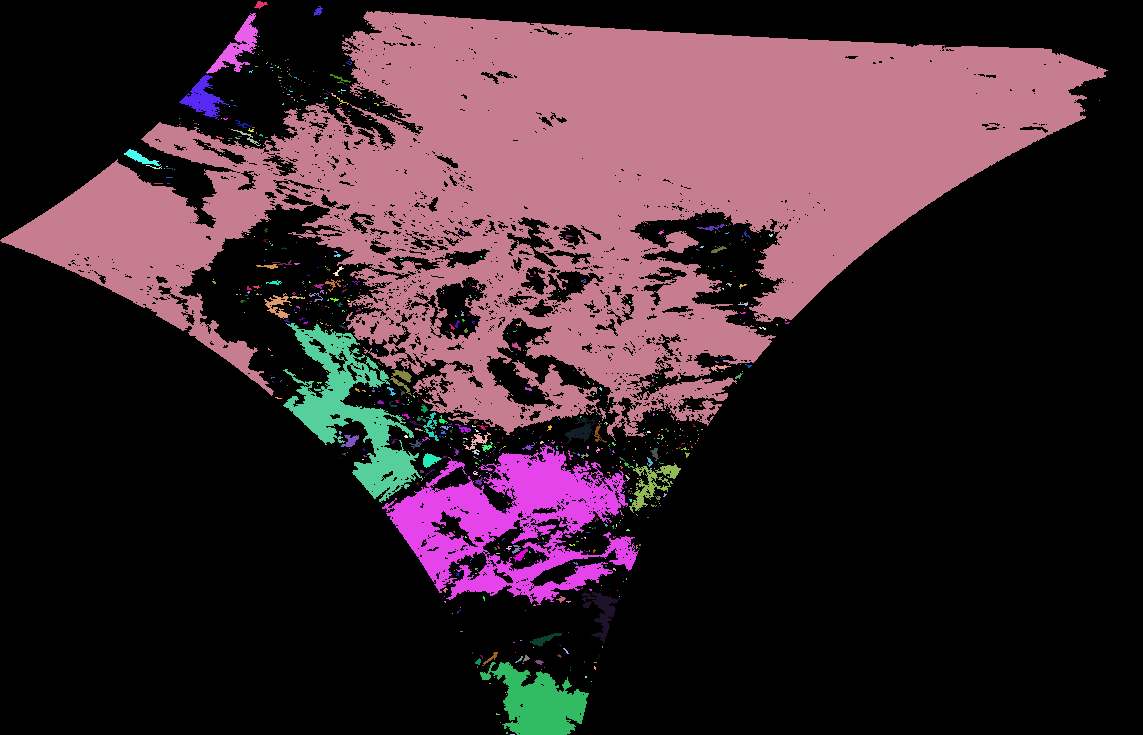
Equirectangular Projection: East 1143x735



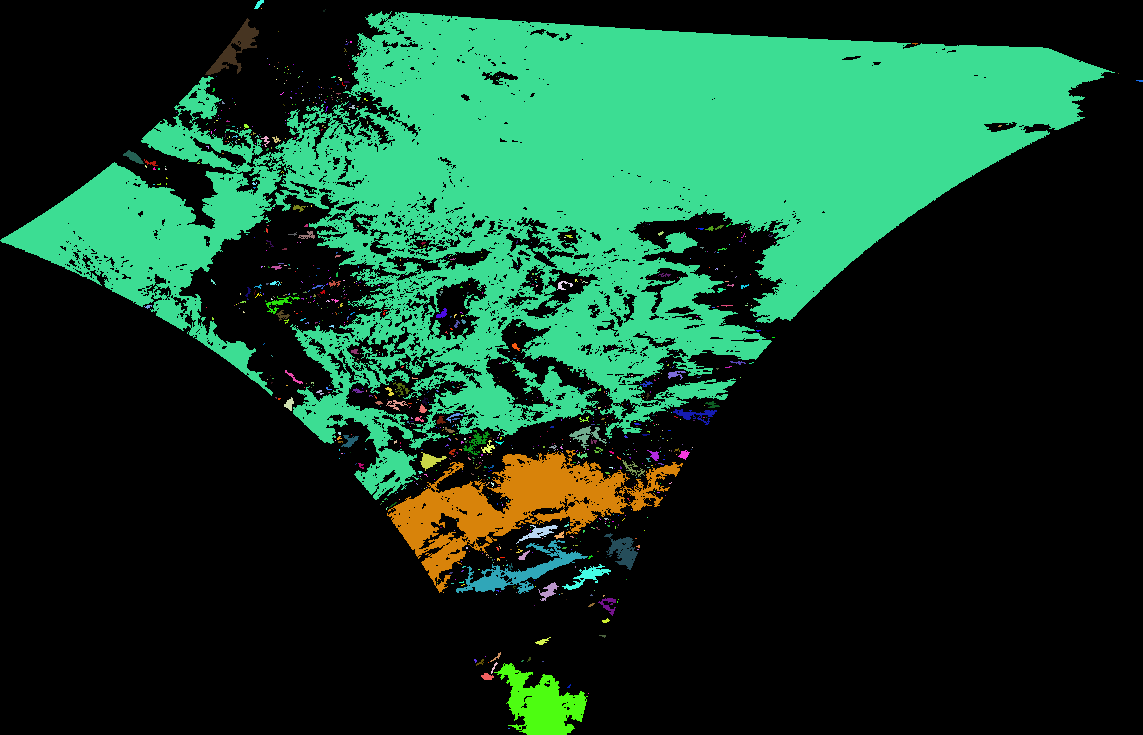
K-Means, Cloud Layer: West



K-Means, Cloud Layer: East

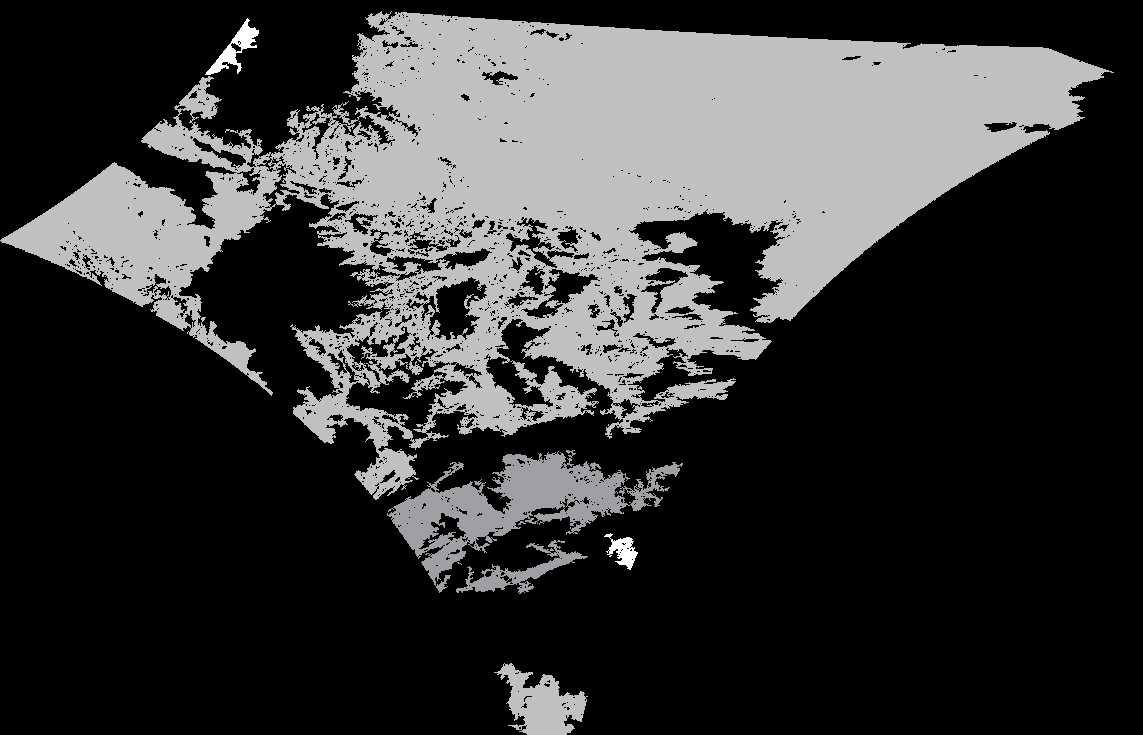


CCA: West

CCA: East



Disparity: West



Disparity: East

**References**

[1] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, Sebastopol, CA: O'Reilly Media Inc., 2008.

[2] USA. U.S. Department of Commerce, National Oceanic and Atmospheric Administration (NOAA), and National Environmental Satellite, Data, and Information Services (NESDIS). *Earth Location User's Guide*, U.S. Department of Commerce, 1998.

[3] D. Fanning, "Navigating GOES Images," March 2008. [Online]. Available: http://www.dfanning.com/map\_tips/goesmap.html [Accessed: Mar. 10, 2010].

[4] "Mercator Projection", Feb. 27, 2010. [Online]. Available: http://en.wikipedia.org/wiki/Mercator\_projection [Accessed: Mar. 10, 2010].

[5] Weisstein, Eric W, “*Mercator Projection*,” [Online]. Available: http://mathworld.wolfram.com/MercatorProjection.html. [Accessed: March 10, 2010].

[6] Shapiro, Linda G. and Stockman, George C. *Computer Vision*. New Jersey: Prentice Hall, 2001.

[7] Space Science and Engineering Center (SSEC), University of Wisconsin-Madison. "McIDAS Learning Guide, Satellite Imagery-Basic Concepts," 2008. [Online]. Available: http://www.ssec.wisc.edu/mcidas/doc/learn\_guide/2008/sat-1.html. [Accessed: Mar. 10, 2010].

[8] National Oceanic and Atmospheric Administration (NOAA). "GOES" Oct. 19, 2009. [Online]. Available: http://www.nsof.class.noaa.gov/release/data\_available/goes/index.htm. [Accessed: Mar 10, 2010]