2016-09

Horizon detection in the visible spectrum

Crisman, Donald M.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

HORIZON DETECTION IN THE VISIBLE SPECTRUM

by

Donald M. Crisman

September 2016

Thesis Advisor: Mathias Kolsch
Second Reader: Daniel Bursch

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE HORIZON DETECTION IN THE VISIBLE SPECTRUM			**5. FUNDING NUMBERS**
6. AUTHOR(S) Donald M. Crisman			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

In the last few decades, machine learning and computer vision techniques have enabled precise and repeatable image recognition. Computer vision techniques can also recognize star patterns in star trackers for satellite attitude determination. Horizon detection in the visible spectrum was largely discarded for attitude determination in favor of thermal imagery, due to the greater consistency of the earth's thermal radiation. This thesis examines computer vision and machine learning techniques to develop a horizon detection algorithm for the visible spectrum.

By examining different features of visual imagery, machine learning techniques were evaluated on the ability to detect a visible horizon and determine its orientation. An empirical analysis of visual imagery from low-earth orbit was conducted to develop a horizon brightness transition model, which allows for consistent and adjustable determination of the horizons location.

The final result is a horizon detection and orientation determination algorithm that successfully indicates if a horizon is present in an image with 96% precision and 92% recall. The brightness model correctly identifies the location of the horizon in 85% of the tested image set.

14. SUBJECT TERMS attitude determination, machine learning, image classification, earth horizon sensor, computer vision, line detection, visible horizon, visible spectrum imagery, horizon detection			**15. NUMBER OF PAGES** 73
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

HORIZON DETECTION IN THE VISIBLE SPECTRUM

Donald M. Crisman
Lieutenant, United States Navy
B.S.E., University of Michigan, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Mathias Kolsch
Thesis Advisor

Daniel Bursch
Second Reader

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In the last few decades, machine learning and computer vision techniques have enabled precise and repeatable image recognition. Computer vision techniques can also recognize star patterns in star trackers for satellite attitude determination. Horizon detection in the visible spectrum was largely discarded for attitude determination in favor of thermal imagery, due to the greater consistency of the earth's thermal radiation. This thesis examines computer vision and machine learning techniques to develop a horizon detection algorithm for the visible spectrum.

By examining different features of visual imagery, machine learning techniques were evaluated on the ability to detect a visible horizon and determine its orientation. An empirical analysis of visual imagery from low-earth orbit was conducted to develop a horizon brightness transition model, which allows for consistent and adjustable determination of the horizons location.

The final result is a horizon detection and orientation determination algorithm that successfully indicates if a horizon is present in an image with 96% precision and 92% recall. The brightness model correctly identifies the location of the horizon in 85% of the tested image set.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

COTS	commercial off-the-shelf
CubeSat	cube satellite
EHS	earth horizon sensor
ISS	international space station
KNN	k-Nearest Neighbors
LEO	low earth orbit
MATLAB	Matrix Laboratory (Mathworks, Inc.)
MAV	micro air vehicle
NASA	National Aeronautical and Space Administration
OpenCV	Open Computer Vision Library
RANSAC	random sample consensus
RGB	red-green-blue
SVM	support vector machine

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisors for all of their help in developing the ideas presented in this thesis to the fullest as well as for their assistance in shaping the development of my academic writing ability.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

In the late 20th century, work on horizon detection for satellite attitude determination hinged on thermal sensors. As autonomous systems and unmanned aerial drones came into focus in the early 2000s, there was a growth in research covering horizon detection in the visual spectrum in order to determine aircraft attitude from low altitudes.

Advances in horizon detection from space at the academic level have even been explored for the development of smaller satellites in cube satellite (CubeSat) programs at many universities. CubeSats are intended to be small, inexpensive satellites with short term research objectives. CubeSats allow universities to explore using commercial off-the-shelf (COTS) equipment as a low-cost alternative for conducting research in space. At the Naval Postgraduate School (NPS), this has led to projects designed around building sensors and equipment with COTS components that match the size and price point of CubeSats, sacrificing as little accuracy and space reliability as possible. One example being the star tracker prototype developed at NPS in 2015 [1] and another is the investigation into the development of accurate star trackers in [2].

With the constraints on CubeSats to be low cost and minimal size, all the sensors should be efficient. An efficient sensor would have multiple purposes, or in this case for determining the orientation of the CubeSat with the horizon and/or a star field. In the case of a star tracker and a visible spectrum camera, if the sensor is pointed toward the horizon, it should then be capable of horizon detection in order to continue to provide useful information to the attitude determination system. Although the algorithm for determining attitude utilizing horizon detection requires a second sensor for the three axis attitude solution, this thesis worked to create the possibility of a sensors that can both do star tracking and horizon detection.

In addition to the specific space application, this research hopes to identify strategies and techniques that can be later translated to horizon finding in imagery from

aerial and terrestrial cameras to enhance the research already being conducted in that field. As addressed by Grace Young, horizon detection is important for identifying and tracking ships on the ocean surface [3]. Detecting the horizon with the naked eye and using a sextant for celestial navigation has also been necessary for centuries. Locating the horizon over the ocean imagery with a computer would be a key first step to automating this type of navigation for ships.

B. SCOPE

The purpose of this research was to explore methods that can be utilized to determine the location of the earth's horizon in satellite imagery and poses the following questions:

1. How is the horizon of the earth defined in the visible spectrum?
2. Can the horizon be modeled to provide consistent and robust identification?
3. What machine-learning techniques can be utilized to automatically detect the presence of a visible horizon in digital imagery?
4. What characteristics of the imagery and machine learning techniques can be utilized to automatically detect the orientation of a visible horizon in digital imagery?
5. What curve best models the shape of the horizon?

The first question can be answered in some respects using current computer vision techniques to determine a horizon line, but in the absence of ground truth, these techniques provide little insight and are difficult to evaluate against one another. This is the reason the second question is so important. By defining a model, a consistent method of detection can be developed.

To illustrate this point, Figures 1 and 2 show a picture of a horizon. Even though it is clear that a horizon exists in this image, notice how difficult it is even for a human to find the exact point, the exact pixel, where the earth gives way to the atmosphere and the darkness of space.

Figure 1. Difficult Horizon Detection. Adapted from [4].

Figure 2. Zoomed Difficult Horizon Detection. Adapted from [4].

Even in Figure 2, which is simply cropped and resized to provide a closer look at the horizon, it is difficult for the human eye to distinguish where the earth ends and the atmosphere and space begin. A major focus of this thesis is to construct a horizon brightness transition model (HBTM) that allows an automated system to effectively and

consistently identify the location of the earth's horizon in imagery similar to Figures 1 and 2. This identification of the precise pixels that compose the horizon is referred throughout this thesis as the horizon determination problem.

The second focus of this thesis was to define a technique to determine whether or not an input image contains a horizon and is referred throughout this thesis as the horizon detection, or two-class classification problem. Taking a closer look at Figure 2, it may be difficult for a human to realize that there is a horizon. This thesis specifically investigated a k-Nearest Neighbors (KNN) and Support Vector Machine (SVM) approach at identifying the presence of a horizon in an image.

The implementation of the horizon determination method was more efficient when the horizon's orientation is known, and therefore these machine learning techniques were also evaluated in their ability to classify different horizon orientations in a nine and eight-class classification problem. This thesis evaluated different features of the images for their suitability for horizon detection and orientation classification tasks.

As a final focus, this paper also examines curvature models for the horizon. The HBTM allows the algorithm to select pixels that indicate the position of the horizon in an image, and the curvature model assists in fitting a curve to these pixels. Because it is possible that the HBTM will generate some outliers based on noise in the data, a linear, parabolic, and circular model were examined to determine the most robust model for the shape of the horizon. This thesis assumed no specific optical parameters, but instead attempted to find the generic model that best applies to COTS optics and legacy images with unknown sensors and optics.

This algorithm was not implemented for a specific star tracker hardware model at NPS or flight tested with a full attitude determination model. It was intended as a generic method, suitable for a set of diverse optics and sensor hardware. Correspondingly, the data set for estimating the parameters of the horizon brightness transition model does not focus on a single field of view, but was rather gathered for the purpose of testing the models against diverse and widely different lighting and brightness conditions. The larger

data set for testing the detection, categorization, and determination algorithms are, however, selected for being smaller fields of view, similar to that of a star tracker.

C. LAYOUT

Chapter II provides a brief background in earth horizon sensors (EHS) and current horizon detection techniques from several different types of imagery. Chapter III outlines the methodology for determining the models that shaped the resulting algorithm. It is followed by Chapter IV, which outlines the experiments conducted to test the hypotheses. Chapter V presents and briefly discusses the results, and Chapter VI concludes the findings and presents opportunities for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

In examining previous research in horizon detection, valuable insights for this research were discovered both from work on horizon detection from space and for vehicles that operate within the atmosphere. Much of the previous work in horizon detection from space was done with infrared sensors and more recent work transitions into the visible spectrum, though previous horizon determination methods do not attempt to model it as this thesis suggests.

A. EARTH HORIZON SENSORS

Development of electronic earth horizon sensors for satellites started in the latter half of the 20th century and was mostly limited to thermal sensors because the earth's emitted radiation in this spectrum provides for a more consistently defined horizon [5]. Techniques for calculating satellite attitude using EHS and other sensors have been around for just as long [5], [6].

For recent developments, visual nadir and earth horizon sensors have been theorized and constructed in several projects, such as the SumbandilaSat [7] and in developing attitude sensing for a solar sail CubeSat [8]. This largely demonstrates a change from the more hardware based models in [9] with radiance detectors and signal processing units, to the software-based models in [7] and [8].

B. DEFINING THE HORIZON

The horizon, according to the *Oxford English Dictionary,* is "the line at which the earth and sky appear to meet" [10]. However, identifying the horizon in visual imagery is no easy task. As demonstrated in Chapter I, even for a human this is difficult.

It is the general consensus of the research cited in this chapter that the computer vision problem of finding the horizon rests on finding the precise pixels in the image that represent the boundary between the earth and sky and that this boundary defines the horizon. In some work on finding the visual horizon for attitude determination of micro-air vehicles (MAV), the earth included all things with similar colors that were grounded

to the earth, such as buildings, trees, and other vertical extensions of the earth. This was advantageous for the application as there was also interest in keeping the MAV from colliding with these structures [11].

The same principle applies to this work in that it must be clear which pieces of the image constitute pieces of the earth, the atmosphere, and space. There is a great difference in the type of imagery between that of an MAV which operates at low altitude and that of a satellite which operates in space. The precise level of detail for trees, buildings, and mountains does not exist for small satellites.

Since a mathematical model is being developed, it is important to examine other mathematical models for the horizon. One specific application is horizon "dip." This is due to the refraction of light at various altitudes and is a well understood phenomenon that is modeled based on temperature and pressure in localized areas for low altitudes [12]. Because the model draws from environmental factors local to a specific area and within the atmosphere, there is no way to account for this in a sensor from space. Therefore, this model was not be integrated into the model developed in this thesis.

C. FINDING THE HORIZON AS AN OPTIMIZATION PROBLEM

1. Features

In continuing to evaluate Scott Ettinger's work for MAVs, he proposes a method to find the horizon by classification in the color feature space of an image under the assumption that earth pixels share similar color characteristics as other earth pixels and the same for sky pixels [11], [13]. That sky pixels tend to be of a blue, gray, and white mixture, whereas earth pixels have a much more variable color makeup.

2. Optimization

The second major assumption of Ettinger's work is that the horizon presents itself as a straight line [11]. By modeling each set of pixels as a set of Gaussian distributions over each color channel and dividing the pixels into two sets via the proposed horizon line, the covariance of the two distributions can be evaluated. From these covariance matrices an optimization function was created in order to minimize the variance of the

colors of the pixels [11]. Therefore, maximizing this optimization function minimizes the differences in color between the two sets. That is, the function is maximized when pixels of similar color are grouped together. By then exploring all possible horizon lines in the image, first in a coarse manner using down-sampled images, and then the higher resolution image, the correct horizon line can be chosen by optimizing the minimization of the covariance of the two pixel distributions the line divides the image into [11].

3. Evaluation

This method could be carried over to satellites and horizon detection from space. Space pixels, or decidedly non-earth pixels are generally black, and earth pixels are generally not black, provided we exclude the case where the sun is present in imagery. However, the assumptions that the horizon presents itself as a straight line, and that the horizon presents itself in such a manner that the sky is in the top of the image do not hold for the horizon detection problem from space. In most imagery taken at low earth orbit (LEO), the earth is better described using a curved line, though in some cases at a smaller field of view, the horizon may at times present itself as a straight line. While the two-parameter equation of a straight line may be suitable for brute force search algorithms, the models examined here include parabolic and circular curves, increasing the number of parameters, making it impractical for this application.

D. FINDING THE HORIZON AS A CLASSIFICATION PROBLEM

Another method of examining the problem of finding the horizon in an image is to frame the problem as classifying pixels as either “earth pixels” or “sky pixels.” By then grouping the pixels together and examining where these two regions of pixels touch is the method of defining the horizon line. This section examines both supervised and unsupervised machine learning techniques as methods of classifying individual pixels in these two classes.

1. Features

For these techniques as presented by Grace Young et al. in [3] and S. Fefilatyev in [14], features were calculated based on the RGB characteristics of individual pixels as well as on small groupings of pixels.

In Young's work, several features of blocks of pixels were first identified as a coarse horizon detection technique, identifying the mean, standard deviation, and top down gradient values in each of the three color channels and in grayscale. In the fine detection (pixel level classification) these characteristics were calculated based on individual pixels vice sub blocks of the image [3]. In Fefilatyev's work, twenty one different features were calculated based on the three color channels of individual pixels, as well as from pixels in a 10x10 pixel block surrounding the pixel in question [14]. These features included those utilized by Young an included others such as calculated smoothness, entropy and uniformity values, though his features excluded features calculated solely based on grayscale values [14].

2. Unsupervised Learning Techniques

Young employed the unsupervised learning technique k-means clustering in over the ocean imagery for horizon detection. The algorithm first conducts coarse horizon detection by grouping sets of pixels into blocks and classifying these regions as either sky or non-sky by calculated features from the RGB color space in the block (e.g., mean, standard deviation, top-down gradient). Then the adjacent blocks that reflect the different classification are relabeled as horizon blocks to undergo post-processing and refined horizon detection [3].

The refined horizon detection is again clustered utilizing k-means clustering, this time utilizing the values of the individual pixels themselves vice the blocks they were divided into for the coarse detection. This provided good results as shown in Figure 3.

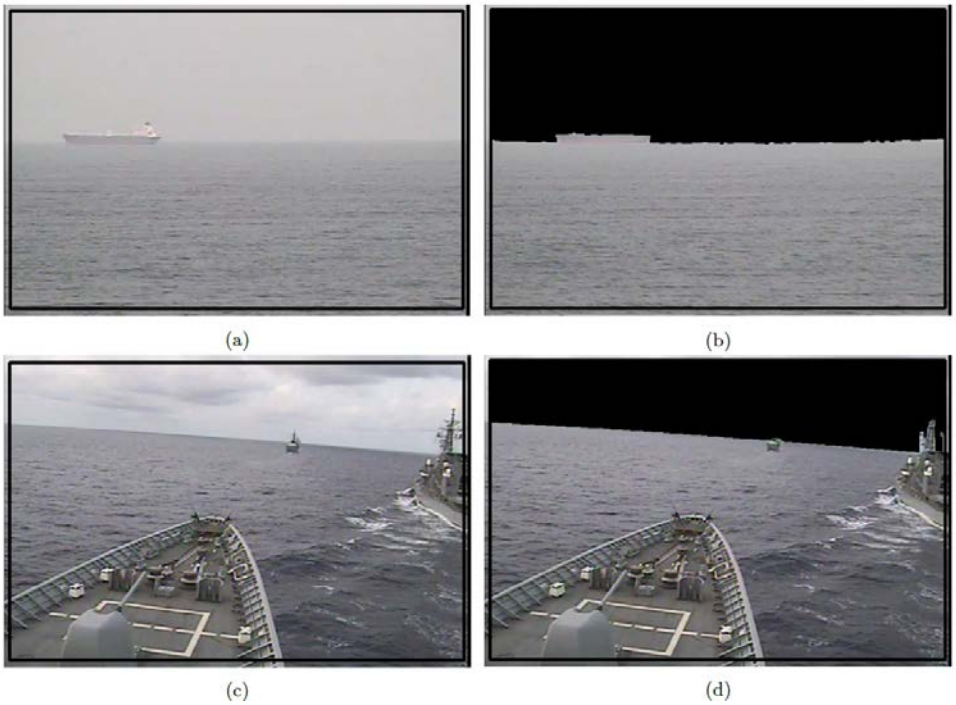


Figure 3. Horizon Detection with Unsupervised Learning. Source [3].

3. Supervised Learning Techniques

Fefilatyev applied supervised learning techniques to include support vector machines (SVM), decision trees, and naïve bayes classification algorithms in over the ocean imagery [14]. These algorithms were all examined using the twenty-one features calculated from the RGB color space and produced very accurate results, above 90% classification accuracy in classifying the individual pixels, which produced very accurate results in the horizon lines that were drawn from these. However, this was in the case that the images in the training and test sets were all very similar in lighting, time of day, which produced earth and sky pixels of roughly similar color in all the images. In the second experiment with more varied data set, classification accuracy dropped, thereby also decreasing the accuracy of the drawn horizon lines as shown in Figure 4.

Figure 4. Horizon Detection with Supervised Learning. Source [14].

4. Evaluation

In short, the RGB color space and grayscale pixels values in and image, and derived features calculated from surrounding pixels proved to be adequate features for common supervised and unsupervised learning techniques to be able to distinguish “sky pixels” from “earth pixels” in over the ocean imagery. With these successful techniques at classifying individual pixels, the horizon lines resulting from these techniques are generally adequate. However, in the supervised training techniques, there was a key importance that the training data be similar in lighting and color as the test data for the best results.

It is also the case in the application of these techniques, that the imagery was assumed to have a horizon within it. Therefore, the results contain no data on how these algorithms operate on imagery that does not contain a horizon. Young’s work also makes the assumption that the sky presents itself in the top half of the imagery, and that the horizon has a primarily horizontal presentation in the image. This can be seen in the fact that the top-down gradient in grayscale and color channels is calculated as a feature. Both of these assumptions cannot be made for imagery from space. It may be the case that the camera in space is pointed in a direction such that the horizon is not seen, or that the sensor is tilted, presenting the horizon on at a steep angle in the imagery.

This research borrows from these techniques in developing a method to solve two-class horizon detection problem as well as the multi-class horizon orientation classification problem. The presence of a horizon should be indicated adequately by the presence of space and earth pixels in space imagery. Since these previous projects have determined that the brightness and color information is adequate for determining these

classifications, these features were considered along with some of the supervised learning techniques considered by Fefilatyev in [14] as solutions to the two-class horizon detection problem and the multi-class horizon orientation classification problem

E. FINDING THE HORIZON AS A LINE DETECTION PROBLEM

Horizon line determination can also be looked at as a line detection problem; this is to identify a technique that locates those pixels that depict the transition from earth-pixels to sky-pixels based on changes in color or intensity. Several computer vision algorithms that have been developed for line detection might be applied to this domain.

1. Features

Line and edge detection techniques are generally implemented by detecting rapid changes in intensity in one of the RGB color channels, or most commonly in brightness (grayscale). In this way, the RGB or brightness values themselves with thresholding are able to find edges and lines, while other more advanced line detection techniques use gradient calculations of these RBG and grayscale channels as features for edge pixel selection.

2. Thresholding

Using grayscale imagery and assuming a black background, it can be fairly simple to find the edges of an object. Hanco Loubser in his nadir sensor edge detection algorithm followed a model that depicted pixels as being either in the background, the object (the earth) or the edge that would only ever be between two pixels in the other categories. The algorithm shown in his thesis does horizontal and vertical scanning along individual rows and columns of pixels, detecting when the brightness values cross calculated thresholds that define its classification [8]. There are some optimizations search patterns to increase the speed of algorithm, but this is an accurate depiction of the technique.

This technique works well if thresholds are determined properly. Thresholds for this algorithm are computed based on only having the sun and the earth in the imagery, and using the average intensities of both to determine threshold markers to sense which bright area is earth and which is the sun which produces good results for his purposes, but

also has a highly specific solution. This can be computationally intensive on high resolution images, though search patterns such as binary search (as demonstrated in his paper) can be implemented to decrease this effect.

3. Gradient-Based Edge Detection

Unlike thresholding, gradient-based edge detection methods typically rely on first and second order derivatives in order to sense changes in image intensity, vice a set threshold of brightness to determine foreground and background, signaling an edge.

The Sobel filter is a kernel that essentially computes a first order derivative of the values of a grayscale image when convolved with an image [15]. However, in selecting which pixels to call edge pixels, a threshold must again be utilized to select those gradient values which constitute an edge. A threshold too low would detect some non-edge pixels, and a threshold too high excludes some edge pixels. Implementation of this method inside the proposed problem of finding edge pixels in space would require fine tuning to determine the proper threshold.

The Canny edge detection algorithm applies a filter similar to the Sobel filter after smoothing the image, and employs an edge thinning technique as well as a self-determining thresholds in order filter weak edges from strong edges, with good results [15], [16]. These and other gradient-based edge detection methods rely on modeling the brightness of an image, with some smoothing to deter noise in the image, and finding the zero crossing of the second derivative as computed by convolving different kernels across an image [16].

Dusha et al. in another project for attitude determination for fixed wing aircraft implemented gradient-based edge detection for finding the horizon in terrestrial imagery. The algorithm separates the three color channels into distinct sub-images, then each image is smoothed, and edge detection is conducted on each of these sub-images with a Sobel filter, with thresholding. The research is key to note that the best edges occur in all three color channels, and to generate the final edge pixel selection all three sets of edge pixels are combined with a logical AND operation [17]. A Hough transform is applied to determine the best straight line from these edge pixel selections [17]. The results proved

to be adequate, except in the case where the strongest or most well defined line in the image was not the horizon as Figure 5(c) shows.

(a) Sun Glare (b) Building (c) Taxiway

Figure 5. Gradient-Based Horizon Detection. Source [17].

4. Evaluation

The gradient-based edge detection as implemented by Dusha et al. does provide satisfactory results when the horizon line is the most prominent line in the image. However, he also assumes the horizon presents itself as a straight line by resolving the line with the Hough transform. This is not a general assumption that can be made from space when the earth is generally observed to be a curved line. In this way the thresholding method demonstrated by Loubser is more generic and better for the horizon detection problem from space.

Also with regards to Dusha's work, it also may not be a good assumption that the horizon is the most prominent line in the imagery from space. The sun or the moon may be in the imagery and may provide a sharper more detectable line.

In summary, detecting a line based on the change of the intensity of the color and grayscale features of the image does produce edge pixels along the horizon, though with some with noise. The horizon brightness transition model borrows from this technique my modeling the change in intensity and using this feature to select pixels that depict the horizon based on this change. In this way it is similar to gradient-based edge detection.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

This chapter examines both the overall algorithm overview, and the methods that led to the development of each individual component of the algorithm.

A. ALGORITHM OVERVIEW

The core steps of the algorithm follow a basic outline, the first of which is image retrieval. For the purposes of this study, the algorithm operates under the assumption that the image collected by the camera is stored in a multi-dimensional array consisting of the traditional RGB color-space, organized by pixel coordinates in the image.

The next step is detecting whether there is a viable horizon in the image for horizon determination. This intends to prevent an image without a visible horizon to continue into the horizon determination algorithm, add a degree of robustness, and provide an opportunity for system resources to be spent on other methods of attitude determination. In addition this should quicken the processing cycle when this step determines that there is no detectable horizon in the image. This part of the algorithm also attempts to determine the coarse orientation of the horizon.

Once the algorithm detects a horizon in the image, and its coarse orientation, it can then set about determining which pixels in the image comprise the horizon. Early experiments with traditional line detection techniques revealed a lack of consistency that required deeper understanding of the visual appearance of the horizon and led to the development of the horizon brightness transition model to accomplish this task.

The final step is to fit the model of the shape of the horizon to the set of pixels determined to make up the horizon. Because of the nature of the problem, it is likely that the pixels selected by the horizon determination method may not entirely be connected across the image. Curve fitting allows for a more robust and consistent representation of the horizon across the image. The goal for this portion of the algorithm is to resist outliers from the detection process and effectively match the curvature of the earth as determined by the horizon brightness transition model.

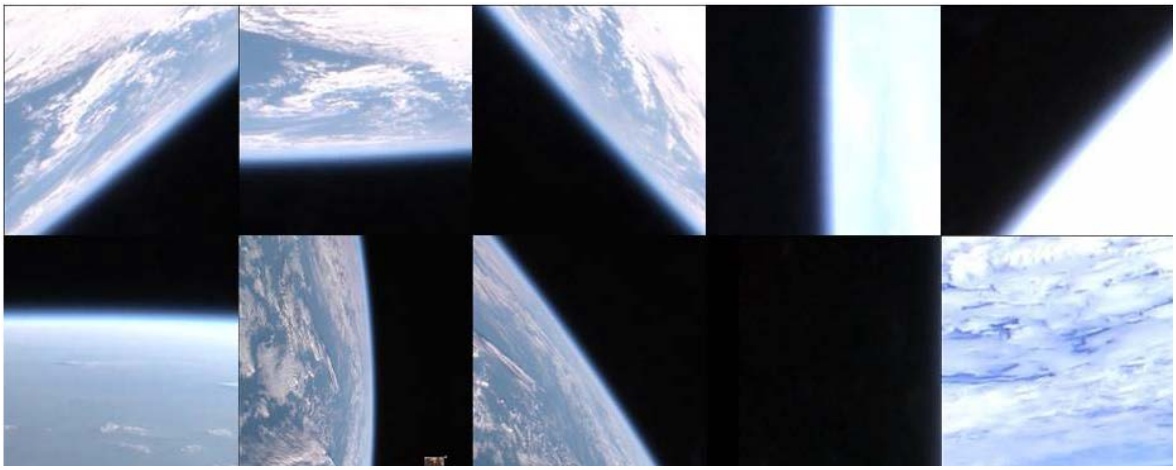
B. DETECTING THE HORIZON

While developing the horizon brightness transition model, certain conditions were found for which it was not feasible to detect the horizon. This portion of the algorithm was developed to keep images that do not contain an adequate horizon from being input into the horizon determination portion of the algorithm. In addition, the methods were also checked for their ability to identify horizon orientations to speed up the application of the HBTM in the horizon determination algorithm.

1. Image Orientation Classifications

With reference to the goal of determining the coarse orientation of the horizon as part of the detection process, the following classifications of images were examined in addition to merely detecting the presence of the horizon as a two-class problem. Figure 6 provides a sample visualization of these classifications.

- NH, or No Horizon, for images that do not have a detectable horizon.
- UM, or Upper Middle, for images that have a detectable horizon where the earth appears to be centered in the top portion of the image.
- UR, or Upper Right, for images that have a detectable horizon where the earth appears to be oriented to the top right corner of the image.
- UL, or Upper Left, for images that have a detectable horizon where the earth appears to be oriented to the top left corner of the image.
- ML, or Middle Left, for images that have a detectable horizon where the earth appears to be oriented to the middle of the left side of the image.
- MR, or Middle Right, for images that have a detectable horizon where the earth appears to be oriented to the middle of the right side of the image.
- LL, or Lower Left, for images that have a detectable horizon where the earth appears to be oriented to the middle of the right side of the image.
- LM, or Lower Middle for images that have a detectable horizon where the earth appears to be oriented centered in the lower portion of the image.
- LR, or Lower Right for images that have a detectable horizon where the earth appears to be oriented to the lower right corner of the image.



From left to right, top to bottom, expected horizon profiles that classify as follows: upper left, upper middle, upper right, middle right, lower right, lower middle, middle left, lower left, no horizon (mostly space or night side earth, majority dim), and no horizon (mostly earth or sunlight, image washed out in brightness.)

Figure 6. Horizon Classification Orientations

2. Feature Extraction

This first step of the algorithm involves a bit of preprocessing in order to uniformly be able to evaluate all images in a consistent manner. Firstly, an OpenCV function is used to convert the image to grayscale, and therefore each pixel is reduced to one value vice the three values of RGB color-space. Then, images are resized to 64 x 64 pixels to ensure size invariance to the feature extraction process. However, this does not preserve aspect ratio, and therefore some characteristics of shape and relative scale may be lost in this transformation. Several feature sets are extracted and evaluated for their ability to inform a classifier, the first being simply the brightness values reshaped into a one dimensional list of features.

A derived feature that was examined is converting the resized image into a two-dimensional feature, a weighted center of mass. To do this, the pixel coordinates are transformed into a vector centered at the image center, and then scaled by its normalized brightness value. This is completed for all pixels, and each vector is summed together to compose the final two feature representation of the image. Equation (1) shows how this feature is calculated, with p_x and p_y representing pixel coordinates in the x and y axes

respectively, and p_b representing the brightness of pixel p. This two-dimensional feature was intended to give a good sense of where the brightest portions of the image are.

$$V = \sum_{p} ([-32 + p_x) \times \frac{p_b}{255}, (32 - p_y) \times \frac{p_b}{255}]) \quad (1)$$

The last feature set was a five-dimensional feature composed of the individual components of the derived vector in Equation 1. This feature was designed specifically with the SVM classifier in mind, to allow it to learn curved boundaries in this two-dimensional by presenting it with a non-linear kernel of the two-dimensional feature. Equation 2 shows this feature vector as generated from the derived two-dimensional features.

$$V_5 = [V_x, V_y, V_x^2, V_y^2, V_x V_y] \quad (2)$$

3. Classification by K-Nearest Neighbors

The K-Nearest Neighbors approach essentially relies on the fact that images in the same class should have similar features. For this implementation of KNN, the Euclidean distance between the calculated features is the distance function. The experiments described in Chapter IV show how the algorithm was optimized by determining the best value for *k*.

4. Classification by Support Vector Machine

Another useful multi-class classifier is the support vector machine, which learns the boundary between the various classes. Training this model over the instances is much more involved mathematically. The SVM model is a set of weights, or values that coincide with specific features within the numerical feature set. This makes the SVM a much more complex model for determining horizon classification, except that once the mathematical equation is learned, only those weights need to be stored for future testing, as opposed to retaining all of the training data as the KNN algorithm does.

Another important note about the SVM is that the implementation chosen implements only the linear kernel for the given feature sets. In the case of the 4096-

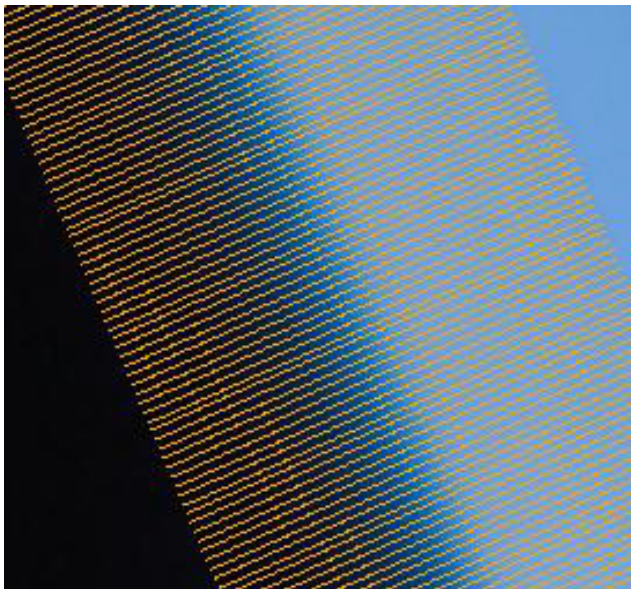
dimensional feature vector, the SVM attempts to learn the hyperplane that best separates different orientation classifications. Early experiments in plotting the two-dimensional features show that a curved line may best describe the boundary between the classifiers in the two-dimensional space. Therefore, the five-dimensional feature vector was derived specifically as a second degree kernel of the two-dimensional feature set to allow the SVM to learn this curved boundary.

C. MODELING THE HORIZON BRIGHTNESS TRANSITION

To model the horizon, it was necessary to sample the pixels in close proximity to and overlapping the horizon. First the images were marked manually with a few points within close proximity to where the visual horizon was perceived by a human being. Then the best fit curve along the horizon was determined, using a developed least squares curve fitting function. After identifying this curve and a set of points along the curve within the image, sampling lines perpendicular to this curve were established. This yielded a collection of values that demonstrated the changes in brightness over the horizon line. Figure 7 shows the sampling lines developed for one particular horizon. In order to collect a generic representation of the brightness transition, the sample lines were averaged to create one sample vector per image. Equations 3 and 4 show the calculation to compute the final sample vector *SV*, where S_{nm} is the *m*th value of the *n*th sample line, and the value of *m* increases from zero to the sample vector length, from the earth side of the horizon, to the space side of the horizon. *SV* therefore contains the averages of the brightness values SV_{ib} as well as a representation of the distance of that sample along the sampling line SV_{ix}. Figure 8 shows a plot of one sample vector from an image before normalization.

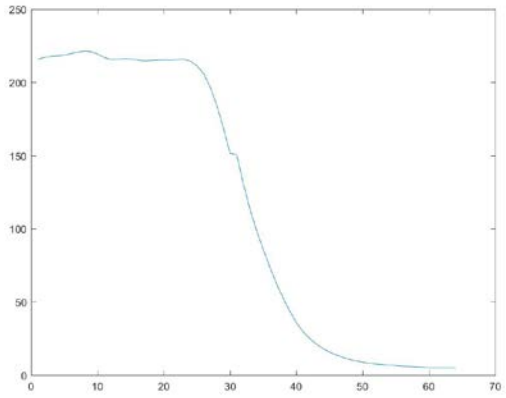
$$SV_{ib} = \frac{\sum_{j=1}^{n} S_{ji}}{n} \tag{3}$$

$$SV_{ix} = \frac{i}{m} \tag{4}$$



A detailed view of the sampling lines highlighting how the sampling lines are designed to run perpendicular to the horizon.

Figure 7. View of Sampled Pixels in a Horizon Containing Image.



The connected plot of one sample vector (SV) representing the brightness transition of one image, brightness (SV_{ib}) on the y axis, and the sample index(SV_{ix}) on the x axis, prior to normalization.

Figure 8. Sample Vector Plot

Initial experiments in just plotting the brightness transitions showed a steady brightness level, near zero, for the space portions of the image, and a wavering but somewhat consistent brightness in parts of the image where the earth was present. The brightness transition model then must model the transition between the high and low

values of brightness. This was influential in selecting equations 5, 6 and 7 as models for testing against this data where *B* is the brightness function for x given parameters *P*, and x is a value from zero to one, zero reflecting clearly an earth pixel with high brightness, and one reflecting a clearly space pixel with very low brightness thus mirroring the method used to calculate SV_{ix}.

Linear

$$B(\mathrm{x}, \mathrm{P}) = \mathrm{P}_0 + \mathrm{P}_1 \mathrm{x} \tag{5}$$

Third Degree Polynomial

$$B(x, P) = P_0 + P_1 x + P_2 x^2 + P_3 x^3 \tag{6}$$

Inverse Tangent

$$B(x, P) = P_0 + P_1 \arctan(\mathrm{P}_2(\theta - P_3)) \tag{7}$$

Each model was evaluated against the sampled data utilizing a least squares method approach for fitting the parameters *P* to the data to provide the best possible fit.

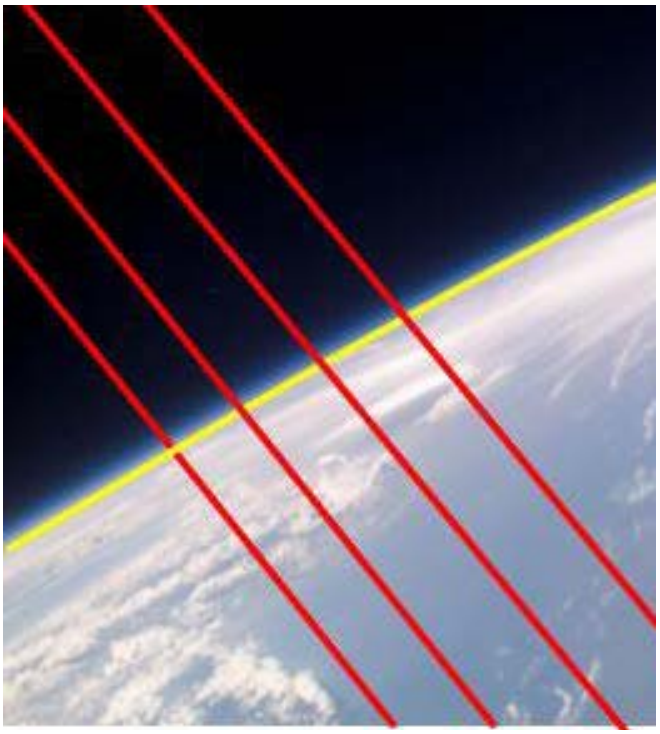
Once the best model was determined, it was optimized with an expectation maximization algorithm to align the data in such a way that the brightness values appear most likely to come from the model. This involved only shifting the data along the axes to best align with the model, retaining all noise and unique features, but in such a way that the data aligned as if it had come from the same horizon point. Then the brightness function was refit using the least squares fitting method for the parameters of the fit to best represent the curve defining the data.

Equation 8 shows the likelihood model, where it is based on a normal distribution centered on the models predicted brightness. My minimizing this function, the maximum likelihood estimation of the parameters was solved for.

$$\Pr(SV) = \sum_{i=1}^{n} -\ln(\Pr(\mathrm{S}_{\mathrm{ib}} \mid \mathrm{N}(\mathrm{B}(\mathrm{SV}_{\mathrm{ix}}, \mathrm{P}), \sigma_{\mathrm{B}}) \tag{8}$$

D. DETERMINING HORIZON PIXELS

By following the same methodology for determining the model in reverse, the model can be applied to a set of samples from an image in order to determine pixels that best represent the position of the horizon. Given orientation information derived from the classification portion of the algorithm, an efficient sampling method can be employed as shown in Figure 9.



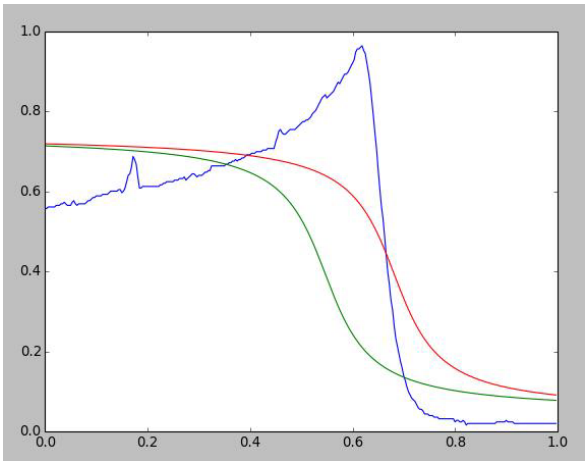
Generic sampling lines that would not' be perfectly perpendicular to the horizon, but approximately so due to the discrete orientation classification step.

Figure 9. Horizon Determination Sampling

With these pixel sampling lines it was then possible to again utilize maximum likelihood estimation of the horizon shift to fit the model to the data. Equation 9 shows the adjusted brightness function to include that shift variable for the arctangent model. Once the model is fit to the data, the point on the curve that most closely reflects the horizon was identified by applying that shift to the base model's horizon point. By again maximizing *Pr(S)* using the parameters *P* found during the brightness models

optimization, and finding the most likely value for *shift*, which is the shift between the models selected horizon point, P_3, and the horizon point reflected by the data. Figure 10 illustrates this graphically. It should be noted that P_3 in this instance is the maximization of the first derivative of *B(x)* for the arctangent model, chosen specifically for this study, but can be changed in order to fine tune the horizon brightness transition models pixel selections.

$$B(x, P, shift) = P_0 + P_1 \arctan(P_2(\theta - P_3 - shift)) \tag{9}$$



The horizon brightness transition model in green, samples *SV* in blue, and the model realigned by *shift* in red, with normalized brightness on the y-axis and normalized sample index on the x axis.

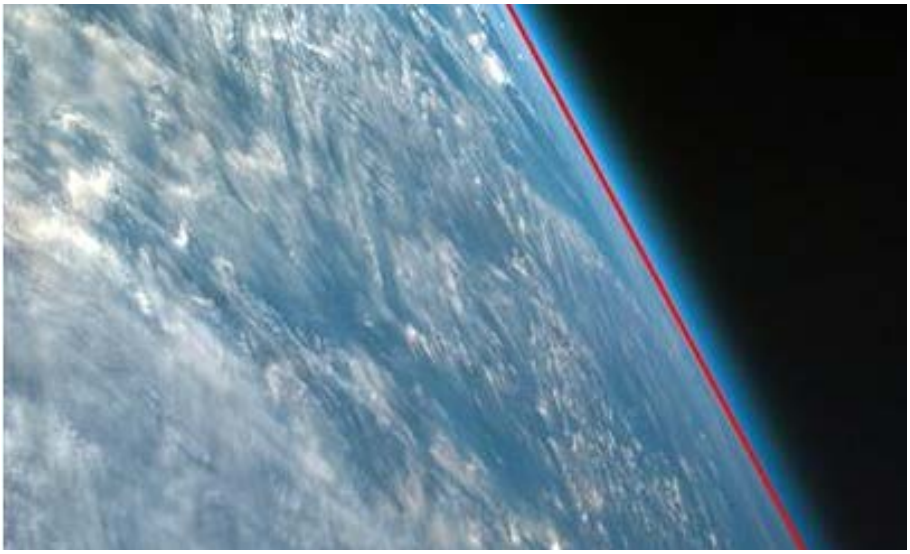
Figure 10. Determining the Horizon Pixel via Maximum Likelihood Estimates

E. FITTING A HORIZON CURVE

In the sampling method, a circular model of the curvature of the earth was implemented in order to produce sample pixel values. However, looking at Figure 11 reveals that even in this early experiment the circular model is sometimes inaccurate and that the best model to adequately describe the curvature of the earth deserves further examination.

It is also fair to assume, based on earlier experiments in horizon detection, that the pixel detection method may in fact incorrectly designate some pixel locations as horizon pixels. Therefore, this curve fitting method must be robust to outliers. This algorithm for

curve fitting implements a random sampling consensus (RANSAC) algorithm, combined with linear, parabolic, and circular curve models. This allows for both the hyper parameter fitting of the RANSAC outlier threshold to best fit the algorithm, and an evaluation the of the sum of the squared errors of each model to determine which model truly is the best fit to the data.



The pictured horizon was not able to best fit to a circle. It presents more like a line, and therefore the RANSAC algorithm could not best fit a circle.

Figure 11. Horizon Presenting as a Straight Line vice a Curve

IV. EXPERIMENTS

The conducted experiments were based on the previously described methodology. They were designed to test different parameters for these models in order to determine an effective solution that achieves high rates of success for the total horizon detection and determination algorithm.

A. DATA SET

The images were collected from a number of data sources, all of which were at least verified to be actual imagery from space. Most the images are from the National Aeronautics and Space Administration (NASA) either via their home website or released imagery, other news or academic articles based on NASA released imagery, or via live feed from the International Space Station (ISS), supported by NASA at www.urthecast.com. Images were also collected from the Naval Postgraduate School's Space Systems Academic Group high altitude balloon launches from 2013 to 2015 mainly to extend the set of images that did not contain a visible horizon.

The data was divided into two sets. One set of 30 images of varying lighting, time of day, sun angle, and other variants was chosen for building the horizon brightness transition model.

The second set consists of 50 original high-resolution images which were randomly cropped and resized to produce a diverse set of 251 images containing images of solely earth, solely space and stars, and a portion containing horizons of varying orientations and proportions. These 251 images were then rotated in the cardinal directions to provide further variance for building the models, yielding a training set size of 1004 images.

B. IMAGE CLASSIFICATION

1. Feature Extraction

For these experiments, a script was written in the Python computer programming language to resize original images to 64 x 64 pixel images utilizing OpenCV's *resize* function. From this smaller image, the features extracted include, a 4096 dimension

feature that includes all pixel brightness values and the two and five-dimensional features derived in Chapter III, Section B.

2. Implementation

For the experiments, the horizon detection script was implemented with the Sci-Kit Learn Python package. For the KNN algorithm the *KNeighborsClassifier* class was chosen. This allows the software to input the desired number of neighbors to be computed over the data. The internal structure of the class itself handles determining the distance between each element of the data by simply computing the Euclidean distance between each element in the feature space provided.

For the SVM implementation, the *LinearSVC* class from the Sci-Kit Learn Python package was selected. It is also important to note that this implementation of an SVM does not automatically implement non-linear kernels over the provided features. This class requires several inputs aside from the data itself, including a loss function, either the hinge loss function, or the squared hinge loss function. Both were examined in the experiments for comparison. Other parameters include the tolerance variable, which is the amount of gradient descent that determines whether the SVM solution has sufficiently learned the appropriate classification boundaries from the data provided.

3. Experiment 1: KNN Horizon Detection

This experiment was designed with the sole purpose of determining the precision and recall tradeoff for the KNN algorithm for the two-class problem of detecting a visible horizon. For this experiment, each image was labeled based on whether there is a clearly visible horizon line in the image. These binary labels were fed into the algorithm. Then, for iterating values of k, from one to eleven, a ten-fold cross validation over the data set was conducted, and the average precision and recall computed.

4. Experiment 2: KNN Horizon Categorization

This experiment is designed to determine tradeoffs in utilizing the KNN algorithm for the nine-class problem of horizon detection and orientation categorization. This experiment runs exactly the same as Experiment 1, conducting a ten-fold cross validation,

except that instead of calculating average classification accuracy from the each fold, the classification accuracy is computed from the entire result set for the 1004 images.

5. Experiment 3: SVM Horizon Detection

Because of the way this algorithm is implemented, it is important to conduct this experiment multiple times to ensure the tolerance variable is such that the classifier converges to produce the best and most consistent results across the data. The default value given by the LinearSVC class is 0.0001. This experiment examined scaling the tolerance up and down by a factor of ten. Ten trials were conducted with each value for gradient movement tolerance. All trials consist of a ten-fold cross-validation over the dataset, and were done with both the hinge and squared hinge loss functions for comparison.

6. Experiment 4: SVM Horizon Categorization

This experiment was designed to examine the SVM conducting the nine-class horizon orientation classification, and was to be run similarly to Experiment 5, iterating over the different tolerance values and loss functions. The results were tabulated in the same manner as Experiment 2, in that the classification accuracy was computed based on the results for the entire data set instead of the averages of the individual folds.

7. Experiment 5: Categorization without Horizon Detection

This experiment was designed to examine the horizon orientation classification of images that have already been verified to have a visible horizon, changing the nine-class classification problem to an eight-class classification problem. This determines whether screening for the presence of a horizon prior to attempting orientation classification provides better orientation classification accuracy. It was run similarly to Experiments 2 and 4, aggregating results through a ten-fold cross validation.

C. ESTIMATING HORIZON MODEL PARAMETERS

For this experiment, a MATLAB script was written, with two inputs—first the image file itself, and second a text file with several pixel locations that were manually

identified as being horizon pixels. The script then samples the image based on the methods described in Chapter III to produce the sample vector $SV = <SVix, SV_{ib}>$.

For each image, the sample vector values were then normalized to hold values from zero to one, zero being the SV_{ix} value for those pixels clearly earth pixels, and one being the SV_{ix} value for those pixels clearly space or non-earth. The brightness values are then also normalized to hold values between zero and one.

The sample vectors for all images were then combined into one vector of data, and several models, linear, partial linear, third degree polynomial, and arc-tangent were fitted to the data in this vector utilizing a least squares fitting method. After finding the most accurate model the data was re-aligned in both axes based on a maximum likelihood that the data had come from the constructed model. With this, the least squares fitting algorithm was run again for optimizing the selected model.

D. FITTING THE HORIZON CURVE

This experiment examined the type of curve that best fit the horizon data points. In the absence of ground truth, the data was generated by the horizon brightness transition model and a RANSAC curve fitting algorithm was run with several types of curves, linear, parabolic, and circular, in an attempt to find the curve that best matches that of the horizon. The RANSAC algorithm was written to take the coordinates of the horizon pixels as input, and output the highest consensus set percentage with the given error tolerance, as well as the sum of the squares of the errors. In this way, the results can be evaluated by determining which model achieved the highest consensus set, and which model produces the least error by comparing the sum of the squares of the errors.

This allows for visual evaluation of the images in the dataset to determine if the resulting horizon curve is near the manually perceived horizon, or is not within a reasonable distance of the perceived horizon. In the absence of ground truth, this evaluation is very subjective to the view of the author, though the best attempt was made to follow a criterion of whether the determined horizon line is within one tenth of the images height or width, whichever is smaller, from the perceived visual horizon, and the orientation had to appear to be very close to accurate.

V. RESULTS AND DISCUSSION

A. DETECTING THE HORIZON & ORIENTATIONS

As discussed in Chapter I and the formulation of the problem, this section provides the results of the experiments described in Chapter IV.

1. Experiment 1

The purpose of this experiment was to find a suitable value of *k* that provides for a high level of precision and recall. The tabulated results are presented in the Table 1.

Table 1. KNN Horizon Detection

Two-dimensional Features		
Value of K	**Average Precision**	**Average Recall**
1	84.7%	82.6%
3	86.3%	83.5%
5	87.9%	82.4%
7	91.9%	82.9%
9	92.0%	82.0%
11	92.2%	80.9%
Five-dimensional Features		
Value of K	**Average Precision**	**Average Recall**
1	84.9%	84.1%
3	84.6%	82.4%
5	85.4%	81.4%
7	88.0%	82.4%
9	89.1%	82.2%
11	89.5%	81.4%
All Pixel Values Features		
Value of K	**Average Precision**	**Average Recall**
***1**	96.3%	91.6%
3	97.6%	85.6%
5	97.2%	81.1%
7	97.8%	78.3%
9	97.5%	76.5%
11	97.7%	76.0%

Here it is observed that for horizon detection purposes, the KNN classifier performed best when given all the pixel brightness values as the features; these precisions greatly outscoring the two and five-dimensional features. However, on average, the KNN classifier had a more consistent recall when using the two and five-dimensional feature

spaces. Still, utilizing all the pixels as the features, and a single nearest neighbor approach (k=1), provides the highest precision and recall for the two-class horizon detection in imagery, resulting in the most horizon images to be processed and ensuring only a few of the no horizon images are processed.

2. Experiment 2

Table 2 shows all the datasets vs. *k* for the nine-class classification problem. It is important to note that the categorization problem here includes the *no horizon* (NH) classification. Therefore, the algorithm was essentially solving the horizon detection problem combined with coarse horizon orientation classification.

Table 2. KNN Horizon Detection and Categorization

Two-dimensional Features

Value of K	Classification Accuracy	NH False Negative Rate	NH False Positive Rate
1	65.5%	22.0%	23.8%
3	69.8%	18.8%	23.9%
5	74.6%	8.3%	24.7%
7	76.2%	4.8%	24.7%
9	76.2%	3.5%	25.2%
11	75.6%	3.0%	26.8%

Five-dimensional Features

Value of K	Classification Accuracy	NH False Negative Rate	NH False Positive Rate
1	50.0%	22.4	23.5%
3	51.8%	21.4	25.4%
5	54.6%	8.8%	28.2%
7	54.9%	6.8%	27.8%
9	54.9%	5.0%	27.8%
11	56%	4.0%	27.7%

All Pixel Values Features

Value of K	Classification Accuracy	NH False Negative Rate	NH False Positive Rate
1	76.7%	4.8%	13.5%
3	76.2%	2.3%	20.0%
5	75.5%	2.0%	24.7%
7	75.6%	1.3%	27.8%
9	73.8%	1.8%	30.2%
11	72.8%	1.3%	31.4%

No horizon class abbreviated as NH.

Table 2 shows the classification accuracy for each set of features for each value of k. It is important to evaluate this data not only based on the accuracy of classification, but also by the particular data associated with the NH, or *no horizon* classification. In Table 2 there are poor results in general that improve as the value of k increases. There is also a fairly consistent false positive rate for NH classification, which means approximately one in four horizon images is being classified as having no horizon. This is worse than using KNN solely for detection purposes, and shows that utilizing the KNN algorithm this multi-class classification problem does not actually add valuable information to the system.

3. Experiment 3

The results for Experiment 3, demonstrating classification with an SVM to detect visible horizons are listed in Tables 3, 4, 5, and 6, listed by the feature set and loss function used in the algorithm

Table 3. SVM Horizon Detection, tol=0.000001

Features	Loss Function	Average Precision	Average Recall
2 Dimension	Hinge	78.9%	30.5%
2 Dimension	Squared Hinge	85.2%	29.1%
5 Dimension	Hinge	78.0%	65.8%
5 Dimension	Squared Hinge	66.0%	45.5%
All Pixel	Hinge	58.1%	47.5%
All Pixels	Squared Hinge	63.0%	44.8%

Table 4. SVM Horizon Detection, tol=.00001

Features	Loss Function	Average Precision	Average Recall
2 Dimension	Hinge	66.9%	23.9%
2 Dimension	Squared Hinge	88.0%	32.1%
5 Dimension	Hinge	78.8%	50.0%
5 Dimension	Squared Hinge	71.7%	62.9%
All Pixel	Hinge	58.2%	51.4%
All Pixels	Squared Hinge	59.5%	45.8%

Table 5. SVM Horizon Detection, tol =.0001

Features	Loss Function	Average Precision	Average Recall
2 Dimension	Hinge	77.3%	33.7%
2 Dimension	Squared Hinge	85.7%	27.8%
5 Dimension	Hinge	83.1%	53.2%
5 Dimension	Squared Hinge	64.5%	43.9%
All Pixel	Hinge	59.0%	47.8%
All Pixels	Squared Hinge	59.8%	48.8%

Table 6. SVM Horizon Detection, tol=.001

Features	Loss Function	Average Precision	Average Recall
2 Dimension	Hinge	70.9%	26.1%
2 Dimension	Squared Hinge	88.4%	31.8%
5 Dimension	Hinge	86.1%	64.4%
5 Dimension	Squared Hinge	52.5%	37.5%
All Pixel	Hinge	58.7%	43.65%
All Pixels	Squared Hinge	58.5%	48.62%

The data shows poor performance by the SVM in the two-class horizon detection problem. This poor performance would seem to indicate that the features and classifier are mismatched. SVM detection of the visual horizon results in roughly 30–50% of the viable horizon images being discarded, as well as a larger number of non-horizon images being put through the horizon determination process needlessly. In the worst case, the recall of this technique is on par with guessing a two-class scenario, a 50% chance of correctness.

4. Experiment 4

Table 4 shows the results for this experiment, employing the SVM algorithm in nine-class problem of horizon detection and orientation classification. Tables 7, 8, 9, and 10show the classification accuracies as well as some detailed data on the *no horizon* classifications

.

Table 7. SVM Horizon Detection and Categorization, tol=.000001

Features	Loss Function	Classification Accuracy	NH False Positive Rate	NH False Negative Rate
2 Dimension	Hinge	39.6%	8.5%	50.8%
2 Dimension	Squared Hinge	38.2%	11.3%	47.2%
5 Dimension	Hinge	38.8%	15.5%	35.0%
5 Dimension	Squared Hinge	32.5%	50.0%	50.0%
All Pixel	Hinge	46.2%	28.5%	48.3%
All Pixels	Squared Hinge	46.2%	28.0%	42.5%

Table 8. SVM Horizon Detection and Categorization, tol=.00001

Features	Loss Function	Classification Accuracy	NH False Positive Rate	NH False Negative Rate
2 Dimension	Hinge	47.2%	4.5%	41.9%
2 Dimension	Squared Hinge	40.3%	5.3%	49.3%
5 Dimension	Hinge	40.1%	14.0%	44.2%
5 Dimension	Squared Hinge	29.2%	68.0%	36.3%
All Pixel	Hinge	41.4%	47.3%	49.9%
All Pixel	Squared Hinge	48.8%	25.3%	41.8%

Table 9. SVM Horizon Detection and Categorization, tol=.0001

Features	Loss Function	Classification Accuracy	NH False Positive Rate	NH False Negative Rate
2 Dimension	Hinge	43.9	8.8	47.8
2 Dimension	Squared Hinge	43.4	7.8	49.1
5 Dimension	Hinge	39.3	17.5	38.0
5 Dimension	Squared Hinge	32.57	58.0	47.5
All Pixel	Hinge	46.4	33.3	47.85
All Pixels	Squared Hinge	46.8	28.5	41.5

Table 10. SVM Horizon Detection and Categorization, tol=.001

Features	Loss Function	Classification Accuracy	NH False Positive Rate	NH False Negative Rate
2 Dimension	Hinge	42.6	6.8	46.8
2 Dimension	Squared Hinge	41.6	9.8	49.4
5 Dimension	Hinge	34.3	31.0	30.0
5 Dimension	Squared Hinge	30.9	61.8	44.4
All Pixel	Hinge	43.9	42.3	47.5
All Pixels	Squared Hinge	48.7	26.8	39.3

As expected, based on the previous results for the two-class horizon detection problem, the data clearly shows that the SVM does a poor job at classifying the images

by coarse orientation. The SVM as a candidate for both the detection and classification of a horizon performs well below a 50% accuracy threshold, and the NH false positive rates show that in the best case roughly one out of twenty non-horizon images were selected for further examination where they should not be. This is only slightly better than guessing the nine-class problem, which would provide an 11% chance of providing the correct classification.

5. Experiment 5

This experiment was meant to be a reconstruction of Experiments 2 and 4, but with data that excludes the *no horizon* classification, specifically to check if these classifiers produce better accuracy on an image that has already been detected to have a horizon, making this an eight-class classification problem. Table 11 and Table 12 show the results. For brevity, only the highest performing options of the SVM results are shown.

Table 11. KNN Classification Without NH Class

Two-dimensional Features	
Value of K	**Classification Accuracy**
1	61.2
3	66.2
5	69.0
7	68.7
9	69.0
11	66.7
Five-dimensional Features	
Value of K	**Classification Accuracy**
1	35.0
3	35.8
5	33.3
7	31.2
9	31.2
11	31.2
All Pixel Values Features	
Value of K	**Classification Accuracy**
1	67.3
3	66.8
5	68.0
7	67.8
9	66.5
11	64.2

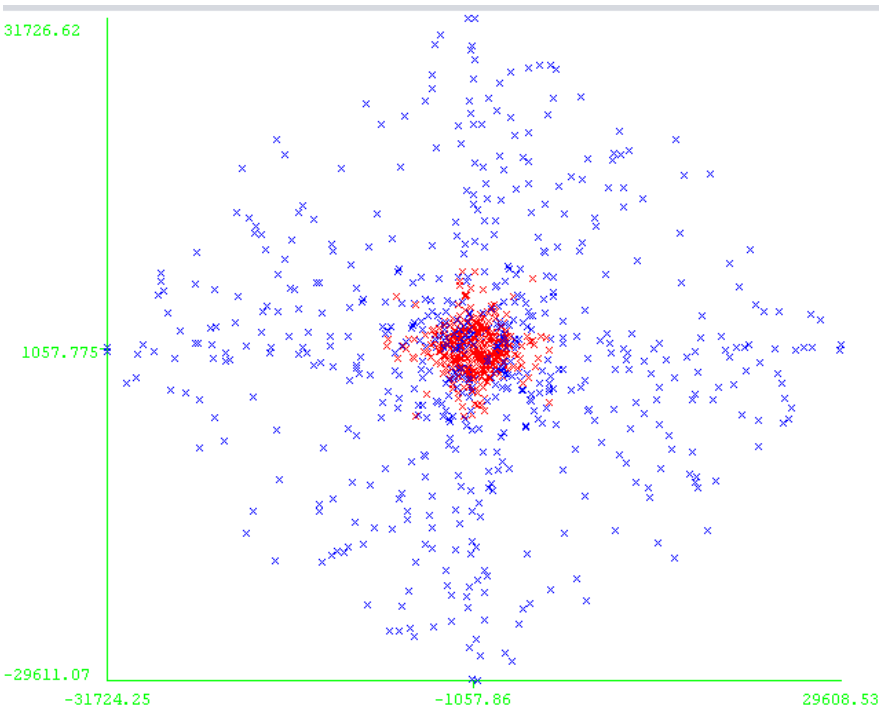
Table 12. SVM Classification without NH Class, All Pixel Values Features

Tolerance	Loss Function	Classification Accuracy
.001	Hinge	48.3
.001	Squared Hinge	46.8
.0001	Hinge	47.2
.0001	Squared Hinge	46.8
.00000l	Hinge	48.7
.000001	Squared Hinge	44.0

Unfortunately, the results show little to no improvement over the nine-class classification problem. It is clear here that the features identified do no clearly separate the data in such a way that the KNN and SVM classifiers can learn to differentiate them.

6. Technique Selection

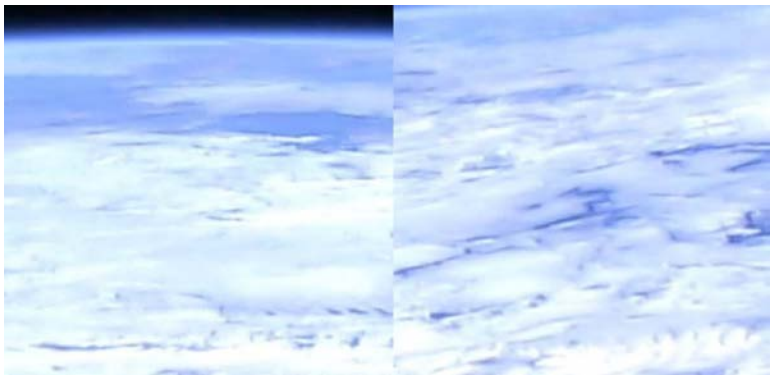
By far, the KNN approach for horizon detection performed the best. This makes it the horizon detector of choice. The SVM achieved both a lower precision and a lower recall from the KNN methods with all pixel values as the features. While it is impossible to visualize the 4096 dimensional space, the two-dimensional features are shown in Figure 12, with the two colors representing the different classes.



The images and their classification displayed in the derived two-dimensional feature space. The red data points indicate the no-horizon classification, whereas the blue data points are the horizon class of images.

Figure 12. Horizon and No Horizon Images in the 2D Feature Space

From Figure 12 we can hypothesize that the performance of the SVM was poor in the two and five-dimensional feature space because the two-classes are not clearly separable. The five-dimensional feature space was specifically derived from the two-dimensional feature space and designed to allow the SVM to learn a curve that might best separate the images, but this generates no improvement. Further examination of how and why this is the case can be seen in Figure 13, where the image on the left contains a visible horizon, and the image on the right is a no horizon image.



An image with a horizon is shown on the left, and an image with no horizon is shown on the right.

Figure 13. Difficulties with Horizon Detection

The data set contains multiple images similar to the instance shown in Figure 13, where the images are similar in brightness, and both nearly cover the entire image. In these cases both the derived features and the large feature vector containing all pixel values are similar despite being from different classes. The KNN algorithm provided with the 4096 dimensional features is able to differentiate between the two images with this kind of similarity in most cases.

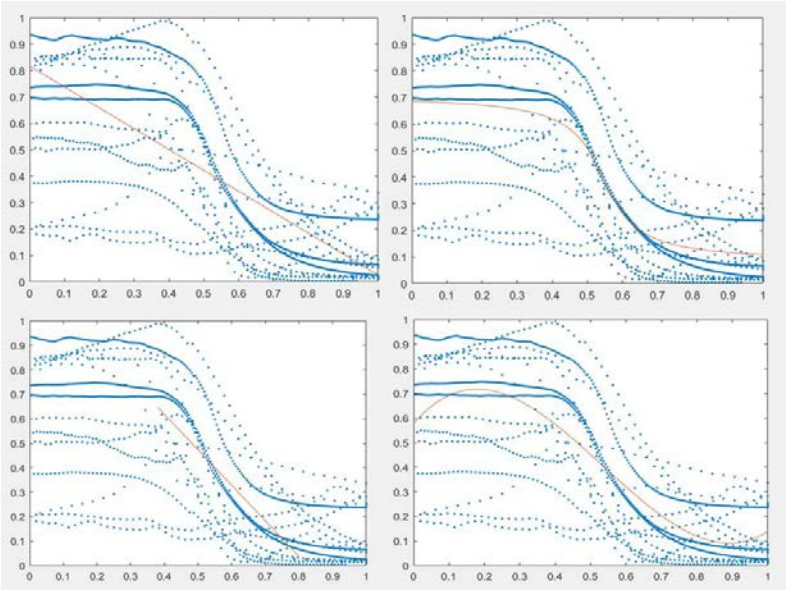
As for the eight and nine-class horizon orientation classification problem, neither the KNN nor SVM techniques proved to be significantly better than guessing. Either the features identified in this thesis do not provide enough information for these classifiers to differentiate between the classes, or another technique should be applied to the selected features for a better outcome.

B. HORIZON BRIGHTNESS TRANSITION MODEL

1. Parameter Estimation

Figure 14 shows the initial curve fitting for the various models fit to the data in an attempt to model the horizon brightness transition. The linear fit was clearly unable to completely model the full horizon transition from end to end, but with a marginally better mean squared error when only modeling the actual transition area (partial linear model). Likewise the third-degree polynomial does not match the transition in the brightness along the entire scanline. The arctangent curve, however, models the gradual decrease in

the steepness of the transition and the as well as the parts of the model that map to the earth and dark space.



Horizon Brightness Transition Model candidates plotted, with brightness on the y axis, and horizon placement on the x axis (zero being bright earth, and a value one indicating clearly non-earth). From top left clockwise, the linear, arctangent, third-degree polynomial, and partial linear models are shown. Blue dots are the individual sample points, the orange lines being the model.

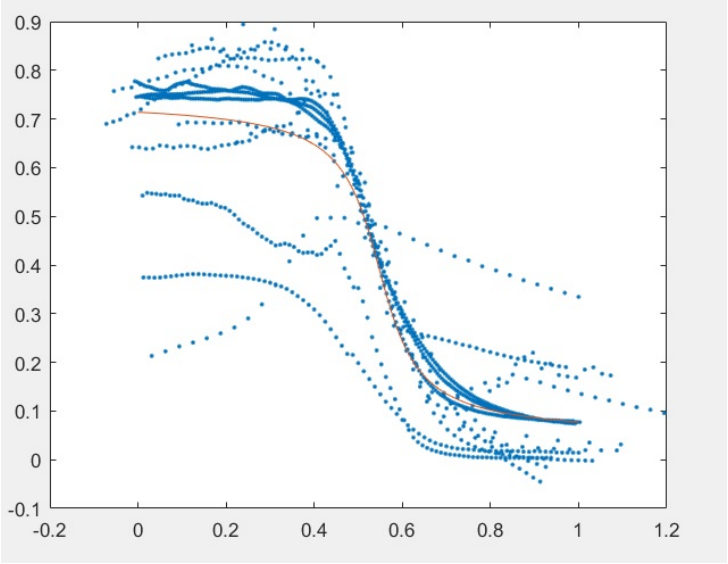
Figure 14. Horizon Brightness Transition Model Fitting

Table 13 shows the empirical data based on nonlinear regression fitting of each model. The partial linear model has the lowest mean squared error, but also models only the transition. The next lowest error is the arctangent model, which also boasts the highest coefficient of determination (R^2) value, indicating it has the most predictive power of all the models tested.

Table 13. Horizon Transition Curve Model Evaluation Pre-Optimization

Model	**R^2**	**Mean Squared Error**
Linear	.5726	.0387
Partial Linear	.5057	.0303
3rd Degree Polynomial	.6257	.0339
Arctangent	.6403	.0326

The final model resulted from optimization of the parameters after using a maximum likelihood estimation to shift the data so that the horizons align as if they came from the model. The images that show horizons that do not align with this model are also removed, acknowledging that his model loses the ability to detect the horizon in extremely dark images where the brightness of the earth is significantly reduced. Figure 15 shows the re-plotting of the data and the revised model after applying a horizontal and vertical shift that gives the highest likelihood for the data coming from the model.



Final horizon brightness transition model alongside collected data, brightness on the y-axis, and displacement from the horizon on the x-axis (zero being earth, and one or greater clearly non-earth pixels). Blue dots are the data, orange curve is the refined model.

Figure 15. Refined Model with MLE Fit Data

The final model and parameters are listed, and give a mean squared error of .0124 and an coefficient of determination 0.8531, showing high correlation of the model to the data, thus indicating a high predictive power to the model. Equation 10 shows the HBTM with estimated parameters.

$$B(\mathrm{x}) = 0.3930 + 0.2212\arctan(-15.1918(\mathrm{x} - 0.5471)) \quad x \in [0,1] \tag{10}$$

2. Horizon Determination Performance

When the horizon determination algorithm was applied to images containing a horizon, and given the correct orientation parameter, the algorithm performed very well in most cases. In some cases, the model results in a selection of pixels that nearly form a complete line through the image without the extra step of curve fitting. The results are listed in Table 14.

Table 14. Horizon Determination Results

Results Detected	Number of Images
No Horizon Pixels	7
Pixel Results Clearly not on the Horizon	18
Pixels Results Very Near the Horizon	128
Total Images with Horizon	153

The *Pixel Results Clearly Not on the Horizon* were made with best approximation of pixels determined by the algorithm to be horizon pixels, that were more than 1/10 of the image height or width, whichever was smaller, off the manually perceived horizon. When the algorithm became stuck in the maximum likelihood fitting of the model, or the model fit the data in such a way that it indicated a set of horizon pixels that were not contained within the image the *No Horizon Pixels* results. Figures 16 and 17 give a good idea of the difference in quality between the two categories of results. They are vastly different in the regions where the horizon pixels were detected.

Orange pixels were selected by the algorithm as horizon pixels.

Figure 16. Accurate Horizon Results

Orange pixels were selected by the algorithm as horizon pixels, the image on the left is a result where no horizon pixels were selected that are visible in the image.

Figure 17. Inaccurate Horizon Results

C. FITTING AND SELECTING THE HORIZON CURVE

In order to determine the best fit of the curve, it is important to note that via a RANSAC fitting algorithm, the models can be evaluated several ways. First, the model

can be evaluated by how well it describes the data, by examining the size of the consensus set.

The more traditional way to examine the quality of the fit is to examine the error in the curve to the dataset. Because of the quality of the horizon determination algorithm that selected the pixels, it was the case in this data set that several of the models were able to fit the data with a large consensus set, given that the allowable error was decided to be one twentieth the image width or height, whichever was smaller. For this data, that computes to an average error allowance of 25 pixels. Table 15 shows how each model stacked against each other when evaluated against the selected horizon pixels.

Table 15. Horizon Curve Model Results

Model	Number of Images	Avg Consensus Set Size	Avg Sum Squared Error
Linear	136	98%	3991px
Parabolic	115	94%	6591px
Circular	73	72%	495860px

Here we can see that the linear model fit more horizons overall than the other two models. It also boasts the highest average consensus set size. Following strictly which model had the least error, the linear curve is also the best model. However, looking at the individual results shows a tradeoff. In most situations where the linear model performs very well, the circular model fails to produce any result, indicating that most of the points were collinear and the linear model must be the best fit. In other cases, where the circular and parabolic curves do well, the linear model fails to produce a result, or the size of the consensus set sharply drops.

As far as which model best fits the horizon curve—it depends on how much of the horizon is observed. For images that have a smaller field of view, it would seem a linear model is sufficient to model the horizon line presented. For some instances where a small piece of the earth is available in a larger field of view image, the circular and parabolic models perform better. In the case of the dataset utilized here, overall the simplest and best performing model is the linear.

D. LIMITATIONS

Although the results for two-class horizon detection and the brightness transition model were very promising, each presents certain challenges. Specifically the results of the machine learning techniques are highly dependent on the training sets, and the horizon brightness transition model had its own limitations based on the lighting of the earth in the imagery.

1. Horizon Detection and Classification

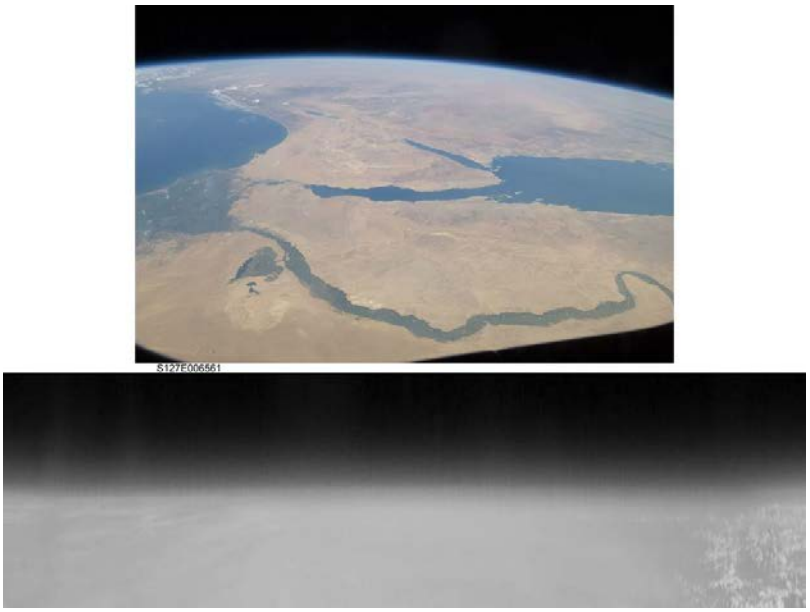
The classifiers chosen for the horizon classification problem greatly depend on the training sets provided to them. If the training set does not possess enough of each class, this can generate bias toward one classification over another. This was mitigated by providing an evenly distributed number of each class across the data set. It should be noted that in the eight and nine-class, classification models, the number of instances of each class is less than in the two-class scenario.

It is equally as important to provide as varying examples of each class in the data set. For example, if each *lower left* (LL) classified image in the training set contained a portion of the earth that only covered 1/8 of the image, and no other examples of the LL class were in the training set, the classifier could misinterpret which features of those input to it define the class. This was mitigated by included varying examples of images for each of the defined classes with varying levels of brightness and at different scales. In short, the larger and more varied the data set, the more robust the detection and classification should be. In short, provided a different training set, a different level of performance can be expected from these techniques.

2. Horizon Brightness Transition Model

Upon applying the data sampling techniques described in the methodology, it was very clear that there were some rough correlations in some of the images in the transition from the dark shades of space to the brighter earth pixels. This only occurred, however, where there was significant lighting on the earth. In the absence of the sunlight, the shades of the earth were largely determined by the condition of the atmosphere, which

varied results quite greatly. Figure 18 shows the ideal horizon image for the horizon brightness transition model to correctly identify horizon pixels, and when plotted with brightness on the y axis and horizon displacement on the x axis show a clear curve that neatly follows the arctangent shape that is seen in Section B. Some images, such as the one in Figure 19, which do not have a clear brightness transition, do not have a determinable horizon using the model described here.



The grayscale sample in the lower part of the image is a direct sampling of pixels roughly where the visual horizon occurs to the naked eye.

Figure 18. HBTM Ideal Horizon Conditions

The grayscale sample in the lower part of the image is a direct sampling of pixels roughly where the visual horizon occurs to the naked eye.

Figure 19. HBTM Imperfect Horizon Conditions

In building the model, it was clear that there were certain situations where the horizon brightness transition model does not work. These situations are very clearly marked by the absence of a lighting source for the earth. When the sun is not illuminating the visible portion of the earth, the scattered brightness of the light sources on earth do not reflect a consistent pattern that the model could fit. In order to further refine the model, the instances of where the model fails to work were removed for the final parameter evaluation. Because of this, it is a fair evaluation that this algorithm will not work for an image taken from the night side of the earth.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

A. SUMMARY

It is definitely desirable and achievable to detect a visible horizon and to determine the horizon line based on the models developed in this research. This thesis presented several methods to conduct the horizon detection and categorization, a means to select those pixels that will best represent the horizon, and evaluated different models for the shape of the horizon while using features from images in the visible spectrum.

For the task of horizon detection, several image features and machine learning techniques, KNN and SVM, were evaluated and a high precision, high recall solution for horizon detection was discovered. However, these features and techniques failed to properly classify horizons by orientation. This classification of horizons by orientation was important in simplifying the horizon determination portion of the algorithm, and therefore presents opportunities for future work.

This research also proved the viability of the horizon brightness transition model for determining which pixels in the image comprise the horizon when provided with a coarse horizon orientation. Though pixels were not necessarily exactly on the horizon perceived by the human eye, the model allows for adjustment of the selection point for the horizon. For this research, the inflection point of the models curve that was fit to the data was defined as the horizon. By selecting a different model horizon point, the model can be tuned to select a different set of pixels closer to the perceived horizon curve. The HBTM provides a means of consistently identifying the horizon with accurate results.

From the pixels determined to be the horizon by the HBTM, several models for the curvature of the earth were tested. The summary of the evidence did not conclude that one model best fit the shape of the earth completely, but that, in fact, the appropriate curve to select would depend on the field of view of the camera, and how much of the horizon is in view.

B. EXTENDED WORK

The classifying of the horizons by orientations via SVM and KNN did not prove very reliable. The reason is either a poor selection of classifiers, or a poor selection of features. The results of this classification were meant to determine the how an image was sampled during horizon determination, i.e., horizontal, vertical or diagonal sample lines. A future extension of this work should examine another means of classifying the horizon orientations. Particularly the examination of different features than those presented in this thesis, or a different classifier should be evaluated.

Another solution to this problem would be to expand upon the horizon determination algorithm in such a way that the horizon orientation information is not required. Particularly a method of evaluating each orientation of the sampling lines could be conducted to find the sampling lines that most likely are perpendicular to the horizon. By handling this internally to the horizon determination portion of the algorithm, the horizon orientation information would be no longer necessary.

Another area of investigation is to determine a method to select horizon pixels in using features from the visible spectrum in imagery where there is no direct sunlight on the earth. This was the key weakness of the horizon brightness transition model. It may be that another model different from the one presented here may better determine the horizon in all imagery, or that another model must be found to handle this case.

The next extension of this work, once acceptable results are achieved for the algorithm, is for it to be integrated into an attitude determination algorithm, and system in order to evaluate the quality of the horizon data provided and the accuracy of the attitude determination that can be made from horizon information that this algorithm would be able to provide.

C. IMPLICATIONS FOR THE FUTURE

As far as horizon detection, the data shows images sub-sampled to a 64 x 64 pixel, or 4096 dimensional representation, is a suitable feature set for utilization with K-Nearest Neighbors machine learning technique to produce a high precision and recall rate. In short, this algorithm satisfactorily detects whether an image has a horizon. It does

this with sufficient precision to ensure that most horizon containing images are selected for processing, and sufficient recall to ensure that very few of the images not containing a horizon are subject to evaluation by the rest of the algorithm.

The horizon brightness transition model, as a horizon determination algorithm shows great promise for future work. In the vast majority of the images run through the horizon determination algorithm, 84% of the images showed horizon pixels at or near the horizon perceivable by the human eye.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] F. Ribbehege, “Development and testing of a high precision prototype star tracker for CubeSats,” M.S., thesis Naval Postgraduate School, Monterey, CA, 2015.

[2] J. Tappe, “Development of star tracker system for accurate estimation of spacecraft attitude,” M.S., thesis, Naval Postgraduate School, Monterey, CA, 2009.

[3] G. Young et al., “Robust real time horizon detection in full-motion video,” in *SPIE Defense + Security*, vol. 9076N.

[4] D. Byers. (2011, Dec, 21). Picture special: The earth from space in 2011. *Times* [Online]. Available: http://www.thetimes.co.uk/tto/science/astronomy/article3264741.ece

[5] J. Wertz, K. Liu, J. Rowe, “Modeling the earth” in *Spacecraft Attitude Determination and Control.* J. Wertz, Ed. Dordrecht, Holland: D. Reidel Publishing Company, 1978.

[6] W. Abdou and S. Tantiphanwadi. “The development of an attitude sensing system for small satellites,” in *Annual AIAA/Utah State University Conf. on Small Satellites*, 1988, 12 pp.

[7] W. Steyn. “An attitude control system for SumbandilaSAT an earth observation satellite,” in *Proc. of the IAA Symp. Small Satellites Systems and Services,* August 2008.

[8] H. Loubser, “The development of sun and nadir sensors for a solar sail CubeSat,” M.S. thesis, Univ. of Stellenbosch, Stellenbosch, South Africa, 2011.

[9] G. Lerner, “Attitude hardware” in *Spacecraft Attitude Determination and Control.* J, Wertz, Ed. Dordrecht, Holland: D. Reidel Publishing Company, 1978.

[10] “horizon,” *Oxford English Dictionary Online,* 2016.[Online]. Available: http://www.oed.com/view/Entry/88458?rskey=i3LCkN&result=1. Accessed: Aug. 12, 2016.

[11] S. Ettinger, M. Nechyba, P. Ifju, M. Waszak, “Towards flight autonomy: vision based horizon detection for micro air vehicles,” in *Florida Conf. on Recent Advances in Robotics*, vol. 2002, 2002.

[12] M. Tschudin, “Refraction near the horizon-an empirical approach. Part 1: terrestrial refraction of the dip.” *Applied Optics*, vol. 55, no. 12, pp. 3104–3115., Apr. 2016.

[13] S. Ettinger, “Design and implementation of autonomous vision guided micro air vehicles,” M.S., thesis Univ. of Florida, Gainesville, Florida, 2001.

[14] S. Fefilatyev, V. Smarodzinava, L. Hall, D. Godlgof, “Horizon detection using machine learning techniques,” in *Proc. of the 5th International Conf. on Machine Learning and Applications (ICMLA ‘06),* 2006, pp. 17–21.

[15] D. Forsythe and J. Ponce, “Edge Detection,” in *Computer Vision: A Modern Approach,* Upper Saddle River, NJ, Prentice Hall, 2003.

[16] R. Szeliski, “Feature detection and matching,” in *Computer Vision Algorithms and Applications,* New York, NY, Springer, 2011.

[17] D. Dusha, W. Boles, R. Walker, “Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow,” in *9th Biennial Conf. of the Australian Pattern Recognition Society*, 2007, pp. 485–492.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California