UNIVERSITY OF CALIFORNIA, SAN DIEGO

Feature Extraction and Classification of Clouds in High Resolution Panchromatic Satellite Imagery

A Thesis submitted in partial satisfaction of

the requirements for the degree Master of Science

in

Electrical Engineering (Signal and Image Processing)

by

Elan Sharghi

Committee in charge:

Professor Truong Q. Nguyen, Chair
Professor Pamela C. Cosman
Professor Mohan M. Trivedi

2013

The Thesis of Elan Sharghi is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

DEDICATION

To my loving and supportive parents, Kim and Ali Sharghi.

TABLE OF CONTENTS

Signature Page iii

Dedication iv

Table of Contents v

List of Figures vi

List of Tables viii

Acknowledgements ix

Vita x

Abstract xi

1. Introduction and Motivation 1
    1.1 Dataset 1

2. Survey of Methods 3
    2.1 Texture Analysis 3
    2.2 Various Feature Extraction Methods 3

3. Feature Extraction Methods and Proposed Algorithm 5
    3.1 Texture Analysis 5
    3.2 Hough Transform 9
    3.3 Wavelet Analysis 11
        3.3.1 Image Rotation to Align Object Major Axis in the Horizontal Direction 12
        3.3.2 Wavelet Edge Image using Various Filter Lengths and Types 15
        3.3.3 Perform Thinning of Binary Image 17
        3.3.4 Image Reconstruction with Horizontal Detail Emphasis 19
        3.3.5 Edges of Clouds 21
    3.4 Final Algorithm 24

4. Simulations and Comparison of Results 25
    4.1 Gabor Results 25
    4.2 Pre-Filtering Dataset using Hough Transform 29
    4.3 SVM Classifiers 30
        4.3.1 Hough Line Angles and Data Normalization 31
        4.3.2 Weight Factor 32
    4.4 Hough Transform Line Detection Length 33

5. Conclusion 34
    5.1 Summary of Contributions 34
    5.2 Future Work 34

    Bibliography 36

LIST OF FIGURES

Figure 1. Test dataset separated by the following four classes: cloud, ship, fast moving ship, and other. 2
Figure 2. Examples of a small cloud object (left) and ship object (right) removed from the experimental dataset. 2
Figure 3. Real components of Gabor wavelet filters at various orientation and scales. 6
Figure 4. Sample cloud image and Gabor convolved images at various frequency and orientation. 7
Figure 5. Sample ship image and Gabor convolved images at various frequency and orientation. 8
Figure 6. Lines detected from sample ship image (left) extracted from Canny edge image (right). 10
Figure 7. Lines detected from sample cloud image (left) extracted from Canny edge image (right). 10
Figure 8. Example where Hough transform line detection fails to identify parallel line segments containing a ship image. 11
Figure 9. Structure for wavelet decomposition. 12
Figure 10. Comparison of wavelet decomposition images at level one for a rotated and non-rotated image. Upper left image represents the original image processed, and the remaining three images are the details following the structure in Figure 9. 13
Figure 11. Comparison of binarized wavelet decompositions at level one for a rotated (top figure) and non-rotated image (bottom figure). The upper left images are computed using an OR operation on the three detailed binary images. 14
Figure 12. Median filtered and rotated test ship image. 15
Figure 13. Level three thresholded detail coefficients for various length filter of image displayed in Figure 12. The upper left images are computed using an OR operation on the three detailed binary images. 16
Figure 14. Median filtered and rotated test image. 17
Figure 15. Edge detection using wavelets and other commonly used edge detector. 18
Figure 16. Example comparing edge map computed using Haar wavelet (bottom left) and Sobel (bottom right) edge detector. 19
Figure 17. Median filtered and rotated test image. 20
Figure 18. Prewitt edge detection applied to the median filtered image in Figure 17 (top) versus the reconstructed horizontal detail emphasized image (bottom). 20
Figure 19. Level three wavelet threshold image (right) and its corresponding lines detected using the Hough transform (left). 21
Figure 20. Level three wavelet threshold image (right) and its corresponding lines detected using the Hough transform (left). 22
Figure 21. Hough lines extracted from Canny edge map for a cloud (top) and ship (bottom) image. 23
Figure 22. Flow diagram of cloud filtering algorithm. 24

Figure 23. Gabor convolved cloud image ranging in four different frequency and four different orientations. 26
Figure 24. Cloud images mislabeled as ships through the Hough transform approach............. 30
Figure 25. Example of ship objects obscured by clouds.. 35

LIST OF TABLES

Table 1. Classification accuracy using 12 correlation coefficients and KNN Euclidean distance classifier. 26

Table 2. Classification accuracy using four correlation coefficients and KNN Euclidean distance classifier. 27

Table 3. Classification accuracy using 16 correlation coefficients and KNN Euclidean distance classifier. 27

Table 4. Classification accuracy using 24 correlation coefficients and KNN Euclidean distance classifier. 28

Table 5. Classification accuracy using 20 correlation coefficients and KNN Euclidean distance classifier. 29

Table 6. Modifications to the KNN distance measure using the twenty correlation coefficients shown in Table 5........ 29

Table 7. Classification accuracy using 20 Gabor correlation coefficients, KNN classifier, and pre-filtering the dataset using line extraction through Hough transform. Edge images are computed using both Canny and Haar wavelet approach. 30

Table 8. Classification accuracy using SVM classifier and Standard Score normalization of training and testing dataset........ 31

Table 9. Classification accuracy using SVM classifier and maximum normalization of training and testing dataset. 32

Table 10. Classification accuracy using the SVM classifier with a 1:100 weighting factor to place high priority in ship classification accuracy. 32

Table 11. SVM classifier with maximum Hough transform line length added as additional feature to SVM classifier........ 33

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Truong Nguyen, for his guidance on my research. Secondly, I would like to thank my co-worker Shibin Parameswaran from the Space and Naval Warfare Systems Center Pacific (SSC PAC) for sharing his knowledge in image classification and machine learning. Last, I would like to thank Heidi Buck and Dr. Katie Rainey, my branch head and project lead respectively from SSC PAC, for providing the research topic for this thesis. I am grateful to have them review the content of this paper and provide valuable suggestions to clarify the materials discussed.

VITA

2005	Bachelor of Science, University of California, San Diego Department of Electrical and Computer Engineering
2006-PRESENT	Electrical Engineer, Space and Naval Warfare Systems Center, Pacific
2013	Master of Science, University of California, San Diego Department of Electrical and Computer Engineering

PUBLICATIONS

Bagnall, B., Sharghi, E., and Buck, H. “Algorithms for the Detection and Mapping of Wildfires in SPOT 4 and 5 Imagery.” *Society of Photo-Optical Instrumentation Engineering (SPIE), Society Proceedings, Imaging Spectrometry XVII*, vol. 8515, 2012.

Buck, H., Sharghi, E., Guilas, G., et al. “Enhanced Ship Detection from Overhead Imagery.” Society *of Photo-Optical Instrumentation Engineering (SPIE), Society Proceedings, Optics and Photonics in Global Homeland Security IV*, vol. 6696, 2008.

Buck, H., Sharghi, E., Bromley, K., et al. “Ship Detection and Classification from Overhead Imagery.” *Society of Photo-Optical Instrumentation Engineering (SPIE), Society Proceedings, Applications of Digital Image Processing*, vol. 6696, 2007.

PATENTS AWARDED

2013	“Method for Detecting and Mapping Fires using Features Extracted from Overhead Imagery,” Patent 8,369,567.
2012	“Ship Detection System and Method for Overhead Images,” Patent 8,116,522.
2011	“System and Method for Enhancing Low-Visibility,” Patent 8,023,760.

ABSTRACT OF THE THESIS

Feature Extraction and Classification of Clouds in High Resolution Panchromatic Satellite Imagery

By

Elan Sharghi
Master of Science in Electrical Engineering (Signal and Image Processing)
University of California, San Diego 2013
Professor Truong Q. Nguyen, Chair

The development of sophisticated remote sensing sensors is rapidly increasing, and the vast amount of satellite imagery collected is too much to be analyzed manually by a human image analyst. It has become necessary for a tool to be developed to automate the job of an image analyst. This tool would need to intelligently detect and classify objects of interest through computer vision algorithms. Existing software called the Rapid Image Exploitation Resource (RAPIER®) was designed by engineers at Space and Naval Warfare Systems Center Pacific (SSC PAC) to perform exactly this function. This software automatically searches for anomalies in the ocean and reports the detections as a possible ship object. However, if the image contains a high percentage of cloud coverage, a high number of false positives are triggered by the clouds.

The focus of this thesis is to explore various feature extraction and classification methods to accurately distinguish clouds from ship objects. An examination of a texture analysis method, line detection using the Hough transform, and edge detection using wavelets are explored as possible feature extraction methods. The features are then supplied to a K-Nearest Neighbors (KNN) or Support Vector Machine (SVM) classifier. Parameter options for these classifiers are explored and the optimal parameters are determined.

Chapter 1

Introduction and Motivation

In the defense community, software has been developed to detect and classify vessels in high resolution panchromatic satellite imagery. At SSC PAC, a group of engineers have developed sophisticated software to search for objects of interest in satellite imagery. This software framework is called RAPIER®, the Rapid Image Exploitation Resource.

In order to detect ships, RAPIER® masks out land and searches for anomalies on the ocean surface. The software then reports the detections to the end user by saving the objects of interest as image chips, and loads these files into HyperText Markup Language (HTML) and Keyhole Markup Language (KML) files. One of the limitations of processing panchromatic imagery (visible spectrum imagery), from sensors such as Quickbird2, IKONOS, and Worldview 1 and 2, is the significant presence of clouds. This problem does not occur when processing synthetic-aperture radar (SAR) imagery as SAR imagers have the ability to penetrate through clouds, which captures data in the microwave electromagnetic spectrum [6].

Clouds can often trigger as anomalies by RAPIER® when processing panchromatic satellite imagery, which results in having a high false alarm rate and increasing processing time. The focus of this thesis is to devise a method which will accurately distinguish clouds from ships in high resolution panchromatic imagery.

1.1 Dataset

The dataset used in this research project is comprised of image chips output from the RAPIER® software. These images were produced from processing high resolution commercial panchromatic satellite imagery, such as IKONOS, Quickbird2, and Worldview 1 and 2. Their resolution ranges from 0.5m to 1m when captured at nadir angle. For comparison consistency, we are using the same dataset set described by researchers of [1].

The dataset is made up of 3709 grayscale images, and is separated into the following four classes: cloud, ship, fast moving ship, and other. Examples from each of these classes are shown below in Figure 1. There are 1037 cloud images, 770 ship images, 487 fast moving ship images, and 1415 images classified as other. The first two classes, cloud and ship, will be used for our research, however, results reported in [1] use all four classes.

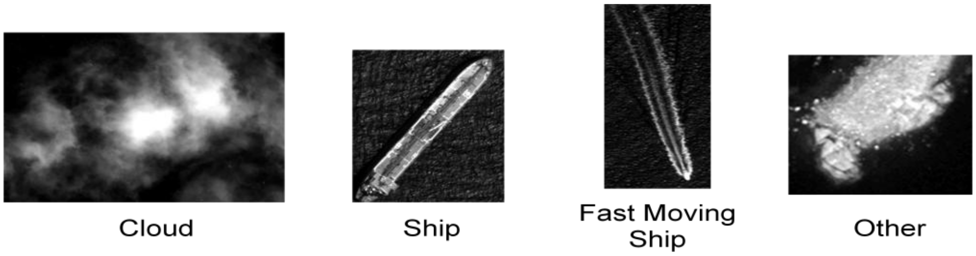


Figure 1. Test dataset separated by the following four classes: cloud, ship, fast moving ship, and other.

Distinguishing small vessels and small popcorn clouds is an extremely challenging task, even for a human observer. Examples are shown below in Figure 2 of the problematic image chips. Therefore, the dataset is pre-filtered to contain only large objects. The final dataset consists of 955 cloud images and 452 ship images.

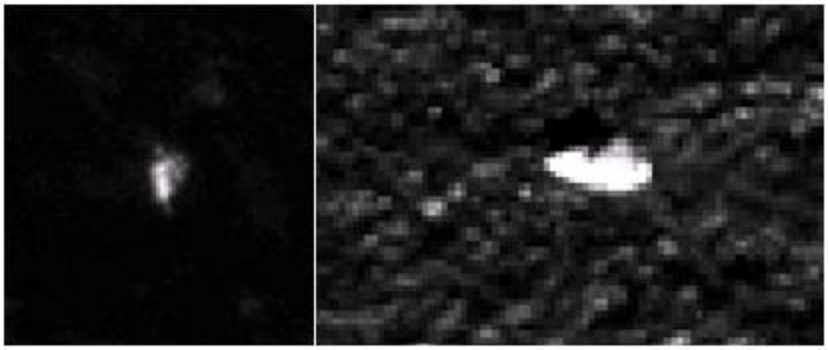


Figure 2. Examples of a small cloud object (left) and ship object (right) removed from the experimental dataset.

Chapter 2

Survey of Methods

2.1 Texture Analysis

Several researchers have explored texture analysis techniques such as Gabor wavelet transforms and Gray Level Co-occurrence Matrices (GLCM) as cloud feature extraction methods [3][5]. These techniques have been applied to low resolution satellite imagery, such as meteorological satellite cloud imagery, to determine the cloud coverage and type [3][5]. It has been concluded in these publications that the use of Gabor wavelets achieves higher classification rates than GLCM for the application of cloud type classification. However, the authors have stated that feature extraction solely based on texture analysis does not yield impressive results. The accuracy for the Gabor wavelets spans from 81% to 92%, and 76 to 88% for GLCM, based on the number of test images used for the purpose of classifying cloud types in low resolution meteorological satellite imagery [3].

2.2 Various Feature Extraction Methods

The issue of high cloud presence has been an ongoing problem for RAPIER® since 2007 [2]. Engineers from Tomnod Inc. were contracted by SSC PAC to research various cloud feature extraction methods to reduce the false alarm rate of RAPIER®. Their approach was to take image chips produced by RAPIER®, and classify the results into one of the four categories: ship, cloud, fast moving ship, and other. Their research results are summarized in [1]. The paper focuses on two aspects: extracting features and applying supervised classification on these features.

The feature extraction techniques discussed in the paper include Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), texture-based features, and ship-specific features [1]. These techniques were tuned to identify specific features in an image, including edges and gradients, rotation and scale-invariant keypoints, and textures and patterns that are specific to ship-like objects. SIFT is used to identify keypoints on an object that are invariant to the object's location, scale, orientation and illumination. SURF is an extension of SIFT, with the exception that it uses Haar wavelets for its keypoint detection. HOG describes the object based on its distribution of edges and gradients in a local region of the image. Out of all the feature extraction techniques surveyed, the authors of [1] concluded that SURF descriptors in combination with Support Vector Machine (SVM) classifiers produced the highest accuracy of classification. The various feature extraction methods applied achieved 81-94% accuracy for the non-cloud class and 71-91% on the cloud class.

Chapter 3

Feature Extraction Methods and Proposed Algorithm

In this Chapter, we investigated methods to extract cloud features in high resolution panchromatic satellite imagery. We begin by investigating Gabor wavelets for image texture extraction, a method used by other researchers in the field [3][5]. The correlation coefficients were used to create features from our Gabor filtered images. The second method uses the Hough transform to extract lines from our test image. We expect images from the ship dataset to contain parallel lines indicative of the position of the ship's hull, and have developed code to extract these features. The last technique involves computing edge maps using wavelets decompositions. This edge image is used by the Hough transform to extract parallel line features. Classical edge detection techniques, such as Sobel, Prewitt, and Canny, perform poorly when processing images with low contrast, high amounts of ocean glint, and trailing wakes. The wavelet approach was investigated to determine a technique to suppress noise and enhance ship edge features. A final algorithm was developed using a combination of feature extraction methods we found useful in our research.

3.1 Texture Analysis

When trying to distinguish a cloud image chip from a ship, we notice that the textures of the two objects are very different. Therefore, we explore different computer vision texture measure approaches to use as feature descriptors for our classifier.

The Gabor wavelet filter can be used to extract texture information from an image. The 2D Gabor wavelet $\psi(x,y)$ is defined by the equations listed below [6].

$$\psi(x,y;f,\theta) = \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}x'^2+\frac{f^2}{\eta^2}y'^2\right)} e^{-i2\pi f x'}, \quad (1)$$

$$x' = x\cos\theta + y\sin\theta,$$

$$y' = -x\sin\theta + y\cos\theta.$$

In the equations above, f is the frequency of the sinusoidal plane, θ is the clockwise rotation of the Gaussian envelope, γ is the spatial width of the filter alone the plane wave, and η is the spatial width perpendicular to the wave.

In the experiments that we have conducted, θ will varies from 0, $\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$, and the sinusoidal plane frequency f varies from $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$. A frequency representation of the different wavelet combinations used is shown below in Figure 3.

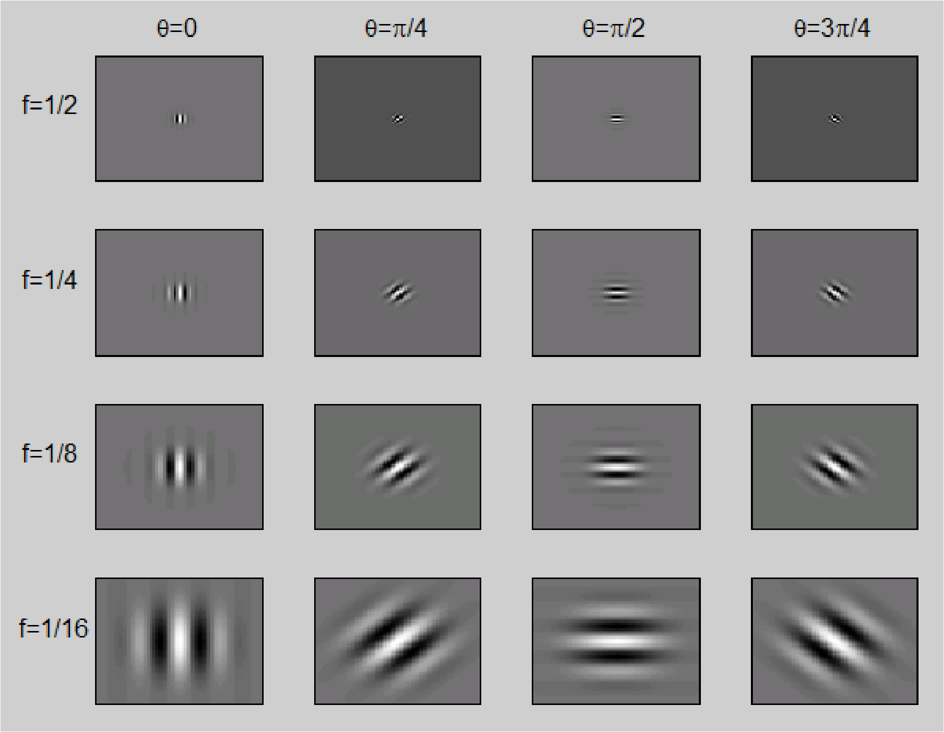


Figure 3. Real components of Gabor wavelet filters at various orientation and scales.

The set of 16 Gabor filters are applied to a sample cloud and ship image shown in Figure 4 and Figure 5 respectively. There is an obvious distinction between the convolved ship and cloud filtered images. In the sample cloud image, using the various filter frequency values, very few features of the object are extracted. It is only at the highest frequency value of the Gabor filter that we are able to extract cloud contours. In contrast, notice in the ship images shows extracted ship contours on the full range of frequency values tested.

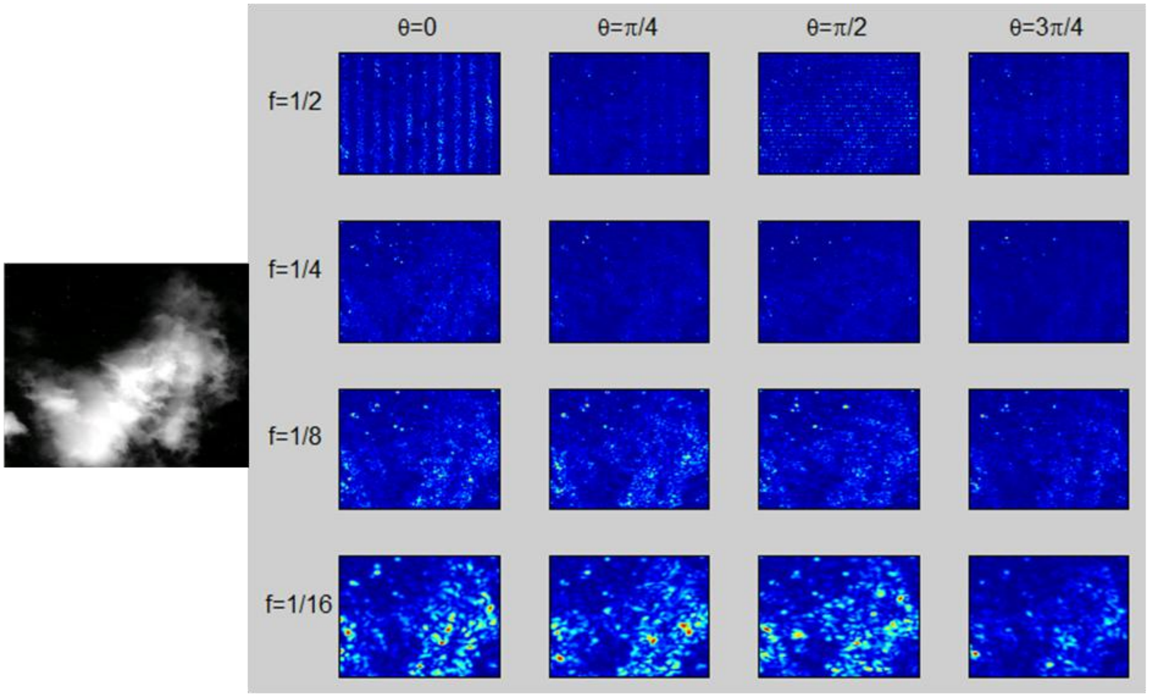


Figure 4. Sample cloud image and Gabor convolved images at various frequency and orientation.

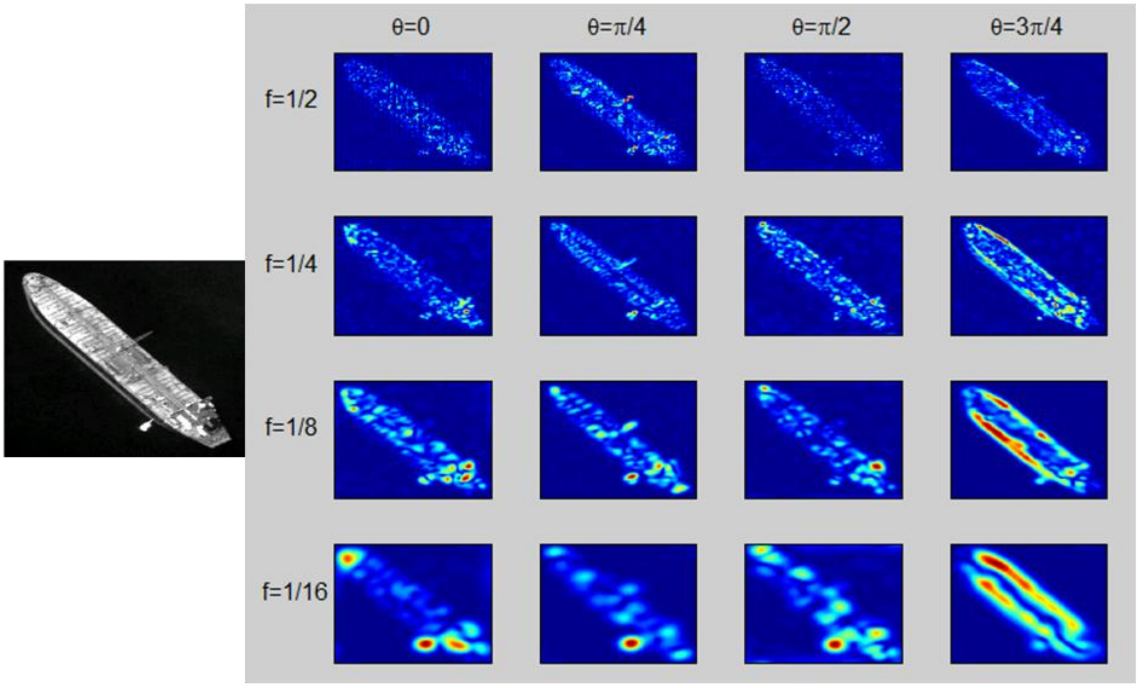


Figure 5. Sample ship image and Gabor convolved images at various frequency and orientation.

For every wavelet angle θ, we compute the normalized correlation coefficient C, defined below, for various combinations of frequency values, where T(x,y) represents the template image, and I(x,y) represents the test image.

$$\mathrm{C}(\mathrm{x},\mathrm{y}) = \frac{\sum_{\mathrm{x}',\mathrm{y}'}\left[\mathrm{T}(\mathrm{x}',\mathrm{y}')\cdot\mathrm{I}(\mathrm{x}+\mathrm{x}',\mathrm{y}+\mathrm{y}')\right]^2}{\sqrt{\sum_{\mathrm{x}',\mathrm{y}'}\mathrm{T}(\mathrm{x}',\mathrm{y}')^2\cdot\sum_{\mathrm{x}',\mathrm{y}'}\mathrm{I}(\mathrm{x}+\mathrm{x}',\mathrm{y}+\mathrm{x}')^2}} \qquad (2)$$

The following wavelet frequency combinations listed below are computed for f. Their correlation coefficients are computed amongst their corresponding filtered images. This will give us 20 correlation coefficient values, which we will use as part of our feature vector.

f = 1/2 vs. 1/4
f = 1/2 vs. 1/16
f = 1/4 vs. 1/8
f = 1/4 vs. 1/16
f = 1/8 vs. 1/16

We expect correlation values calculated from the ship dataset to be higher than those calculated from the cloud dataset. The high correlation in the ship images is a result of the high amounts of texture extracted at various frequency levels, as shown in Figure 5. In contrast, the cloud images typically extract texture information only at high frequency values. This allows us to separate the two classes. However, our experimental results summarized in Chapter 4.1 shows these features alone will not be sufficient to reliably separate the two classes. Therefore, we will continue to investigate additional features.

3.2 Hough Transform

The sides of large ships form straight edges. If you compute the edge map of a ship image, the straight side edges can be identified using the Hough transform. The shape of a ship is a unique feature which allows us to distinguish it from a cloud object. If we can accurately identify the two long parallel lines in our image chip, we can pre-label the image as containing a ship.

The Hough transform is a feature extraction technique that is used to extract line segments in an image [4]. This technique takes a binary edge image as an input, and extracts line segments that meet a certain pixel length threshold, as shown below in Figure 6. In the sample ship and cloud images shown in Figure 6 and Figure 7 respectively, the Canny edge detector was used to compute the edge map of the images (right images) and the Hough transform was computed (left Images). The Matlab implementation of the Canny edge detector was used with ideal thresholds selected by the software, and sigma set at $\sqrt{2}$. We explore different edge extraction techniques in the following discussion about wavelets.

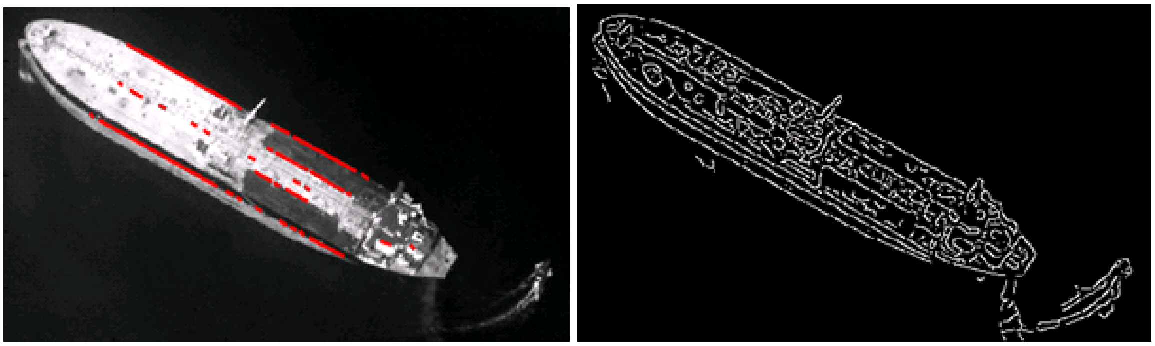


Figure 6. Lines detected from sample ship image (left) extracted from Canny edge image (right).

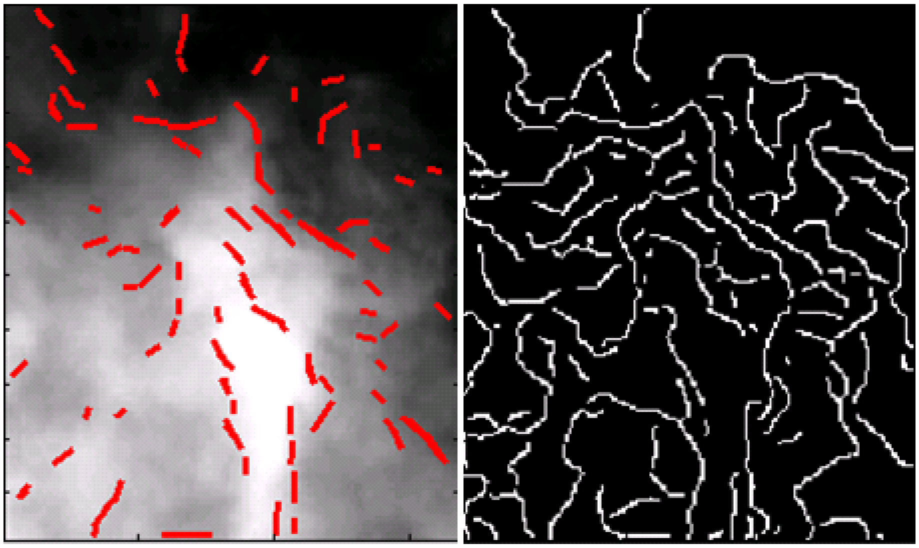


Figure 7. Lines detected from sample cloud image (left) extracted from Canny edge image (right).

Although there are several line segments detected in the sample cloud image shown in Figure 7, the orientations of the lines are widely distributed. In contrast, we see the lines detected in the sample ship image in Figure 6 are approximately parallel. Therefore, we identify parallel lines in the image by calculating the standard deviation of detected line angles and selecting images below a set threshold. The image chip is then pre-labeled with high confidence as containing a ship, and no further processing is needed. In addition to pre-labeling our dataset for image chips containing parallel lines, we can use the line lengths and line angle distributions as feature descriptors for our classifier.

The limitation of performing line detection using the Hough transform is that the method relies on processing an accurate edge image. If an accurate edge image is not obtained, the Hough transform will perform poorly in extracting the parallel edges of a ship. As shown in Figure 8, poor contour extraction of a ship is caused by an image containing ocean waves at high levels, often as a result of high wind speed. These conditions are characterized as having a high sea state. Other contributors to poor contour extraction include low contrast in images and ships with trailing wakes. Therefore, we explore wavelet decomposition techniques to suppress noise and enhance edge detection for ship feature extraction.

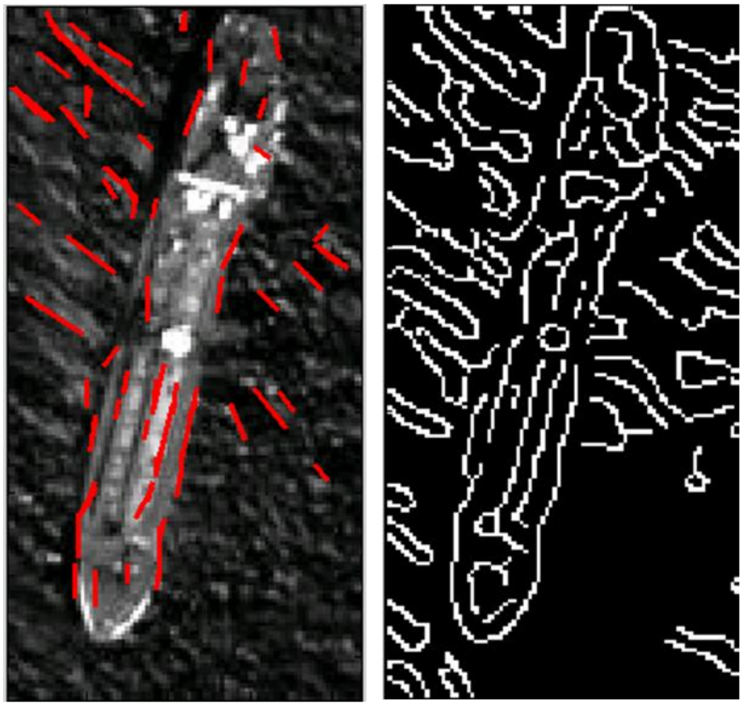


Figure 8. Example where Hough transform line detection fails to identify parallel line segments containing a ship image.

3.3 Wavelet Analysis

The wavelet decomposition approach was investigated as a pre-processing stage in order to improve the edge image. In order to compute the edge image, a three-level wavelet decomposition is obtained using the structure shown below in Figure 9. For each level, the high frequency detail sub-band images HL, LH, and HH are used. These images are computed using a combination of highpass and lowpass filtering. They extract horizontal, vertical, and diagonal details respectively.

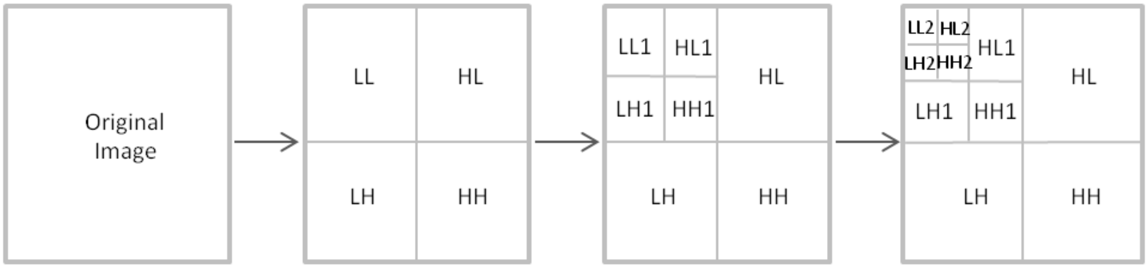


Figure 9. Structure for wavelet decomposition.

3.3.1 Image Rotation to Align Object Major Axis in the Horizontal Direction

When performing the wavelet thresholding operations, it has come to our attention that it is difficult to capture edge details for some images, especially at the level one decomposition. We speculate that if the ship is oriented such that its length is oriented horizontally, more details of the ship edges would be captured in the horizontal detailed image (HL). In Figure 10, a comparison is made between the level one decomposition images of a rotated and non-rotated image. The figure consists of two set of images. For each set of four images shown in Figure 10, the upper left image represents the original image processed, and the remaining three images are the detail images following the display convention shown in Figure 9.

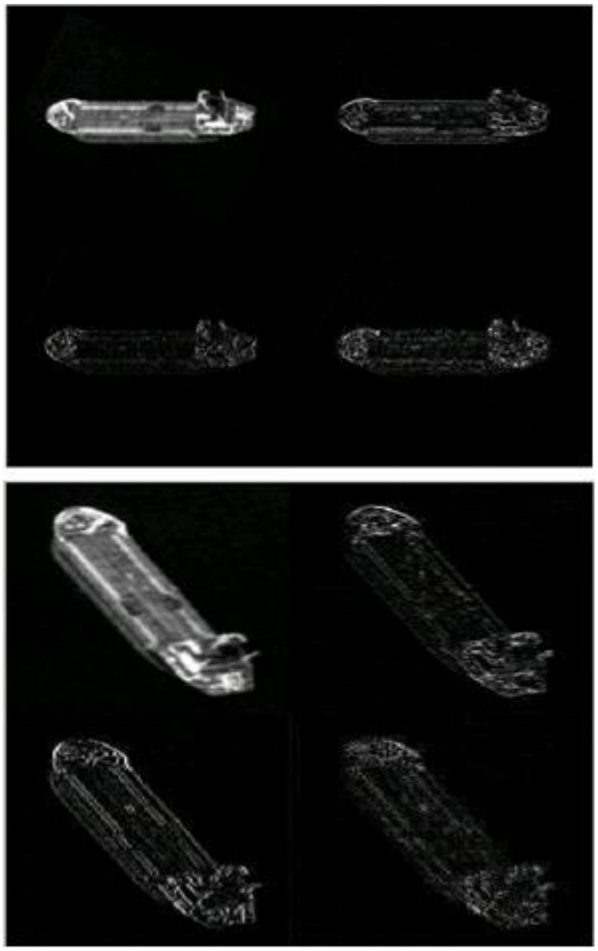


Figure 10. Comparison of wavelet decomposition images at level one for a rotated and non-rotated image. Upper left image represents the original image processed, and the remaining three images are the details following the structure in Figure 9.

In order to binarize the detailed component images shown above, we set the value of our binary image to 1 for pixels above the defined threshold, and 0 for pixels below the threshold. The thresholds, defined below, are dependent on the mean, E[], and standard deviation, σ, of the detailed image. The three binary detailed images are then thresholded using the following equations:

$$\begin{gathered} T\_HL = E[HL] + 1.5\ \sigma_{HL} \\ T\_LH = E[LH] + 3.0\ \sigma_{LH} \\ T\_HH = E[HH] + 3.0\ \sigma_{HH} \end{gathered} \tag{3}$$

Notice that stricter parameters are set on the vertical and diagonal details, and looser parameters are set for the horizontal details. This is done in order to capture more edges in the horizontal direction. For the experiments described below, we perform three-level wavelet decompositions and use the thresholds stated above.

The corresponding binarized edge images are shown below in Figure 11. These images are thresholded using the threshold values defined in Equation (3). The figure contains two sets of binarized edge images. For each of the set of four images shown in Figure 11, the upper left image is computed using a binary OR operation on the three detailed binary images.

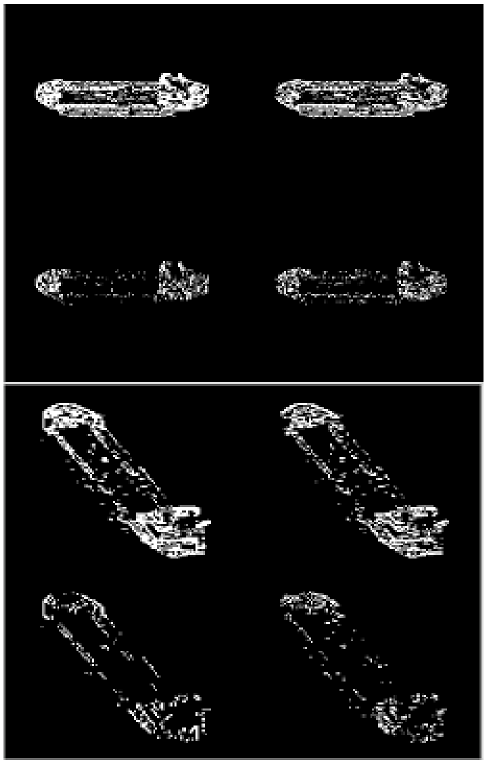


Figure 11. Comparison of binarized wavelet decompositions at level one for a rotated (top figure) and non-rotated image (bottom figure). The upper left images are computed using an OR operation on the three detailed binary images.

If we examine the upper left image from Figure 11 from both the rotated and non-rotated images, we see that the rotated image performs better in capturing the ship's contour. This is a result of orienting the ship such that the ship sides are extracted from the horizontal detailed image. We therefore loosen the threshold parameter of the horizontal detailed image to obtain a better edge image. For the remainder of the wavelet experiments, all images are rotated such that the major axis of the detected object is oriented horizontally.

The image rotation is an automated process. To compute the object orientation, the image is binarized by thresholding the image. The threshold is determined by computing two standard deviations above the image mean. The largest detected binary object is used to determine the object orientation, and the Matlab function imrotate was used to rotate the object horizontally.

3.3.2 Wavelet Edge Image using Various Filter Lengths and Types

For the following experiment, we use the test image shown below in Figure 12. This image consist of a ship oriented horizontally and has been preprocessed with a 5x5 median filter for noise reduction. The Matlab function medfilt2, from the Image Processing Toolbox, is used to perform this operation.

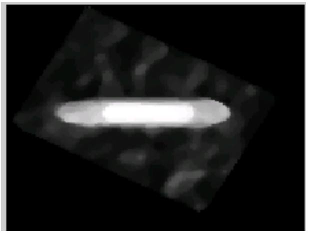


Figure 12. Median filtered and rotated test ship image.

The image in Figure 12 has been decomposed using various wavelet filter lengths and types. The result for the thresholded image at level three wavelet decomposition is shown below in Figure 13. Notice the Haar filter produces the best edge image. The results also show the Daubechie wavelet applied at various filter order. As the order of the filter increases, the performance of the edge detector degrades.

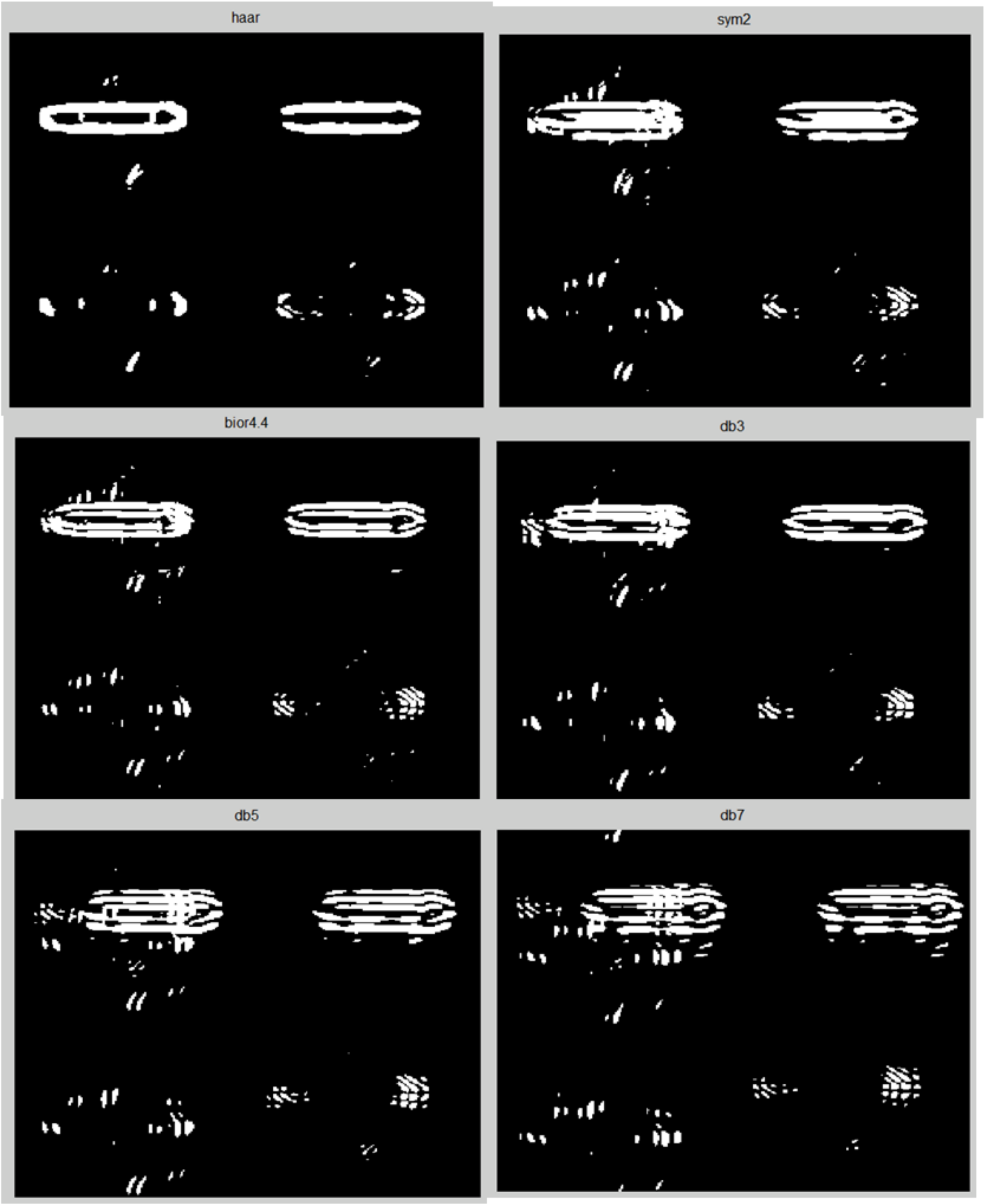


Figure 13. Level three thresholded detail coefficients for various length filter of image displayed in Figure 12. The upper left images are computed using an OR operation on the three detailed binary images.

The remainder of the experiments are conducted using Haar wavelets since it produces the best edge detection for our application.

3.3.3 Perform Thinning of Binary Image

The most visibly appealing edge map is obtained from the level three thresholded image. However, the edges are very thick. To obtain a true edge map, we perform thinning to the binary image, and compare the results to commonly used edge detectors such as Canny, Sobel, and Prewitt. For this experiment, we process the image shown below in Figure 14.

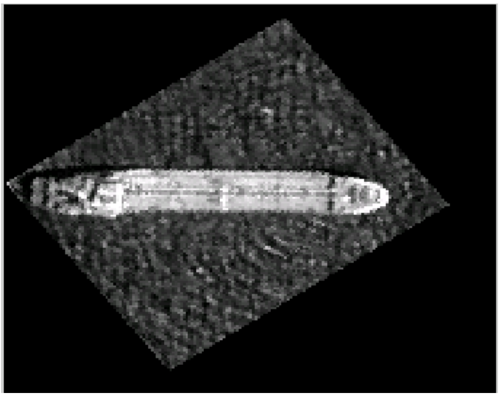


Figure 14. Median filtered and rotated test image.

The wavelet edge image is computed by taking the level three thresholded image, skeletonizing the image, and pruning the binary image over ten iterations using the Matlab implementation bwmorph. The results for the wavelet edge image and other commonly used edge detectors are shown below in Figure 15. Notice that Prewitt and Sobel provide the most visually appealing edge map. In this example, the result for the wavelet decomposition edge map is comparable to Sobel, but no improvement is seen. The Canny edge detector is sensitive to background noise, and picks up edges caused by ocean glint. Sigma for the Canny edge detector is set at $\sqrt{2}$.

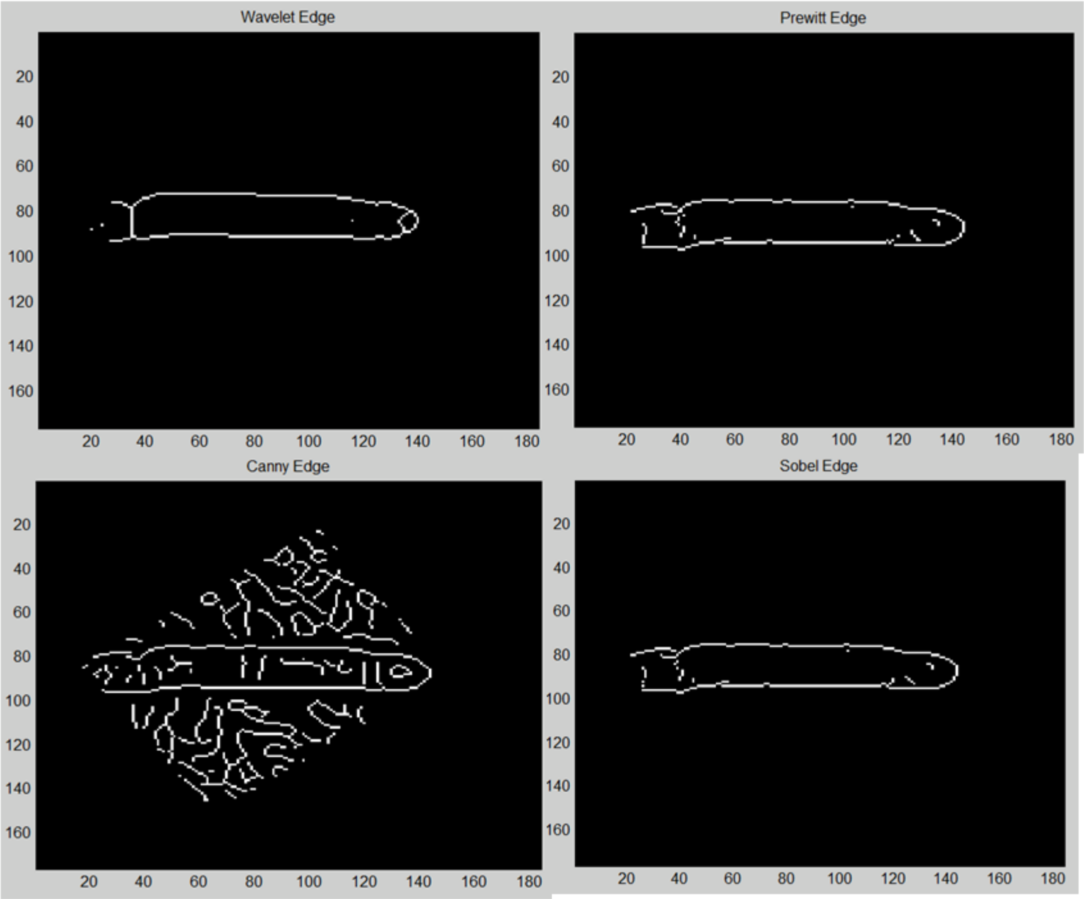


Figure 15. Edge detection using wavelets and other commonly used edge detector.

Below in Figure 16 is another example of a computed edge map. Notice in this example, the wavelet results performs more poorly than Sobel. One important detail to note is the edges of the ships detected using the wavelet method does not produce a straight line segment. This will produce poor line detection results when implementing the Hough transform.

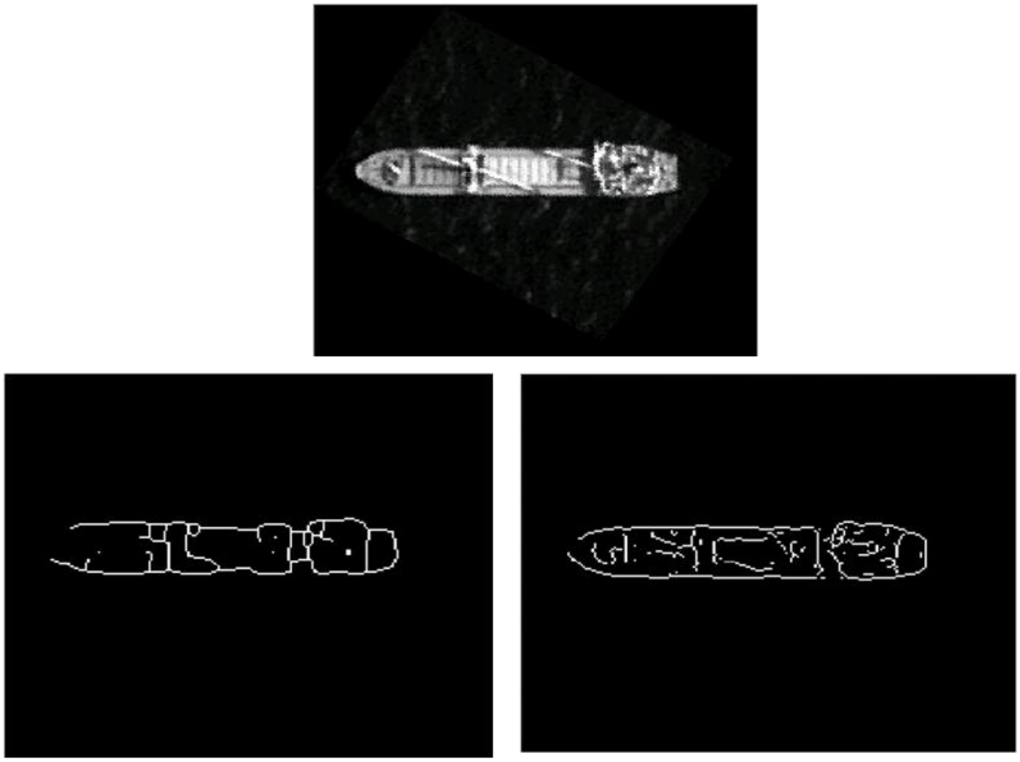


Figure 16. Example comparing edge map computed using Haar wavelet (bottom left) and Sobel (bottom right) edge detector.

3.3.4 Image Reconstruction with Horizontal Detail Emphasis

Since the Prewitt edge detector clearly outperforms the wavelet edge image approach, it is pointless to further pursue wavelet edge approach for this application. The next approach we take is to use wavelet decomposition to decompose the image, multiply the detail component of the image with a factor α_1, α_2, α_3 (where $\alpha_1 > \alpha_2$ and $\alpha_1 > \alpha_3$), and reconstruct the image. The α_1 scaling factor is for the horizontal detail component, α_2 is for the diagonal detail, and α_3 is for the vertical detail. The image is then reconstructed containing emphasis on the horizontal detail component. For this example, we process the image shown below in Figure 17.

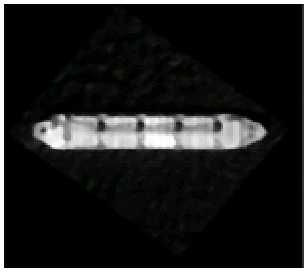


Figure 17. Median filtered and rotated test image.

The edge images obtain from using Prewitt edge detector on the original image versus using the reconstructed image are shown below in Figure 18.

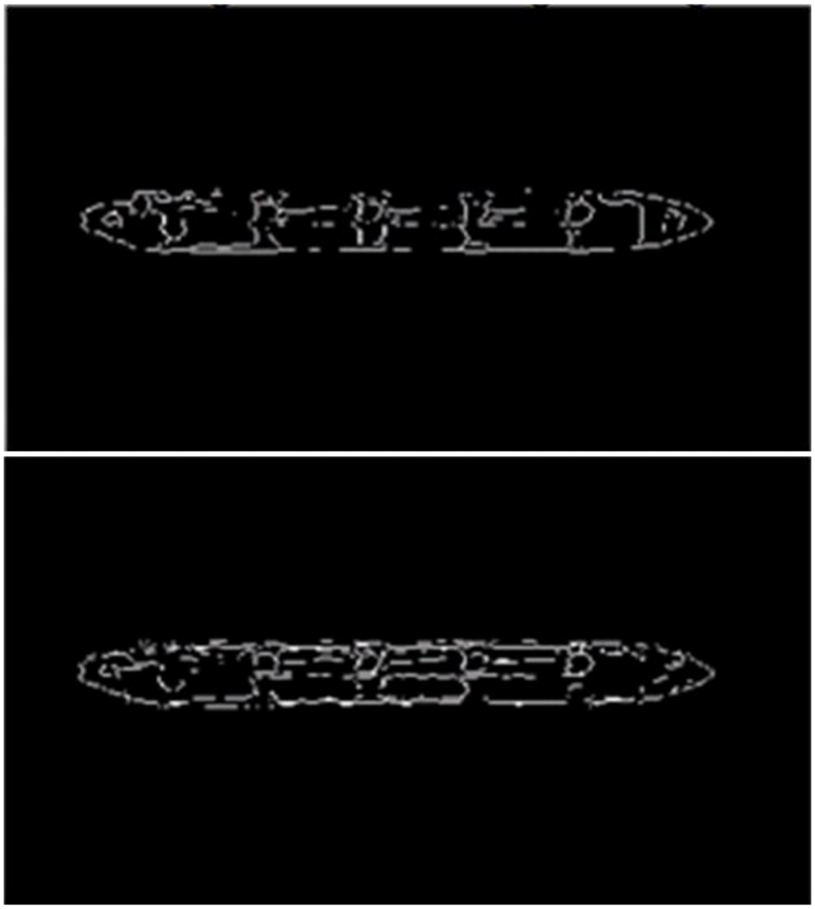


Figure 18. Prewitt edge detection applied to the median filtered image in Figure 17 (top) versus the reconstructed horizontal detail emphasized image (bottom).

Notice that more edge segments from the side of the ship image are detected using the reconstructed images.

3.3.5 Edges of Clouds

So far, all of our experiments have been focused on finding edge images of ships. We have not displayed any results from our cloud dataset. It is immediately discovered that using the level three wavelet thresholded image performs poorly in extracting edge contours of a cloud. An example is shown below in Figure 19. As a reminder, the standard deviation of line angles detected from the Hough transform is used to determine is the object is a ship or cloud. Ship objects will have several parallel lines, and therefore a low standard deviation, while clouds will have high standard deviation. However, the standard deviation in the example shown below is not very high because a poor edge map of the cloud object was detected.

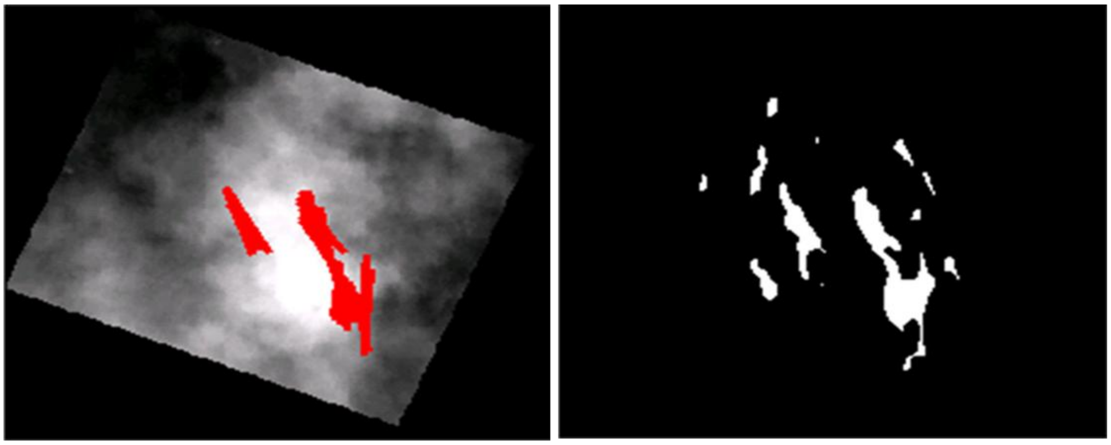


Figure 19. Level three wavelet threshold image (right) and its corresponding lines detected using the Hough transform (left).

In contrast to the cloud dataset, as we have seen earlier, the level three wavelet threshold image does an excellent job in extracting a ship's contour. We can then apply the Hough transform to the image to identify the side edges of the ship, as shown below in Figure 20.

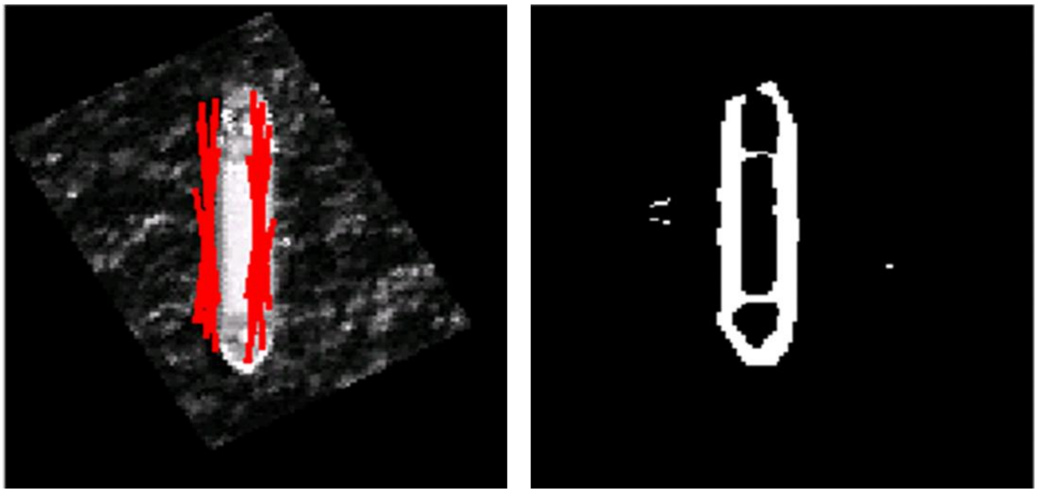


Figure 20. Level three wavelet threshold image (right) and its corresponding lines detected using the Hough transform (left).

Prior to experimenting with wavelet edge extraction methods, we applied the Hough transform to the Canny edge map of the image. Examples of a cloud and ship line extraction result using Canny edge is shown below in Figure 21.

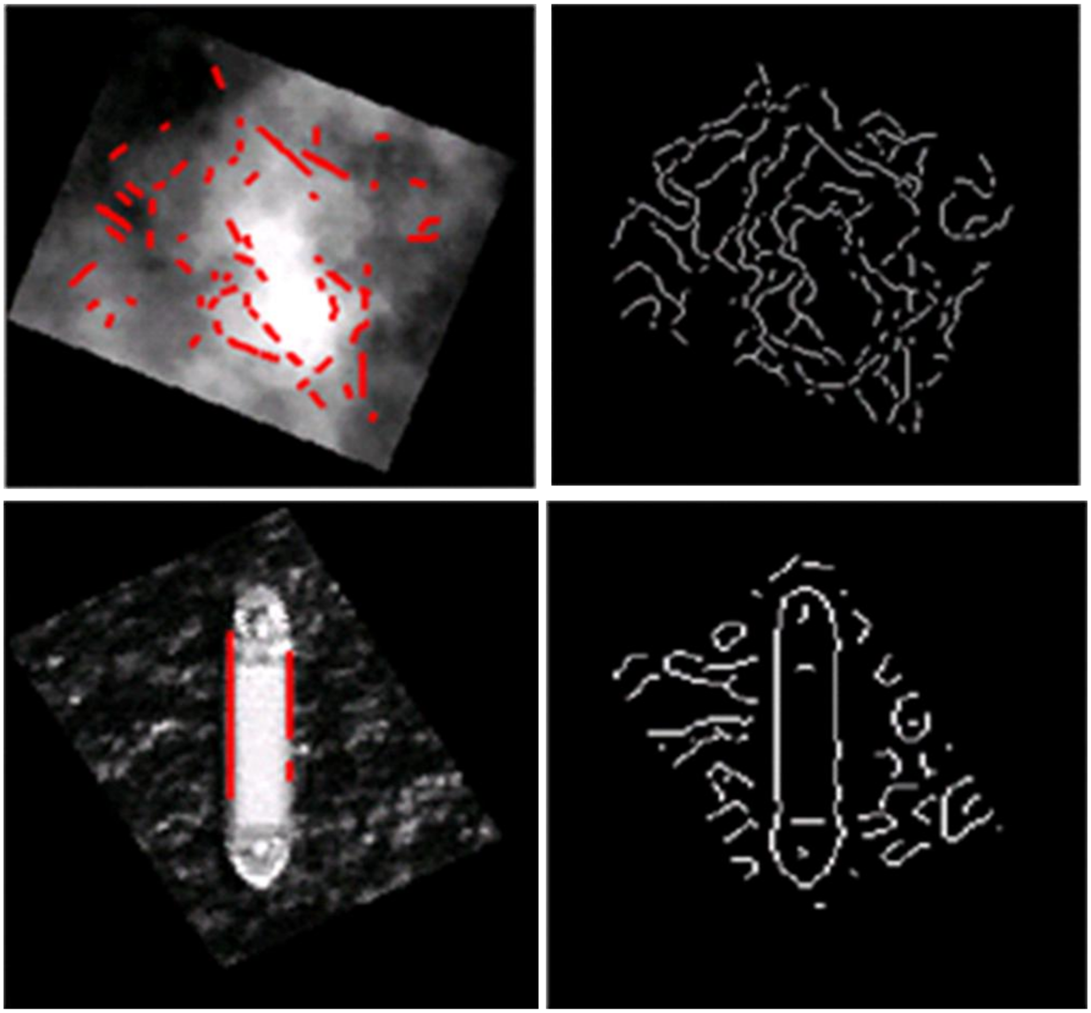


Figure 21. Hough lines extracted from Canny edge map for a cloud (top) and ship (bottom) image.

In contrast to the wavelet edge detection approach, Canny edge detection performs well in extracting details from a cloud image. This method allows us to extract several line segments and recognize the line angles are oriented in various directions. We can then confidently label the image as containing a cloud object. However, the downfall of using a Canny edge detector is its sensitivity to extracting background noise. We see this in the ship example in Figure 21.

We clearly see that both the Canny edge and Haar wavelet edge approaches have their advantages and disadvantages. Therefore, both methods will be used to determine if parallel lines are present in an image in order classify ship objects.

3.4 Final Algorithm

A final algorithm has been created to use all feature extraction methods discussed in this chapter. A flow chart shown in Figure 22 summarizes the feature extraction and classification algorithm. The method requires that edge detection is performed using both Canny and Haar wavelets. Parallel line detection using Hough transform is performed on both edge images, and the image under test is either labeled as containing ships or further processing is done. If further processing is needed, the Gabor wavelets are used to filter the image, and correlation coefficients among the frequency combinations of the filtered images are calculated. These 20 features are then supplied to a classifier to determine the image class.

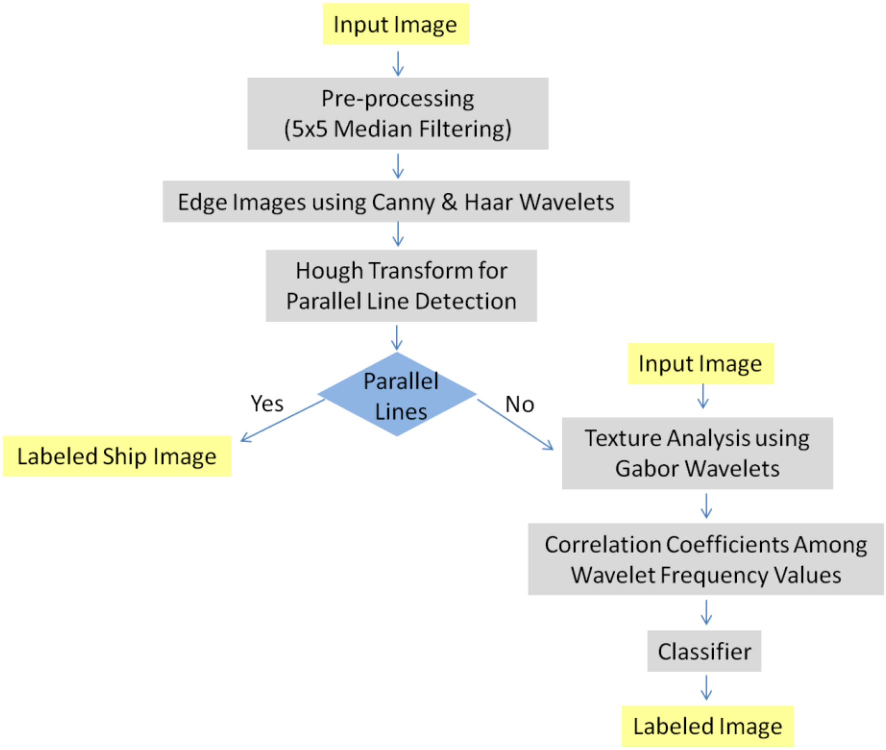


Figure 22. Flow diagram of cloud filtering algorithm.

Chapter 4

Simulations and Comparison of Results

In the previous Chapter, we discussed several feature extraction techniques. This Chapter discusses the experimental results for those techniques. Tests were performed on the Gabor wavelet filtered images, and a set of useful correlation coefficients were determined. The Hough transform approach was investigated as both a pre-filtering method and the line angles and line length were investigated.

For the results reported in this section, images were separated randomly for the testing and training dataset. Each test is repeated for ten iterations, and the mean and standard deviation of the classification accuracy is reported. In Chapter 4.1 and 4.2, experiments will be conducted using the KNN classifier. In Chapter 4.3, analyses using the SVM classifier were performed. Parameters were modified for the different classifiers to determine the ideal settings for our application.

4.1 Gabor Results

For all test procedures, the training set varies between 50, 100, 150, 200, and 400 images. Ten iterations are calculated, and the average and standard deviation of the probability of detection is calculated. Figure 23 displays the different combinations of correlation coefficients used for training and testing our dataset. Notice that the frequency f varies in four values (f = 1/2, 1/4, 1/8, and 1/16), and the orientation θ ranges in four values (θ = 0, π/4, π/2, and 3/4π).

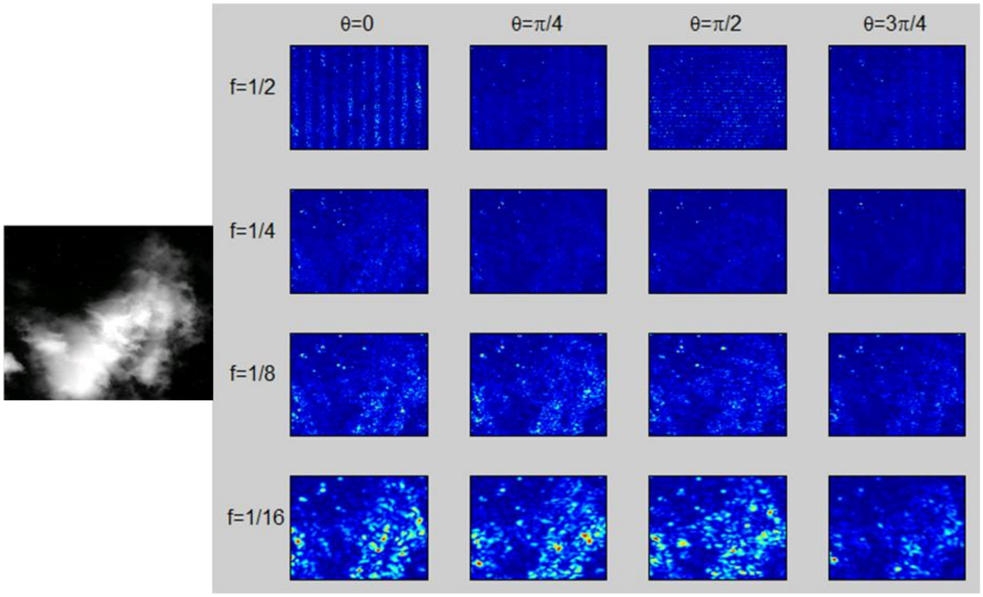


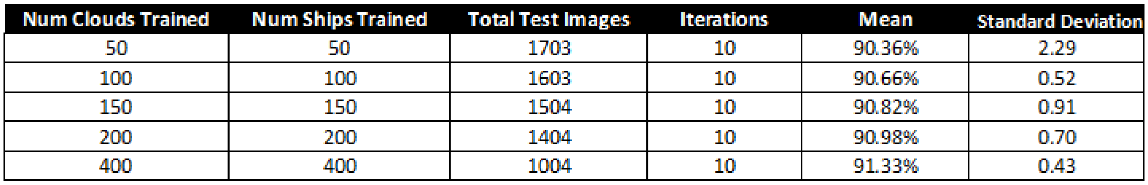
Figure 23. Gabor convolved cloud image ranging in four different frequency and four different orientations.

The first test performed is to calculate the correlation coefficient among the different frequency values of the Gabor filtered images. In other words, for every orientation value θ, we compute the normalized correlation coefficient for the following frequency pairs:

f = 1/2 vs. 1/4
f = 1/2 vs. 1/8
f = 1/2 vs. 1/16

The results are summarized below in Table 1, using the Matlab implemented KNN classifier with default Euclidean distance measure.

Table 1. Classification accuracy using 12 correlation coefficients and KNN Euclidean distance classifier.



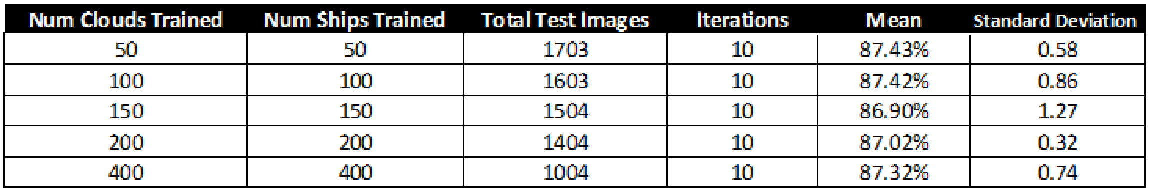
Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
50	50	1703	10	90.36%	2.29
100	100	1603	10	90.66%	0.52
150	150	1504	10	90.82%	0.91
200	200	1404	10	90.98%	0.70
400	400	1004	10	91.33%	0.43

For our second experiment, we calculated the correlation coefficient for the following scale combination, giving us only four correlation coefficients:

f = 1/4 vs. 1/8

The results are shown in Table 2. Notice that the detection rate has dropped to 87% when using only four features.

Table 2. Classification accuracy using four correlation coefficients and KNN Euclidean distance classifier.



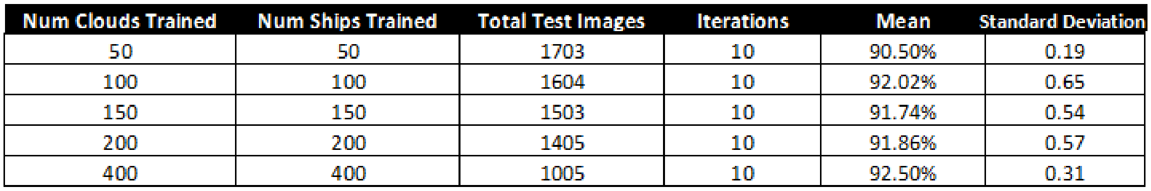
Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
50	50	1703	10	87.43%	0.58
100	100	1603	10	87.42%	0.86
150	150	1504	10	86.90%	1.27
200	200	1404	10	87.02%	0.32
400	400	1004	10	87.32%	0.74

For our third experiment, we combine twelve correlation coefficients with the above four correlation coefficients from experiments one and two. This will give us a total of 16 correlation coefficients. The four different frequency pair combinations are the following:

f = 1/2 vs. 1/4
f = 1/2 vs. 1/8
f = 1/2 vs. 1/16
f = 1/4 vs. 1/8

Results are summarized in Table 3. Notice that the detection rate has improved in comparison to our previous results, with the highest classification accuracy at 92.5%.

Table 3. Classification accuracy using 16 correlation coefficients and KNN Euclidean distance classifier.



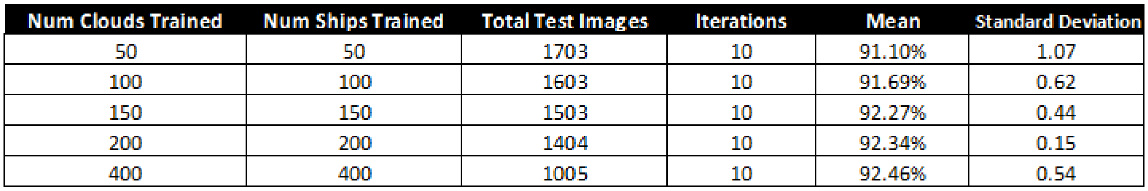
Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
50	50	1703	10	90.50%	0.19
100	100	1604	10	92.02%	0.65
150	150	1503	10	91.74%	0.54
200	200	1405	10	91.86%	0.57
400	400	1005	10	92.50%	0.31

For experiment four, we will test every combination of scale values. This will give us a total of 24 correlation coefficients. The six different scale combinations are the following:

f = 1/2 vs. 1/4
f = 1/2 vs. 1/8
f = 1/2 vs. 1/16
f = 1/4 vs. 1/8
f = 1/4 vs. 1/16
f = 1/8 vs. 1/16

Results are summarized in Table 4. The highest classification accuracy has not changed much compared to the previous experiment.

Table 4. Classification accuracy using 24 correlation coefficients and KNN Euclidean distance classifier.



Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
50	50	1703	10	91.10%	1.07
100	100	1603	10	91.69%	0.62
150	150	1503	10	92.27%	0.44
200	200	1404	10	92.34%	0.15
400	400	1005	10	92.46%	0.54

For experiment six, we will repeat the combinations set in the previous experiment, with the exception of

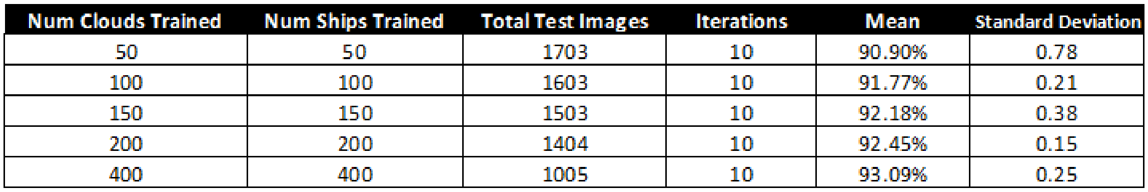
f = 1/2 vs. 1/8.

This will give us a total of 20 correlation coefficients. The five different scale combinations are the following:

f = 1/2 vs. 1/4
f = 1/2 vs. 1/16
f = 1/4 vs. 1/8
f = 1/4 vs. 1/16
f = 1/8 vs. 1/16

Results are summarized in Table 5. Notice that this has now improved our detection rate to 93%.

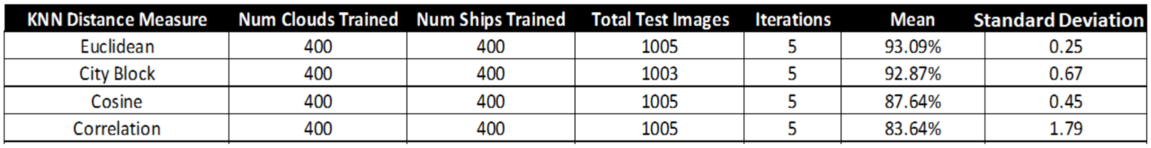
Table 5. Classification accuracy using 20 correlation coefficients and KNN Euclidean distance classifier.



Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
50	50	1703	10	90.90%	0.78
100	100	1603	10	91.77%	0.21
150	150	1503	10	92.18%	0.38
200	200	1404	10	92.45%	0.15
400	400	1005	10	93.09%	0.25

The method of distance measure used in the KNN classifier can have a significant impact on the classification results. For the last experiment, we will test four different distance measure techniques, and determine which method works best for the dataset. The results are shown in Table 5. The table below confirms for our dataset, the most accurate distance measure for KNN uses Euclidean distance.

Table 6. Modifications to the KNN distance measure using the twenty correlation coefficients shown in Table 5.



KNN Distance Measure	Num Clouds Trained	Num Ships Trained	Total Test Images	Iterations	Mean	Standard Deviation
Euclidean	400	400	1005	5	93.09%	0.25
City Block	400	400	1003	5	92.87%	0.67
Cosine	400	400	1005	5	87.64%	0.45
Correlation	400	400	1005	5	83.64%	1.79

In conclusion, our results show that the highest classification accuracy at 93.09% was achieved in experiment five using 20 correlation coefficients. It shows that having too few or too many features will reduce the classification accuracy. We can also conclude that the Euclidean distance measure used in KNN works best for our test set.

4.2 Pre-Filtering using Hough Transform

In Chapter 3.3, we discussed at length wavelet techniques used to improve edge detection. The edge image is then processed using the Hough transform, and the ship images are pre-filtered as ship objects. Using this technique, five out of 955 cloud images were mislabeled as ship objects using the Hough transform approach to identify parallel lines present in the test image. These mislabeled image chips are shown below in Figure 24. However, 345 out of 481 ship image chips were correctly labeled by the Hough transform approach.

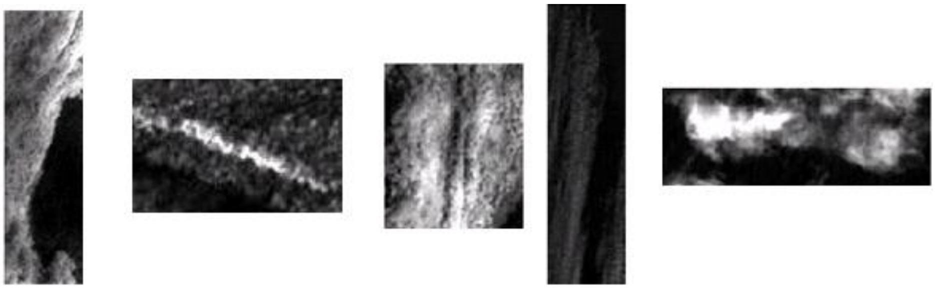


Figure 24. Cloud images mislabeled as ships through the Hough transform approach

Implementing the edge extraction method described in Chapter 3, and feeding the non-ship labeled images to a KNN classifier, we obtain the following classification accuracy summarized in Table 7. Notice that using the wavelet edge extraction approach has improved our results from 93% to 95%.

Table 7. Classification accuracy using 20 Gabor correlation coefficients, KNN classifier, and pre-filtering the dataset using line extraction through Hough transform. Edge images are computed using both Canny and Haar wavelet approach.

Trained Ships	Trained Clouds	Test Ships	Test Clouds	# Iterations	Mean	Std. Dev.
200	200	753	251	10	94.9	0.71
300	300	653	151	10	95.2	0.48

4.3 SVM Classifiers

The SVM classifier is a more complex classifier than KNN. It has several parameters the user can modify, such as assigning class weight values. We will explore various parameters of the classifier and determine what works best for our application. Additionally, we will examine the importance of data normalization for this classifier.

4.3.1 Hough Line Angles and Data Normalization

An initial test was performed to add the distribution of line angles as an additional feature. The new feature caused our classification accuracy to drop to 91%. We suspect this is due to a normalization issue and will explore a couple normalization techniques.

The first normalization technique we will apply to the testing/training set is the Standard Score approach. This is done by subtracting the mean μ from the dataset, and scaling by the standard deviation σ, as shown in the equation below.

$$X_i = \frac{X_i - \mu}{\sigma} \tag{4}$$

The SVM classifier works well when the majority of the training/testing dataset lies between -1 and 1. By implementing the Standard Score normalization to our dataset, the majority of the dataset will be in our target range. The results for this approach are shown below in Table 8. Notice that the classification accuracy has improved to 97.16%.

Table 8. Classification accuracy using SVM classifier and Standard Score normalization of training and testing dataset.

Iterations	Train Cloud	Train Ships	Test Clouds	Test Ships	Total Test	Clouds Classification Accuracy	Ships Classification Accuracy	Overall Accuracy (Mean)	Standard Deviation
10	200	200	753	251	1004	98.46%	93.27%	97.16%	0.65

The second normalization technique scales the dataset by the maximum value of the training set. The equation below is used.

$$X_i = \frac{X_i}{|\max(X)|} \tag{5}$$

This normalization approach will limit the dataset between -1 and 1. The classification results are a slight improvement from the previous method at 97.31%, as shown in Table 9.

Table 9. Classification accuracy using SVM classifier and maximum normalization of training and testing dataset.

Iterations	Train Cloud	Train Ships	Test Clouds	Test Ships	Total Test	Clouds Classification Accuracy	Ships Classification Accuracy	Overall Accuracy (Mean)	Standard Deviation
10	200	200	753	251	1004	98.55%	93.59%	97.31%	0.50

In section 4.1, we used the correlation coefficients between different scale factors of the Gabor convolved images as features for the training/testing set. These features were already bounded between -1 and 1. However, we recently introduced the distribution of Hough angles as a new feature, which requires that the dataset to be bounded. We have seen from our results in this section the importance of data normalization when using an SVM classifier.

4.3.1 Weight Factor

The results in Table 8 and Table 9 show that the cloud dataset has a significantly higher classification accuracy rate than the ship dataset. For our application, it is our preference to have a higher ship classification accuracy, even if this will result in having a lower overall classification accuracy. SVM has the option of modifying the box constraint, a weight factor that allows the user to penalize a certain class from being misclassified. We have modified the weight factor to be 1:100, with emphasis for ships to be correctly classified. The results are shown in Table 10. Notice that the ship classification results have improved from 93% to 96%.

Table 10. Classification accuracy using the SVM classifier with a 1:100 weighting factor to place high priority in ship classification accuracy.

Iterations	Train Cloud	Train Ships	Test Clouds	Test Ships	Total Test	Clouds Classification Accuracy	Ships Classification Accuracy	Overall Accuracy (Mean)	Standard Deviation
10	200	200	753	251	1004	97.84%	96.02%	97.38%	0.46

4.4 Hough Transform Line Detection Length

Our instinct would tell us that the line segments for the ship images are longer than those of the cloud segments. Therefore, using the maximum line length of the detected Hough lines as a feature could possibly increase our classification accuracy. A test is performed where we compute the Canny edge image of our test and train set, detect the length of the line segments using the Hough transform, and use the maximum line length as an additional feature for our SVM classifier. The results for this experiment are summarized in Table 11. This added feature has boosted our overall accuracy to 97.86%, but has slightly lowered our ship classification accuracy by 0.5%. Our explanation for the decline in ship accuracy is that it may be difficult to detect line segments of moving vessels, and therefore the line length may be similar to the clouds.

Table 11. SVM classifier with maximum Hough transform line length added as additional feature to SVM classifier.

Iterations	Train Cloud	Train Ships	Test Clouds	Test Ships	Total Test	Clouds Classification Accuracy	Ships Classification Accuracy	Overall Accuracy (Mean)	Standard Deviation
10	200	200	753	250	1003	98.62%	95.56%	97.86%	0.32

Chapter 5

Conclusion

5.1 Summary of Contributions

The thesis researched confirms that performing texture analysis alone is not enough to accurately classify cloud versus ship objects in high-resolution panchromatic satellite imagery. However, we have devised an algorithm which measures texture using Gabor wavelets, detects straight line features using the Hough transform, enhances our line detection using Haar wavelets, computes distribution of line angles and maximum line length, to accurately distinguish ships from cloud objects. We have shown for our application and dataset, the SVM produces the highest accuracy at 97.86%.

5.2 Future Work

The research conducted suggests a highly accurate algorithm for a two class problem, distinguishing ships from cloud objects. However, the objects detected by RAPIER® can be divided into four classes: clouds, ships, fast moving ships, and others (including land detects). One of the challenges in this research topic is to distinguish clouds from land objects. Both objects have similar textures and do not contain straight edges, making it a challenging computer vision problem to separate the two classes. Another challenge is to recognize an image chip that contains both a cloud and ship object. Often times the ship is obscured by clouds, such as the example shown below in Figure 25. Finally, the code developed for this research is written in Matlab. It will need to be converted to C++ in order to be integrated into RAPIER®.

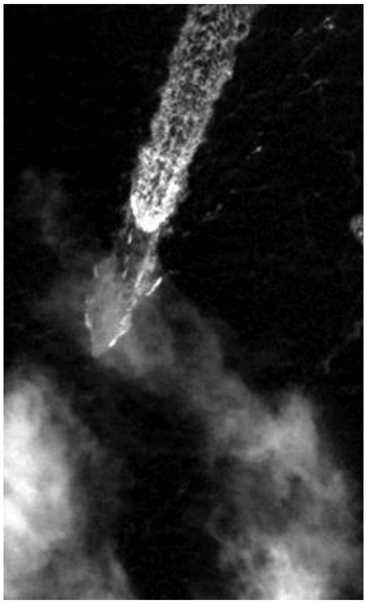


Figure 25. Example of ship objects obscured by clouds.

Bibliography

[1] Barrington, L., Har-Noy, S., Ricklin, N., Stastny, J., Rainey, K., and Buck, H. Ship "Identification in Cloudy Satellite Imagery: Computer Vision Methods for Cloud Rejection." *Applied Imagery Pattern Recognition: Imaging for Decision Making. AIPR 2011*, Oct 2011.

[2] Buck, H., Sharghi, E., Guilas, C., Stastny, J., Morgart, W., Schalcosky, B., Pifko, K.. "Enhanced Ship Detection from Overhead Imagery." *Society of Photo-Optical Instrumentation Engineering (SPIE), Society Proceedings, Optics and Photonics in Global Homeland Security IV*, vol. 6696, 2008.

[3] Fu, D., and Xu, L. "Satellite Cloud Image Texture Feature Extraction Based on Gabor Wavelet." *2011 4th International Congress on Image and Signal Processing*, vol. 1, 248 –251, Oct 2011.

[4] Gonzalez, R. C., and Woods, R. E., *Digital Image Processing*, Upper Saddle River: Pearson Prentice Hall, 2008.

[5] Khazenie, N., Richardson, K. "Comparison of Texture Analysis Techniques in Both Frequency and Spatial Domains for Cloud Feature Extraction*." International Society for Photogrammetry and Remote Sensing* (*ISPRS)*, Aug 1992.

[6] Kyrki, V., Kamarainen, J.-K., "Simple Gabor Feature Space for Invariant Object Recognition." *Pattern Recognition Lett*ers, vol. 25, 311-318, 2004.

[7] Rees, W. *Physical Principles of Remote Sensing, Second Edition*, Cambridge: Cambridge University Press, 2001.