Detection and Segmentation of Ship Exhaust Tracks off the Californian Coast

by

Tobias Schmidt

Under the Supervision of Dr. Philip Austin

Signature: [illegible]

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Science (Combined Honours)
(Atmospheric Science and Computer Science)

in

THE FACULTY OF SCIENCE
(Department of Earth, Ocean and Atmospheric Sciences,
Department of Computer Science)

The University of British Columbia
(Vancouver)

April 2021

Table of Contents

List of Figures

Glossary

ABI	Advanced Baseline Imager
GOES	Geostationary Operational Environmental Satellite
AWS	Amazon Web Services
NOAA	National Oceanic and Atmospheric Administration
BTD	Brightness Temperature Difference
IID	Independent and Identically Distributed
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
MOSSE	Minimum Output Sum of Squared Error
KCF	Kernelized Correlation Filter
CSRT	Channel and Spatial Reliability Tracker

Chapter 1

Introduction

Cloud-aerosol interactions are an area of considerable uncertainty in our current understanding of climate change. Multiple competing processes influence the expected net effect of changes in aerosol concentration on cloud properties. Generally, an increase in atmospheric aerosols might be expected to have a cooling effect as they tend to produce brighter, more reflective clouds (Chen et al. [7]). Clouds modified by anthropogenic aerosols also tend to be at a low enough altitude such that they don't have a thermal greenhouse effect to counteract the increased cooling due to the reflectivity change. On the other hand, increasing aerosols can also accelerate cloud breakup by suppressing droplet sedimentation, leading to enhanced evaporative cooling, and enhanced mixing of inversion air resulting in a transition to broken cloud (Bretherton et al. [5], Ackerman et al. [1]). This implies that the warming effect of shorter cloud lifespans may dominate over the cooling effect. The uncertainty was large enough to warrant both changing and removing sections pertaining to cloud-aerosol interactions from several Intergovernmental Panel on Climate Change reports (Boucher et al. [3]). It is clearly important to gather as many real-world observations of this process as possible to reduce these unknowns. One possible anthropogenic example of these observations are ship exhaust tracks. We propose sampling en masse from these ocean-based clouds as we can confidently state they are filled with cloud-aerosol interactions. These tracks are a "natural experiment" of sorts that cannot be disregarded.

The first step in sampling from these structures is to locate them efficiently and robustly. Existing studies focused on the detection of ship exhaust tracks have already shown considerable promise. An excellent example being the work done by (Yuan et al. [24]) where a convolutional neural network is trained from 1,500 examples to find ship tracks. The results from this study were impressive but the number of examples their model was trained on highlights one of the primary disadvantages to using machine learning: an expansive amount of data is required to achieve a non-trivial result. A considerable amount of effort to manually gather and classify all of this data would be necessary.

The purpose of this study was to create an algorithm for finding ship track examples that does not require large datasets or training. This would be achieved by exploiting the persistence of a ship track's linear structure through time relative to its more noisy surroundings. Our dataset's domain was the ocean off the coast of California during the spring, summer, and fall months. This area has heavy ship traffic and is known to produce the ideal atmospheric conditions needed for the development of cloud droplets from the ship aerosols. Specifically, 2019 was chosen since the Californian wildfire season was less extreme and less smoke would be expected to block the satellite's line of sight to low level clouds. See **fig. 1.1** for an example of a ship track heavy event near California.

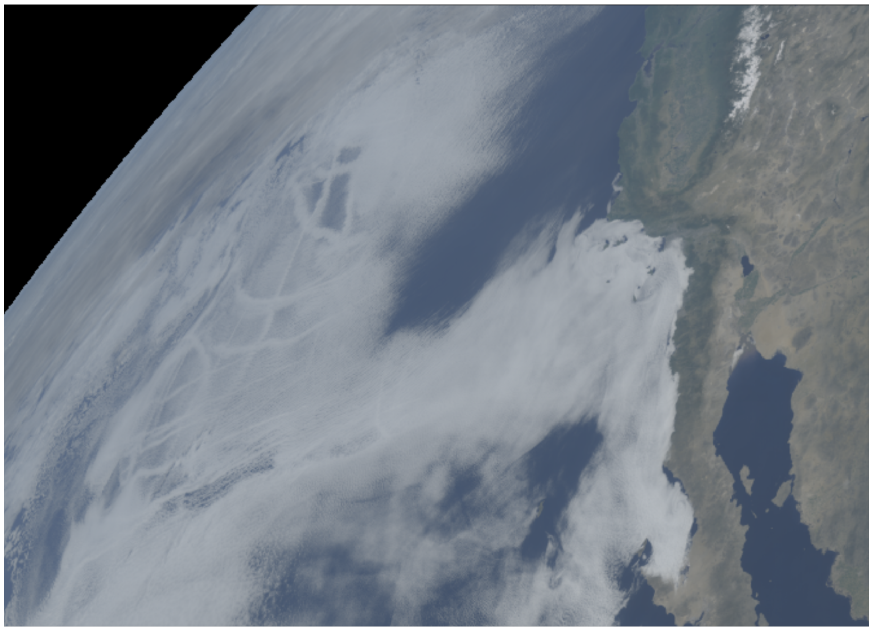


Figure 1.1: Visible satellite view of ship exhaust tracks off the coast of California on Apr. 24, 2019

Chapter 2

Methodology

Cloud droplets created from ship exhaust aerosols have been shown to display identifiable signatures in remote sensing data (Szczodrak et al. [21]). From testing we found that these signatures were not robust enough to be the exclusive step in ship exhaust track masking. Many other pixels in the scene were erroneously selected and some ship tracks were not detected at all. Additional steps needed to be taken in order to have a ship track masking algorithm accurate enough for use in machine learning training, validation, and testing datasets. This study explored the use of some modern computer vision techniques on the visual structures associated with ship tracks in an attempt to further differentiate between ship track and non-ship track pixels within these signatures. The following sections will be presented in order of their positions inside our algorithm.

2.1 Data Selection and Pre-Processing

The Geostationary Operational Environmental Satellite 17 Advanced Baseline Imager (GOES-17 ABI) was selected as the primary data source for this study. This choice was made because geostationary satellites are ideal for algorithms that require multiple images in temporal succession. The National Oceanic and Atmospheric Administration (NOAA) has also made many of the GOES datasets available on modern cloud storage services such as Google Cloud and Amazon Web Services (AWS) which made the data mining process relatively effortless. The final version of the study's software required the following raw input data at each time step (each are full disk files, defined by the prefixes ABI-XX-XXXF below):

1. ABI channel 2 radiances with 0.59–0.69 µm wavelengths (ABI-L1b-RadF)
2. ABI channel 7 radiances with 3.80–4.00 µm wavelengths (ABI-L1b-RadF)
3. ABI channel 14 radiances with 10.8–11.6 µm wavelengths (ABI-L1b-RadF)
4. ABI level 2 cloud particle size (ABI-L2-CPSF)
5. ABI level 2 cloud optical depth (ABI-L2-CODF)

Additionally, brightness temperature difference (BTD) was produced from the channel 7 (3.80–4.00 µm) and 14 (10.8–11.6 µm) radiances to be used as the primary input for the computer vision algorithms. Brightness temperature refers to the temperature that a blackbody would have to be at to produce the observed radiance. BTD was chosen because it defines the structure and boundaries of the ship tracks particularly clearly (Yuan et al. [24]). BTD was generated from the following equations (Schmit et al. [20], Yuan et al. [24]):

$$\mathrm{BTD} = T_{\mathrm{ch7}} - T_{\mathrm{ch14}}$$

$$T = \frac{\frac{\mathrm{fk2}}{\ln(\frac{\mathrm{fk1}}{L_\lambda}+1)} - \mathrm{bc1}}{\mathrm{bc2}}$$

Where L_λ is the desired channel's radiance and fk1, fk2, bc1, bc2 are constants determined by the wavelength of the chosen channel and universal constants used in Planck's law. See table 8 in Schmit et al. [20].

In order to properly exploit the chosen computer vision algorithms, several pre-processing filters were used to prepare the data. Portions of the data that may have caused unnecessary noise or that were not physically associated with ship tracks were removed. The first filter removed any data situated over land. This was achieved by checking to see if each pixel fell inside of the coastline polygon shapefile distributed by OpenStreetMaps (OpenStreetMap contributors [17]). The second filter removed any high-cloud data as these clouds are not usually associated with ship tracks. A data point is classified as a high-altitude cloud if its channel 14 (10.8–11.6 μm) brightness temperature is below 5°C. This was considered cold enough to represent high-altitude clouds at this latitude during the spring and summer months that this study covers.

The final filter removed open ocean pixels as these are clearly not associated with ship tracks and would often introduce a lot of noise. A GOES-17 ABI open ocean mask is supplied by NOAA but we decided that a filter could be made from the datasets already in use. The development process for this filter was more elaborate than the others and several methods were tested before reaching an optimal solution. The first method that was tested was more traditional. This approach involved separately applying a local mean filter and a standard deviation filter to the channel 14 (10.8–11.6 μm) brightness temperature (with various kernel sizes). Next, the pixels that have both local mean above a predefined threshold and local standard deviation below a predefined threshold is considered open ocean (Coakley Jr. and Bretherton [8]). When a scatter plot is generated with the local mean data on the x-axis and the local standard deviation data on the y-axis, the structure that is observed is often compared to an ”arch”, where the base of the right ”column” corresponds to data points over open ocean (Coakley Jr. and Bretherton [8]). The two aforementioned thresholds were designed to select the data on this right column's base. This method had shown promise during some of the early test cases however, we concluded that this method was not robust enough to work in all scenarios. Either universal thresholds could not be determined or, in certain scenes, the expected arch structure did not appear at all.

Clearly, a more robust method was required for reliable open ocean filtering. What was next investigated was using channel 2 (0.59–0.69 μm) radiances and channel 14 (10.8–11.6 μm) brightness temperature with a clustering technique. A density highlighting hexbin plot was produced for these two dimensions and a dense number of pixels was observed at the bottom right (see **fig. 2.1**). This was observed to be a signature directly associated with pixels over open ocean. The density of the signature implied that a density-based clustering algorithm would robustly and accurately label the correct pixels. The chosen method was the popular density-based spatial clustering of applications with noise (DBSCAN) algorithm. The method detects clusters of arbitrary shape using the density as a measure of likeness between points (Ester et al. [9]). The primary drawbacks to DBSCAN are that it is more computationally expensive than KMeans and it is non-inductive (it cannot predict the labels of new data) (Pedregosa et al. [18]). We found that it would take upwards of 1 hour to run a single scene, which was far greater than the few minutes of runtime per scene that would be acceptable. The most direct solution to the runtime issue was to train the model on a smaller subset of the pixels but the non-inductive property of DBSCAN implied that this could not be done. The solution was to merge DBSCAN with a classification algorithm through a method outlined in Scikit Learn's API documentation (Pedregosa et al. [18]). The clustering/classifying algorithm combination could work in a pseudo-inductive fashion. 10,000 pixels were IID sampled (the samples were independent and identically distributed implying each sample was chosen randomly and had no co-dependence) from the image and used to fit DBSCAN and then a decision tree was used as a classifier for predicting the label of the other pixels. This was an adequate solution to the runtime issue as it brought down the average runtime per scene to approximately 2 minutes on a 2020 MacBook Pro.

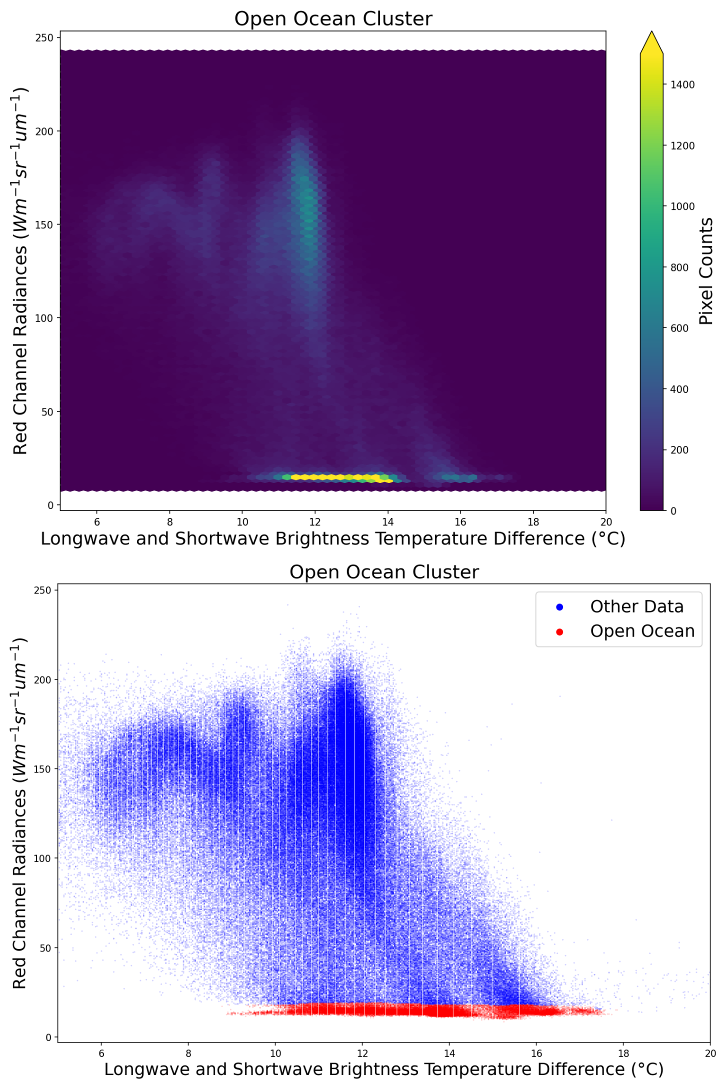
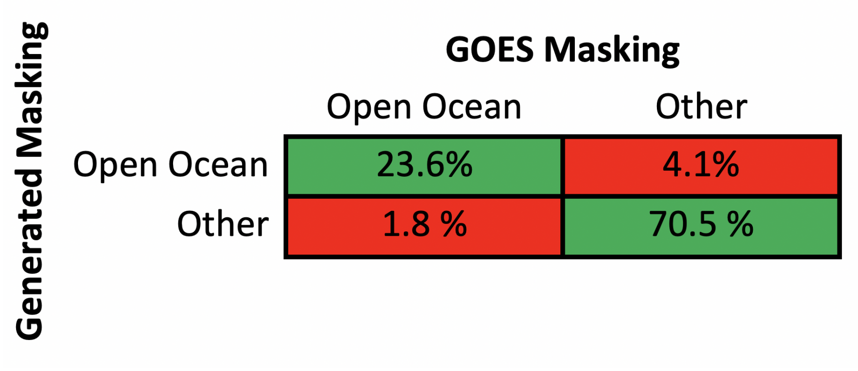


Figure 2.1: Top: A hexbin plot from Apr.24th 2019 showing the clusters associated with channel 2 (0.59–0.69 µm) radiance and the brightness temperature difference. **Bottom:** A scatterplot of the same event where the results of the clustering algorithm are highlighted in red.

The clusters found seemed to qualitatively match the cluster we expected to observe (see **fig. 2.1**). Using the GOES-17 open ocean mask as a true label, the error of the clustering process is described below in **fig. 2.2**. The error is within acceptable bounds, but it should be noted it may be possible to improve it further by additional hyperparameter tuning. Also note that the GOES mask consistently, but randomly mislabels large patches of pixels over obvious open ocean areas during each frame. These patches change their frequency and position between each frame but are certainly widespread enough that the error was considerably worsened. The actual performance is better than what is displayed. These patches are nonphysical and should this process be studied further, a better set of true labels will have to be considered. This process's potential was not exhaustively explored as this was not the main objective of this study.



Generated Masking \ GOES Masking	Open Ocean	Other
Open Ocean	23.6%	4.1%
Other	1.8 %	70.5 %

Figure 2.2: Confusion matrix of the clustering algorithm generated from 174 scenes over 4 independent days.

2.2 Edge Detection

Data processing begins with the application of an edge detection filter. This filter finds adjacent pixels with a difference in magnitude over a user defined threshold. It then creates a line though these adjacent points referred to as an ”edge”. To find an edge detection filter that reliably highlighted ship tracks, while generating as little edges as possible for other artifacts such as clouds or dataset noise was critical.

Firstly, the Sobel edge detection filter was tested. This is a common filter used in many photo processing applications. An example of it being used to assist with face recognition was made by (Thakur [22]). We found this filter to be overly sensitive, but reducing the edge threshold would just cause the filter to miss less well-defined ship tracks. Several ridge detection filters were also tested. Ridge detection differs from edge detection in that it finds the overall ship track structure rather than just the edges. Ridge detection is often used in the identification of human anatomy features such as neurites, tubes, vessels, and wrinkles (Frangi et al. [10], Meijering et al. [14], Ng et al. [16], Sato et al. [19]). Four methods were tested: the Meijering filter, Sato filter, Frangi filter, and Hessian filter. We observed that all four of these ridge detection methods could not resolve small diameter ship tracks. These methods also returned structures that were not adequate for input in later steps of our pipeline. With these issues in mind, ridge detection methods were subsequently discarded.

The method that had qualitatively delivered the best results was the Canny edge detection filter. Edges are found in the Canny edge detection filter by first calculating the gradient of the pixels. Areas with the largest local gradient magnitude are considered edges. The less well-defined edges (that likely originated from noise) are removed using a hysteresis threshold set by user defined parameters (that are discussed below). It is very similar in output to the Sobel edge detection filter, with the exception that it has a built in Gaussian filter that is applied before finding edges. The Gaussian filter became a crucial part of our process as it ensured ship tracks became more well-defined while greatly reducing background noise. Without some degree of Gaussian smoothing it is unlikely an edge detection algorithm could find any tracks. The parameter σ controls the width of the Gaussian function (a higher σ indicates a greater amount of smoothing), while low and high threshold parameters determine the amount of edges detected. The low threshold determines how long an edge may be, while the high threshold determines if an edge exists at a particular point. After initial tests, we determined qualitatively that $\sigma = 2.5$, low_threshold $= 0$, and high_threshold $= 0.8$ would produce the optimal results. See **fig. 2.3** for an example of this filter.

2.3 Probabilistic Hough Transform

The edges were created to identify the outline of the ship tracks, but it is impossible to exclusively highlight their structures without highlighting at least some of the background scene. The next step was to single out the tracks from the background. This was achieved by exploiting the linear structure associated with tracks. (Aircraft contrails would not be detected because the high cloud filter mentioned in section 2.1 removed them from the data.) To detect and flag straight lines, the probabilistic Hough transform algorithm was used. This algorithm detects straight lines using the Hesse normal form ($r = x\cos(\theta) + y\sin(\theta)$) to find the distance between each line point and the origin (r) along with the line's angle (θ) (Galamhos et al. [11]). It has three parameters: a threshold, a max line gap, and a min line length. Respectively these limit the number of lines detected, set the maximum allowed number of empty pixels between each line pixel, and set the minimum length for a line to be considered valid. This detected most tracks in a particular frame but also still detected a few erroneous lines within the background (see **fig. 2.3**).

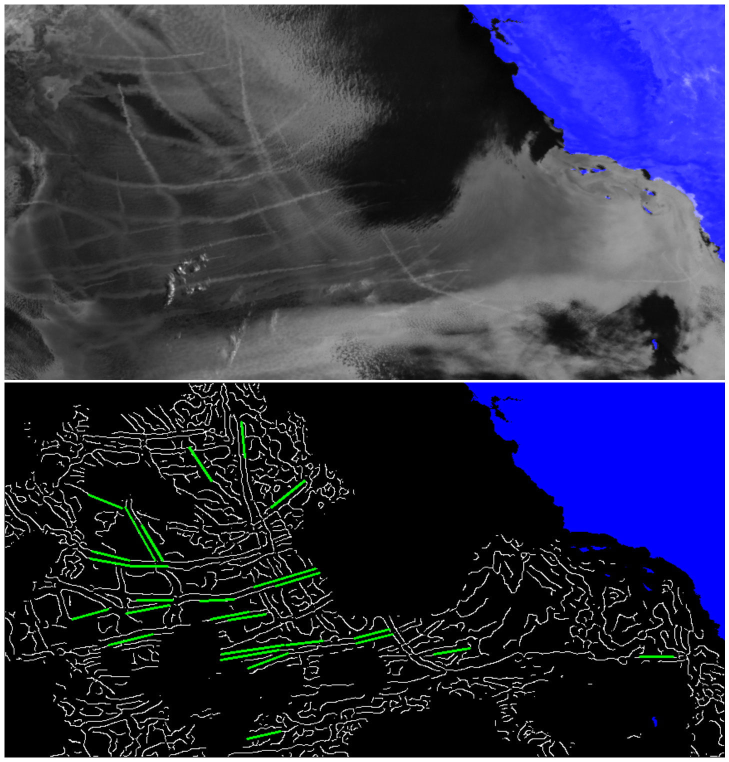


Figure 2.3: Top: The brightness temperature difference on Apr.24th 2019 with the Californian coast highlighted in blue for reference. Darker colors indicate more negative BTD while brighter colors indicate more positive. **Bottom:** The Canny edge detection of the same event with the probabilistic Hough transform detecting straight lines indicated in green.

2.4 Discriminative Correlation Filter Tracking

We observed that the noisy edges associated with the background would not maintain their structure through the passage of time as well as ship tracks. This integral observation would become the inspiration for the primary component of this study's algorithm. This component being the use of an object tracking filter that would take into account the object's temporal state. The lines supplied by the probabilistic Hough transform were then fed into the filter and only those lines with surrounding pixels that remain mostly unchanged during a certain amount of time are deemed associated with true ship tracks.

Kalman filters are among the most popular and commonly used object tracking filters (Chen et al. [6]). They would seem ideal for our purposes as they contain a component that directly describes the error associated with an observations' deviation through time. However, we decided that similar, but more modern object tracking algorithms would be used. This is due to their widespread use in modern computer vision applications (Chen et al. [6]) and their public distribution through python packages such as OpenCV. Several of the OpenCV methods were tested. First the Minimum Output Sum of Squared Error (MOSSE) tracker was used. It was a very quick and efficient algorithm but it failed to track almost any of the ship tracks (Bolme et al. [2], Bradski [4]). It was quickly discarded and replaced with the Kernelized Correlation Filter (KCF) Tracker. This tracker performed more slowly, but with much more accuracy. Unfortunately, it often could not track the ship tracks beyond 3-5 frames (Henriques et al. [12], Bradski [4]). Thirdly, we experimented with the Discriminative Correlation Filter with Channel and Spatial Reliability (or CSRT in OpenCV vernacular) (Lukezic et al. [13], Bradski [4]). The spatial reliability here refers to the model constantly updating the tracked segment of the scene as it moves through space. This filter tends to run slower than the other object trackers distributed by OpenCV, and is often considered too computationally expensive for intensive computer vision applications such as tracking an object on a 60 frames per second video stream. For our applications, this limitation was not an issue as we did not require quick results for each frame and a single day's worth of data rarely exceeded 50 total frames. This filter performed far better than the alternatives and was often able to track structures for the duration of a day. This filter clearly became the ideal choice for our algorithm.

A system for managing all of the objects being tracked by the CSRT tracker had to be devised. This was essential for determining if a particular object would be considered a true ship track. It was also important for determining when and if a ship track dissipated or moved off the scene. OpenCV has a MultiTracker class that is meant to perform this task (Bradski [4]), but it was lacking several key functions. The primary issue being that it could not remove an object once it had started to be tracked. This was clearly an issue as a ship track could leave the scene, but the tracking algorithm may still try to find it elsewhere, potentially producing a false positive. The OpenCV class also could not be easily modified to support the final thresholds we wanted to enforce (listed below). This led us to create our own version of a CSRT MultiTracker class. It supported the removal of any tracked object at any time. An object was removed for any of the following reasons:

1. The CSRT tracker could no longer detect the object in the scene.
2. The object was not detected for at least 3 consecutive frames in a row (or 30 minutes).
3. The object moved too quickly to be an average ship track (in excess of approximately 1 km/min).
4. The object was not detected at least twice during its existence by the probabilistic Hough transform.

Items 2-4 are of greatest importance as these are the thresholds that exploit the temporal aspect of the tracking algorithm. If none of these are triggered the object is formally considered a ship track and can next be labelled as such.

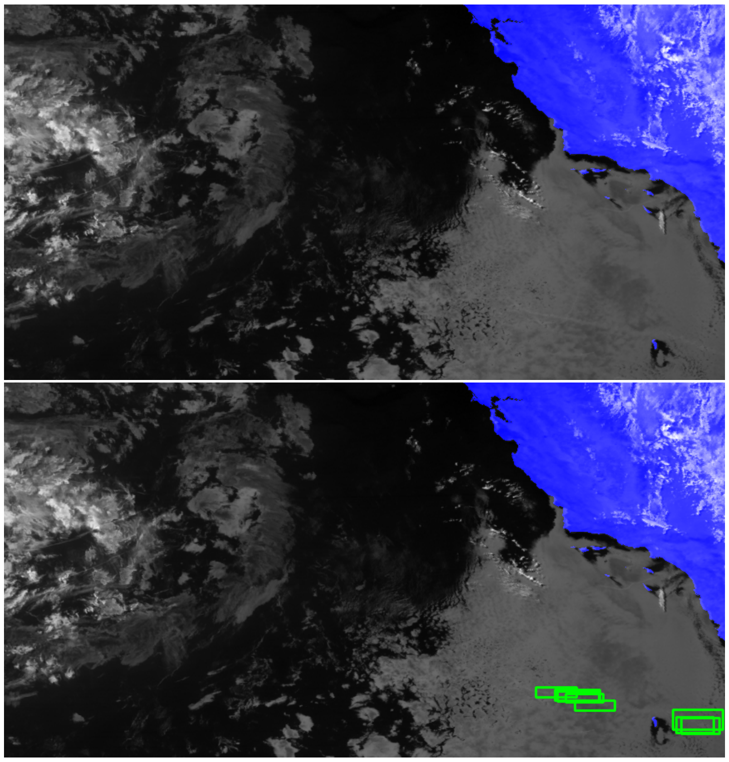


Figure 2.4: Top: The brightness temperature difference on Aug.7th 2019 before the CSRT algorithm is run. Note the ship tracks in the bottom right of the image. **Bottom:** CSRT algorithm highlighting several sections of those tracks with green boxes.

2.5 Ship Track Pixel Labelling

Each object tracked in the previous section is bounded by a rectangular box (see **fig. 2.4**). The pixels within these boxes is what the tracking algorithm searches for during each frame. The tracking algorithm itself cannot recognize which pixels within this box belong to the ship tracks and which do not. An additional step was required to perform this final image segmentation.

As with the other sections, several methods were considered and tested before a procedure was decided upon. Firstly, functions using channel 2 (0.59–0.69 µm) and channel 6 (2.225–2.275 µm) reflectances identified in the GOES ABI documentation and by others (Walther et al. [23], Nakajima and King [15]) were experimented with (**see fig. 2.5**). Each of these functions highlighted a relationship between channel 2 (0.59–0.69 µm) reflectances, channel 6 (2.225–2.275 µm) reflectances, and cloud optical depth for a particular cloud droplet size. Where cloud optical depth and cloud particle size are derived following (Szczodrak et al. [21]):

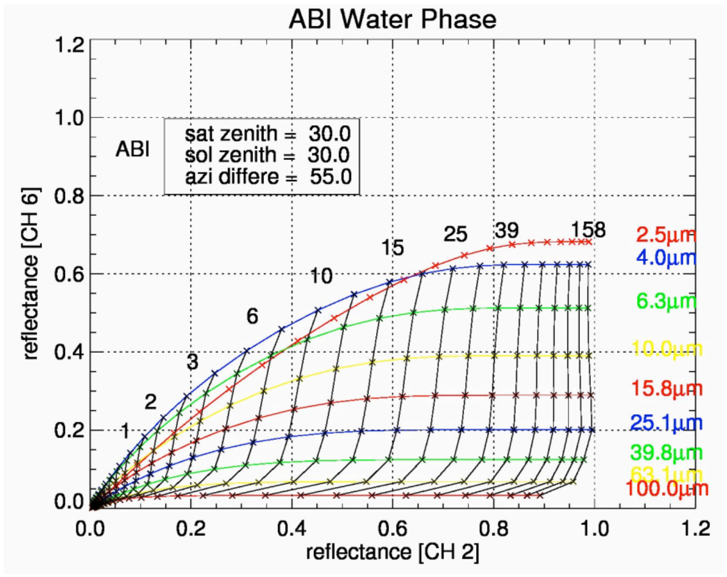
$$\tau = Q_{\text{ext}} \int_0^H \left[\int_0^\infty n(z,r) \pi r^2 dr \right] dz \tag{1}$$

$$r_{\text{eff}}(z) = \frac{\int_0^\infty n(z,r) r^3 dr}{\int_0^\infty n(z,r) r^2 dr}$$

Where r is the droplet radius in meters, $n(z,r)dr$ is the number concentration of drops at height z with radii between r and $r + dr$, and Q_{ext} is the scattering extinction efficiency ($Q_{\text{ext}} = 2$ for channel 2's 0.6 µm photons). The scattering of channel 2 photons is not sensitive to droplet size, but for channel 6 photons we expect to see reflectively vary strongly with droplet size. Since we expect ship tracks to be associated with smaller diameter cloud droplets relative to their surroundings, we theorized that some parts of a scene would resemble the GOES ABI graph (**fig. 2.5**) in the channel 2 (0.59–0.69 µm) and channel 6 (2.225–2.275 µm) reflectance space. The ship tracks would have a curve slightly below other curves representing their surroundings. Unfortunately, in practice no obvious curve like structures with vertical spacing were identified. This was primarily because our study focused on clouds with relatively low optical depth. As shown in **fig. 2.5**, particle size is strongly differentiated by the channel 6 (2.225–2.275 µm) reflectance when the optical depth is large, but the relationship is weaker when the optical depth is small. The thinness of the cloud layers resulted in the sampled data points forming an overly noisy, structureless shape.

This issue inspired the next method, which was to use a clustering algorithm. We theorized that perhaps a clustering algorithm would be able to piece apart the noise and find a ship track signal. Both KMeans and the inductive combined DBSCAN/decision tree algorithm from section 2.1 were run on several combinations of normalized variables in an attempt to find a signal. Firstly, channel 2

(0.59–0.69 µm) and channel 6 (2.225–2.275 µm) reflectances were examined. This was to determine if there was a ship track cluster concealed inside of all the noise that was not visible with inspection. No satisfactory results were produced, so additional options were explored. At the conclusion of these tests 7 different variables that were theorized to accentuate ship tracks were tested in varying combinations. In no particular order they were: channel 2 (0.59–0.69 µm) reflectance, channel 6 (2.225–2.275 µm) reflectance, our BTD, cloud particle size, cloud optical depth, channel 14 (10.8–11.6 µm) radiance, and channel 7 (3.80–4.00 µm) radiance. We observed that no combination produced robust results with either KMeans or DBSCAN. While some ship tracks would be successfully labelled during these tests, many would remain unlabeled and a substantial amount of background cloud would be erroneously labelled. Clearly, additional methods would still need to be explored.



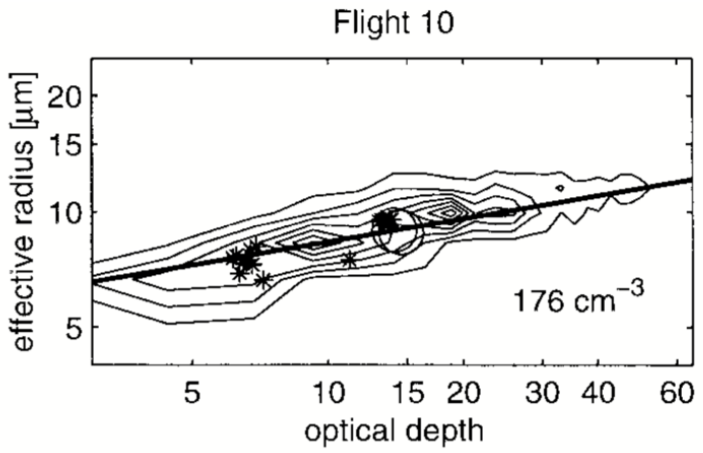


Figure 2.5: Top: Graphical depiction of the relationships between channel 2 (0.59–0.69 µm) reflectances, channel 6 (2.225–2.275 µm) reflectances, cloud droplet sizes, and cloud optical depths given in NOAA's manual (Walther et al. [23]). The optical depths are the values above the lines while the droplet sizes are to the right of each line. **Bottom:** The observed relationship between optical depth and droplet size in a polluted cloud given by Szczodrak et al. [21]. The contour lines give τ and r_{eff} values for satellite pixels. The starred points are in situ aircraft data.

The final method explored was similar in concept to the (Walther et al. [23], Nakajima and King [15]) method. We decided that the identification of a traditional function relating two dimensions was the best way to label the tracks with accuracy acceptable enough for our purposes. NOAA produces derived data products from the raw radiances. Two of these products are the cloud optical depth and the cloud particle size. They are inferred using a lookup table derived from the radiative transfer calculations shown in **fig. 2.5**. In a separate study Szczodrak et al. [21] have identified a function relating cloud optical depth to cloud particle size through satellite and aircraft observations of clouds. Specifically, if the number of droplets is assumed to be constant, they showed that (1) can be rewritten as a power law relationship between cloud optical depth and cloud particle size:

$$\tau = a_1^{-1} N_{\text{sat}}^2 r_{\text{eff}}^5$$

Where τ is the optical depth, a_1 is a constant based on the scattering extinction and the liquid water lapse rate set at $2 \times 10^{-3}\text{gm}^{-3}\text{m}^{-1}$, N_{sat} is related to the number of droplets in the cloud, and r_{eff} is the effective radius of the droplets at cloud top. Refer to Szczodrak et al. [21] appendix A for details of its derivation. When cloud optical depth and cloud particle size are plotted in a loglog scatterplot, this can be visualized as a linear relationship (a line). Flight 10 in **fig. 2.5** (Szczodrak et al. [21]) gives an example of this relationship. This flight was through clouds polluted with aerosols which is an environment similar to what would be expected within ship track clouds.

After some trial and error, one or more straight line structures were observed in most scenes we examined (see **fig. 2.6**). A certain amount of pixels above and below this line were labeled as ship tracks, producing a rectangular shape. This algorithm was found to produce the least amount of erroneously labelled pixels in the immediate vicinity around the ship tracks and was therefore the best choice for labeling within the CSRT boxes. Unfortunately, we found that the exact position and slope of these lines varied between each scene. This is due to the number of particles differing between each track in a scene which broke our assumption for certain tracks. This implied the method was not universally robust and could not be used for consistent detection. Alternative methods for future studies are discussed in the results and conclusions section.

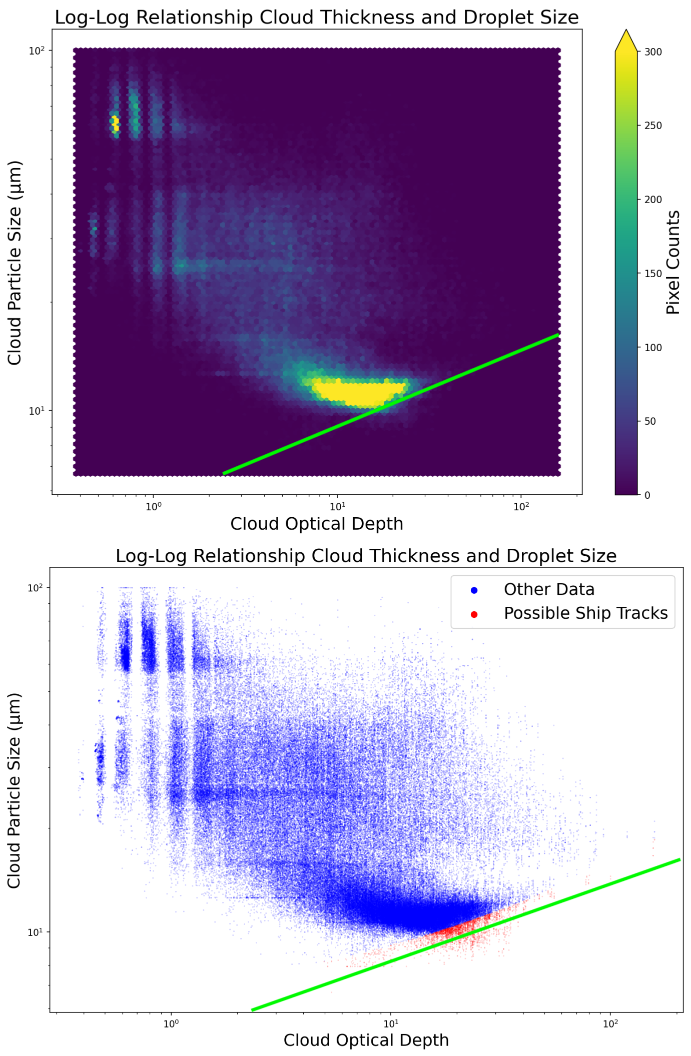


Figure 2.6: Top: The loglog hexbin plot showing the overall structure of the scene on Aug.7th 2019. The green line indicates the chosen function. **Bottom:** A loglog scatterplot of the same scene. The green line indicates the chosen function with its surrounding rectangle highlighted by the red points. These red-labeled pixels are primarily associated with ship tracks.

Chapter 3

Results and Conclusions

Several scenes with completed output are shown below in **fig. 3.1**. Our algorithm seems to correctly label pixels that are clearly part of ship tracks, but it fails to find all tracks. It also tends to consistently miss the ”tips” of the tracks where the clouds taper out. The objective of this study was to create an algorithm that could reliably return pixels that we confidently believed were from ship tracks. Our output seems to indicate that it achieves this, but it does not return all ship track pixels in a scene. It is not immediately clear if the pixels that are returned are associated with a certain amount of pollution (and therefore a stronger signal) or if the algorithm requires a certain width of track for a successful hit. The algorithm also tended to not perform very well in scenes with a large number of tracks. This was because the constant particle radius assumption required for the section 2.5 step was far more likely to break down with a greater number of tracks. A further issue that limits the number and type of ship tracks returned is the assumption made for the Hough transform step. That is the assumption that ship tracks are straight cloud structures with turns only at sharp angles. This assumption does hold for most tracks since most ships do not move in massive curving turns or circles. The issue arises when certain environmental conditions cause the wind to advect the track with differing velocities across its length creating a curved line. These large curves cause the track to become completely undetectable by our method.

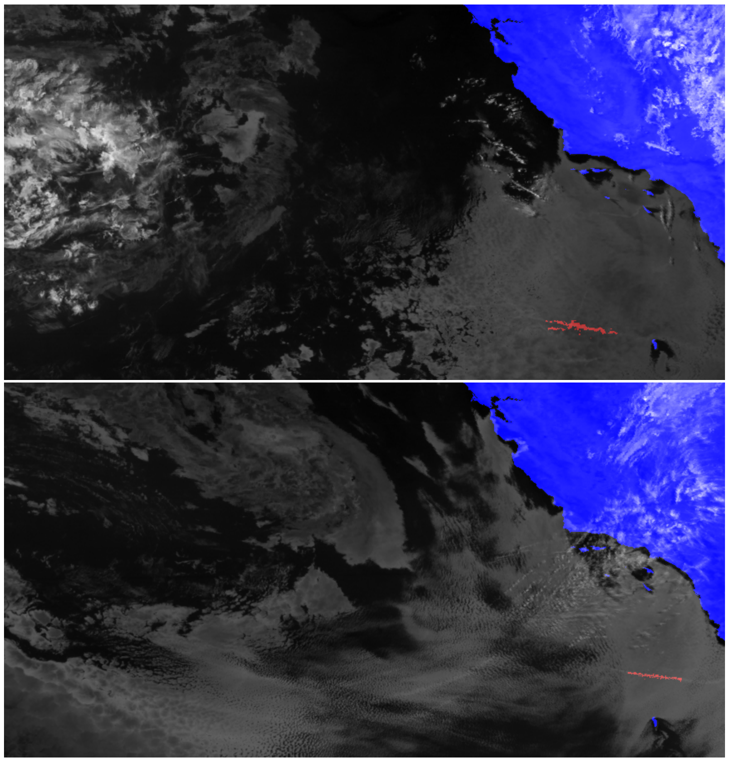


Figure 3.1: Two examples of completed output displayed on BTD images. The red highlighting represents the pixels returned as ship exhaust tracks. The pixels appear to be over fairly well defined tracks. **Top:** Aug.7th scene. **Bottom:** Jul.3rd scene.

While the results of this study appear promising, there are still several more steps that must be taken in order to ensure this method would be ready for operational use. It is clear that a greater level of error and hit rate quantification is required to guarantee the reliably of the results. At a high level we are comfortable with the method missing some of the aforementioned cases, but we must make sure that what is detected originates from a ship track with very little probability of error. A false positive rate is likely an ideal measure for this, but it is not clear what population should be counted. A population of raw labeled and non-labeled pixels were considered, but this results in error values that do not offer insight into the actual behaviour of the model. For example, a single pixel immediately adjacent to a ship track might have been erroneously labelled, but practically this should not contribute to the false positive rate. Instead, an objective for a future study would be to create an error measuring system where the population samples are the ship tracks themselves. Each ship track that is at least partially returned by the model would be counted as a single true positive sample and any entity that is erroneously returned outside of a ship track would be counted as a single false positive sample. This would make each counted false positive sample a very obvious representation of the error and would emphasize the significance of getting anything above a zero. In addition to using this system of error measurement on observations, it could be used with software that simulates scenes filled with ship tracks. This would ensure a controlled environment for stress testing our algorithm. All of these ideas were considered and would have been explored fully given less of a time constraint.

The most significant issue that would have to be overcome for this method to be robust enough for operational use is the limitations imposed by the manual work required in the section 2.5 step. This step must be replaced by a method that robustly works across all possible scenes. An example of this is a fully automated method for finding the relationship between cloud optical depth and cloud particle size. As mentioned in the section, some clustering methods were already tested with little success. A more likely choice will be the use of an entirely different process on different datasets, but that alone could be a separate study. However, if a method is found and the error is quantified, this model could have uses in operations or further climate studies.

Many future studies could originate from this project as there are still paths left unexplored. For example, one of the primary benefits of using BTD is that it can still highlight ship tracks during the night. It is also possible that the tracks may be more well defined at night as the BTD may become larger for their pixels since the emitted photons would not be partially masked by reflected sunlight. Our algorithm is restricted to the daytime due to the open ocean masking so future work could be focused on transitioning the algorithm over to nighttime. Another idea was to transition the used datasets to anomaly variables rather than traditional absolute variables. It is possible this will make the difference between the ship tracks and the background noise even greater for the CSRT step and improve the true positive rate. Another limitation with the data is that the temporal component of the algorithm forces it to originate from geostationary satellites which leaves many high resolution modern datasets unusable. In particular, the high resolution data from the moderate resolution imaging spectroradiometer (MODIS) sensor on the Terra and Aqua satellites would have likely proven invaluable and a future study could be focused on finding a way to use MODIS data in tandem with GOES data.

Overall this method has great potential, but it is hard to precisely judge its usefulness. We believe the highest priority now is to run many more tests and quantify its error. Once this is complete, the manual section 2.5 step must be replaced with something more automated. This algorithm would then be ready for general use. It is possible it will not return as many ship tracks as the model employed by (Yuan et al. [24]) which is driven by machine learning, but our method is able to function without the need for pre-training or massive datasets. Therefore, this has the potential to become an easier way for many climate scientists to gain access to cloud-aerosol interaction data. With the great significance of climate change our objective should always be to provide accessible data to as many people as possible.

Bibliography

[1] A. S. Ackerman, M. P. Kirkpatrick, D. E. Stevens, and O. B. Toon. The impact of humidity above stratiform clouds on indirect aerosol climate forcing. *Nature*, 432:1014–1017, 2004. doi:10.1038/nature03174. → page 1

[2] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010. doi:10.1109/CVPR.2010.5539960. → page 13

[3] O. Boucher, D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, P. Rasch, S. Satheesh, S. Sherwood, B. Stevens, and X. Zhang. *Clouds and Aerosols*, book section 7, page 571–658. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013. ISBN ISBN 978-1-107-66182-0. doi:10.1017/CBO9781107415324.016. URL www.climatechange2013.org. → page 1

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. → page 13

[5] C. S. Bretherton, P. N. Blossey, and J. Uchida. Cloud droplet sedimentation, entrainment efficiency, and subtropical stratocumulus albedo. *Geophysical Research Letters*, 34(3), 2007. doi:https://doi.org/10.1029/2006GL027648. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006GL027648. → page 1

[6] X. Chen, X. Wang, and J. Xuan. Tracking multiple moving objects using unscented kalman filtering techniques. *CoRR*, abs/1802.01235, 2018. URL http://arxiv.org/abs/1802.01235. → page 13

[7] Y.-C. Chen, M. W. Christensen, L. Xue, A. Sorooshian, G. L. Stephens, R. M. Rasmussen, and J. H. Seinfeld. Occurrence of lower cloud albedo in

ship tracks. *Atmospheric Chemistry and Physics*, 12(17):8223–8235, 2012. doi:10.5194/acp-12-8223-2012. URL https://acp.copernicus.org/articles/12/8223/2012/. → page 1

[8] J. A. Coakley Jr. and F. P. Bretherton. Cloud cover from high-resolution scanner data: Detecting and allowing for partially filled fields of view. *Journal of Geophysical Research: Oceans*, 87(C7):4917–4932, 1982. doi:https://doi.org/10.1029/JC087iC07p04917. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JC087iC07p04917. → page 6

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996. → page 6

[10] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In W. M. Wells, A. Colchester, and S. Delp, editors, *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98*, pages 130–137, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49563-5. → page 10

[11] C. Galamhos, J. Matas, and J. Kittler. Progressive probabilistic hough transform for line detection. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 554–560 Vol. 1, 1999. doi:10.1109/CVPR.1999.786993. → page 11

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 702–715, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33765-9. → page 13

[13] A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, and M. Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126(7):671–688, 07 2018. URL https://ezproxy.library.ubc.ca/login?url=https://www-proquest-com.ezproxy.library.ubc.ca/scholarly-journals/discriminative-correlation-filter-tracker-with/docview/1985775004/se-2?accountid=14656. Copyright - International Journal of Computer Vision is a copyright of Springer, (2018). All Rights Reserved; Last updated - 2018-05-25. → page 13

[14] E. Meijering, M. Jacob, J.-C. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*, 58A(2):167–176, 2004. doi:https://doi.org/10.1002/cyto.a.20022. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.20022. → page 10

[15] T. Nakajima and M. D. King. Determination of the optical thickness and effective particle radius of clouds from reflected solar radiation measurements. part i: Theory. *Journal of Atmospheric Sciences*, 47(15): 1878 – 1893, 01 Aug. 1990. doi:10.1175/1520-0469(1990)047⟨1878:DOTOTA⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/47/15/1520-0469_1990_047_1878_dotota_2_0_co_2.xml. → pages 16, 19

[16] C.-C. Ng, M. H. Yap, N. Costen, and B. Li. Automatic wrinkle detection using hybrid hessian filter. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *Computer Vision – ACCV 2014*, pages 609–622, Cham, 2015. Springer International Publishing. ISBN 978-3-319-16811-1. → page 10

[17] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017. → page 5

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011. → page 7

[19] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143 – 168, 1998. ISSN 1361-8415. doi:https://doi.org/10.1016/S1361-8415(98)80009-1. URL http://www.sciencedirect.com/science/article/pii/S1361841598800091. → page 10

[20] T. Schmit, M. Gun, G. Fu, T. Rink, K. Bah, W. Zhang, and W. Wolf. *GOES-R Advanced Baseline Imager (ABI) Algorithm Theoretical Basis Document For Cloud and Moisture Imagery Product (CMIP)*, 2012. URL https://www.star.nesdis.noaa.gov/goesr/docs/ATBD/Imagery.pdf. → page 5

[21] M. Szczodrak, P. H. Austin, and P. B. Krummel. Variability of optical depth and effective radius in marine stratocumulus clouds. *Journal of the*

Atmospheric Sciences, 58(19):2912 – 2926, 01 Oct. 2001. doi:10.1175/1520-0469(2001)058⟨2912:VOODAE⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/58/19/1520-0469_2001_058_2912_voodae_2.0.co_2.xml. → pages iv, 4, 16, 18, 19

[22] S. Thakur. Analysis of sobel edge detection technique for face recognition. 05 2015. → page 10

[23] A. Walther, W. Straka, and A. Heidinger. *ABI Algorithm Theoretical Basis Document For Daytime Cloud Optical and Microphysical Properties (DCOMP)*, 2013. URL https://www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_Cloud_DCOMP_v3.0_Jun2013.pdf. → pages iv, 16, 18, 19

[24] T. Yuan, C. Wang, H. Song, S. Platnick, K. Meyer, and L. Oreopoulos. Automatically finding ship tracks to enable large-scale analysis of aerosol-cloud interactions. *Geophysical Research Letters*, 46(13): 7726–7733, 2019. doi:https://doi.org/10.1029/2019GL083441. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL083441. → pages 2, 5, 24