

Ship Detection in High-Resolution Optical Imagery Based on Anomaly Detector and Local Shape Feature

Zhenwei Shi, *Member, IEEE*, Xinran Yu, Zhiguo Jiang, and Bo Li

Abstract

Ship detection in high resolution optical imagery is a challenging task due to the variable appearances of ships and background. This paper aims at further investigating this problem and presents an approach to detect ships in a “coarse to fine” manner. First, to increase the separability between ships and background, we concentrate on the pixels in the vicinities of ships. We rearrange the spatially adjacent pixels into a vector, transforming the panchromatic image into a “fake” hyperspectral form. Through this procedure, each produced vector is endowed with some contextual information, which amplifies the separability between ships and background. Afterwards, for the “fake” hyperspectral image, a hyperspectral algorithm is applied to extract ship candidates preliminarily and quickly by regarding ships as anomalies. Finally, to validate real ships out of ship candidates, an extra feature is provided with Histograms of Oriented Gradients (HOG) to generate hypothesis using AdaBoost algorithm. This extra feature focuses on the gray values rather than the gradients of an image and includes some information generated by very near but not closely adjacent pixels, which can reinforce HOG to some degree. Experimental results on real database indicate that the hyperspectral algorithm is robust, even for the ships with low contrast. Besides, in terms of the shape of ships, the extended HOG feature turns out to be better than HOG itself as well as some other features such as Local Binary Pattern.

Index Terms

Ship detection, Optical panchromatic image analysis, “Reed-Xiaoli” algorithm, Circle Frequency-Histograms of Oriented Gradients feature.

This work was supported by the National Natural Science Foundation of China under the Grants 61273245 and 91120301, the 973 Program under the Grant 2010CB327904, the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-12KF-07), and Program for New Century Excellent Talents in University of Ministry of Education of China under the Grant NCET-11-0775. The work was also supported by Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, P.R. China.

Zhenwei Shi, Xinran Yu and Zhiguo Jiang are with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: shizhenwei@buaa.edu.cn (Zhenwei Shi); yuxinran.buaa@gmail.com (Xinran Yu); jiangzg@buaa.edu.cn (Zhiguo Jiang)).

Bo Li is with the Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: boli@buaa.edu.cn).

I. INTRODUCTION

Ship detection, one of the hottest issues in the field of remote sensing, is of great significance owing to its wide range of applications such as oceanic traffic surveillance, ship rescue, fishery management and so forth. In the existing researches, synthetic aperture radar (SAR) images are one of the data sources which have been most widely studied because they can be obtained day and night, regardless of the impact from meteorological conditions [13]. Furthermore, with the successful launching of the high-resolution SAR sensors, more high-resolution SAR data become available now, which enlarges the potential of SAR data for ship detection [40].

Another important data source for ship detection is optical imagery, which is investigated in this paper. The foremost advantage of optical imagery is its higher resolution. The relatively high resolutions of optical images can generate more detailed information of ships. However, high resolution image also complicates the background, which will increase the processing time and even cause a lot of false alarms. To handle the complexity of optical imagery, the first need could be a robust feature set discriminating the ships from non-ship objects cleanly, even in some cluttered background such as waves, clouds, small islands and so forth. Besides, with the purpose of accelerating the whole process and decreasing false alarms, some efficient predetection algorithms are also required to extract ship candidates preliminarily.

A. Related Work

To solve the problems of ship detection in high resolution optical images, several approaches have been investigated. The existing methods can roughly be divided into three categories. The first ones are threshold segment methods which focus on the differences in gray values between ships and background. The second ones are statistical methods concentrating on the gray value distributions of ship scenes. The last ones are classification based methods in which great attention has been paid to the different kinds of features as well as various classifiers. Most existing methods took the “coarse to fine” strategy and combined different algorithms together to detect ships more efficiently. For instance, Burgess [4] proposed a method including masking, filtering and shape analysis techniques to detect ships in optical imagery. Proia and Page [35] raised a Bayesian algorithm to accomplish predetection for small ships. In their method, a statistical test was built first, then ships were detected by choosing the most likely result using Bayesian decision theory. Xia, Wan and Yue [41] came up with an uncertain ship target extraction algorithm based on dynamic fusion model. They first chose and fused several geometrical features by the dynamic fusion model and then extracted ships by the support vector machine. Corbane, Marre and Petit [5] developed a method for operational detection of small ships. The ships were first segmented by the contrast between ships and background, afterwards, some feature sets were extracted and the detection tasks were accomplished by genetic algorithm and neural networks. Corbane *et al.* [6], [7] proposed a processing chain involving statistical methods, mathematical morphology and some signal-processing techniques to detect ships. In their approach, the input image was first preprocessed by methods such as cloud masking and contrast enhancement. Then the targets were detected preliminarily by connected filtering and automatic threshold. Finally, Wavelet and Radon transforms were applied in combination with logistic model to achieve the postprocessing. Zhu *et al.* [42] put forward a hierarchical method

including shape analysis, feature extraction and a semi-supervised classification algorithm. They first extracted ship candidates by a threshold and employed a novel hierarchical classification stage to detect ships. Bi *et al.* [3] raised a visual search inspired computational model. In their approach, the salient candidates regions were first selected by a bottom-up mechanism and then real ships were validated by local feature descriptors and the support vector machine.

In summary, significant efforts have been taken to detect ships from high resolution optical imagery and various approaches have been investigated in this field. However, some open issues still exist. The false alarm rate tends to be high due to clouds, sea waves and small islands. In addition, missing detection also exists, especially when a ship is occluded or with low contrast. Therefore, there still remains room for improvement in this field.

B. Contributions

To further investigate the problems arising from optical panchromatic images, a hierarchical framework including a predetection stage and an accurate detection stage is utilized in this paper. In the predetection stage, we first convert the panchromatic image into a “fake” hyperspectral form to project ship candidates. Then a hyperspectral algorithm (a hyperspectral anomaly detector) is applied to extract ship candidates preliminarily and quickly based on anomaly detection model. Afterwards, in the accurate detection stage, a new extension of Histograms of Oriented Gradients (HOG) is extracted and hypotheses for ships are generated using AdaBoost algorithm to detect ships more accurately. Fig. 1 shows the whole process of this approach.

In this paper, we have three major contributions. First, we transform the panchromatic image into a “fake” hyperspectral form. The “fake” hyperspectral image can represent data more properly because the separability between ships and background is augmented and the contextual information of ships is converted into a hyperspectral form, which can be naturally processed by the subsequent hyperspectral algorithm. Second, in the anomaly detection fields, we adopt a hyperspectral anomaly detector to the panchromatic image to extract the ship candidates. This hyperspectral anomaly detector can project the abnormal and fluctuating spectral vectors while suppressing the normal and flat ones. Combined with the “fake” hyperspectral image, this hyperspectral algorithm can extract ship candidates efficiently, especially for the sea scenes which contain large areas of smooth background. Third, in terms of the shape of ships, we provide the existing Histograms of Oriented Gradients (HOG) feature with some extra information based on the gray values along a circle over the ship, which HOG is deficient in, leading to more discriminative power for ships.

The rest of this paper is organized as follows. Section II describes the extraction of ship candidates, including the process of “fake” hyperspectral data generation and the hyperspectral predetection algorithm. Section III depicts the process of accurate detection including feature extraction and classification. Section IV presents the experimental results from real database. Finally, Section V provides the discussion and conclusion of this paper.

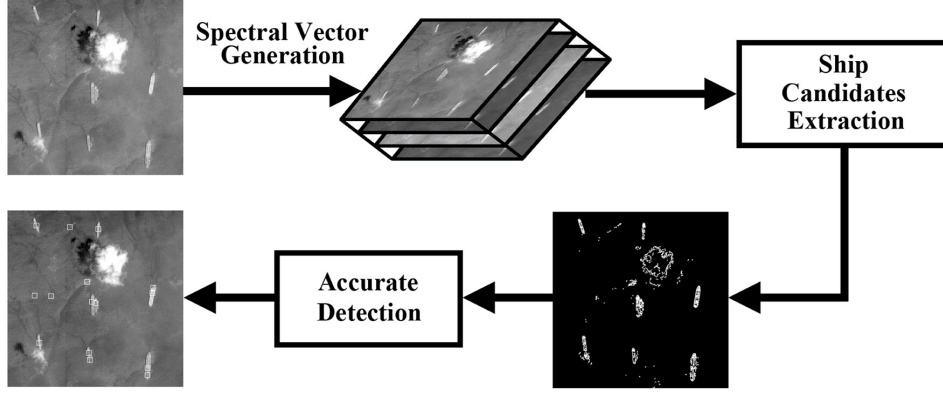


Fig. 1. The outline of our method, which can mainly be separated into two parts: Ship Candidates Extraction and Accurate Detection.

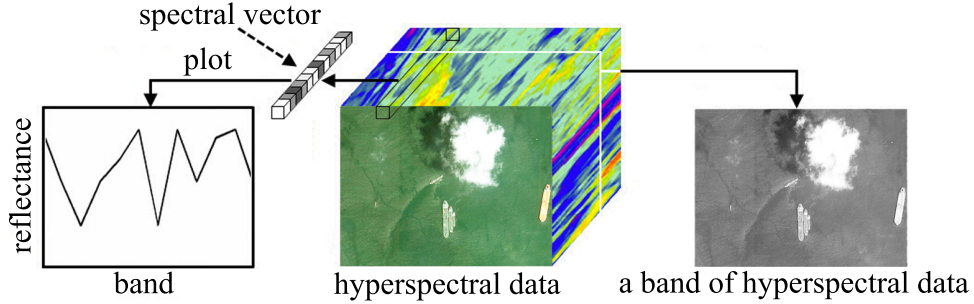


Fig. 2. A hyperspectral image taken by a hyperspectral sensor. The left part is a spectral vector of a pixel and the right part is a certain band of the hyperspectral image.

II. PREDETECTION

In this stage, with the purpose of detecting ship candidates preliminarily and quickly, a method to extract all ship candidates from the whole image without any omission is needed. Before elaborating our method, it is necessary to introduce the hyperspectral data first since our method involves a hyperspectral algorithm.

Hyperspectral images are images with a set of bands or channels [28]. Specifically, a hyperspectral image is a three-dimensional data cube including a set of bands, as shown in Fig. 2. Besides, each pixel in a hyperspectral image does not correspond to a scalar value, but a set of values, namely, a vector. The left part of Fig. 2 plots such a vector in an x-y plane and this vector will be denoted as “spectral vector” in the rest of this paper. Spectral vectors can be used to identify different materials in a scene, for example, the pattern of the spectral vectors of oil is usually different from that of sea water. Furthermore, a hyperspectral image can also be visualized as a stack of images and Fig. 2 shows a certain band of the hyperspectral image in the right part.

After the introduction of hyperspectral imagery, the main idea of ship candidates extraction will be presented here. In a hyperspectral image, different materials can be identified by their spectral vectors. While in an optical panchromatic image, different objects can be distinguished by the gray values, or more exactly, by the gray values of

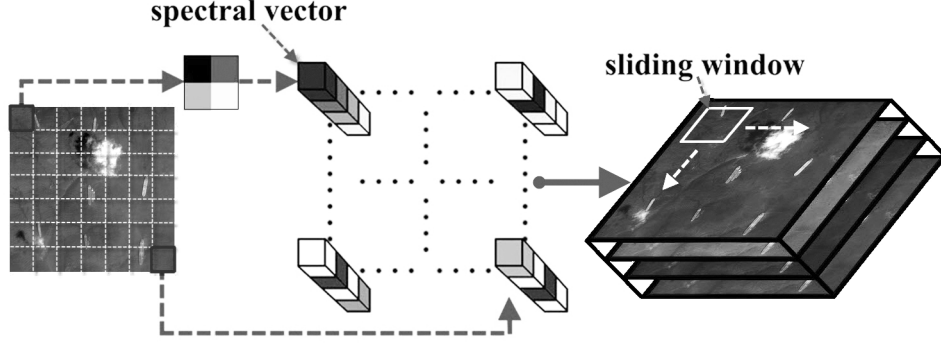


Fig. 3. The process of spectral vector generation. The left part shows the input image and the right part shows the produced “fake” hyperspectral image. The white parallelogram denotes the sliding window to guarantee the gaussianity of background.

spatially adjacent pixels. There could be some underlying relationships between the two types of data. Motivated by this idea, we transform the spatially adjacent pixels in an optical panchromatic image into spectral vectors, making a new hyperspectral form of data. Since the pixels of ships and some clutters are variable, their spectral vectors in the “fake” hyperspectral image tend to fluctuate drastically. On the contrary, the pixels of smooth background are more liable to be homogeneous, accordingly, their counterparts in the “fake” hyperspectral image trend to be flat, or at least not as fluctuating as the spectral vectors of ships. Then the two different patterns of spectral vectors can be identified by a hyperspectral anomaly detector and those spectral vectors which are identified as fluctuating are the detected ship candidates to be further analyzed.

A. Spectral Vector Generation

Given the analysis above, the panchromatic image will be first transformed into a hyperspectral form. Specifically, given an $H \times W$ panchromatic image \mathbf{I} , hypothetical non-overlapping $k \times k$ patches are masked on it and the total number of patches is $L = (H/k) \times (W/k)$. For each patch, the pixels within it are sampled and rearranged into a k^2 -dimensional vector \mathbf{x} to serve as the produced spectral vector. After all patches are transformed, the produced vectors are combined together, making a three-dimensional $(H/k) \times (W/k) \times k^2$ data cube, namely, a “fake” hyperspectral form of data. This process is presented in Fig. 3. The left part of Fig. 3 is the panchromatic image with small patches masked on it. The middle part of Fig. 3 shows how the pixels in the panchromatic image are transformed into spectral vectors. The right part of Fig. 3 is the produced three-dimensional data cube.

However, with such non-overlapping patches, the ships which happen to lie on the borderlines between two patches will be cut into pieces. In fact, this problem can be avoided by replacing the non-overlapping patches with a window to generate spectral vectors pixel by pixel. That is to say, for each pixel in the panchromatic image, its neighbors confined by a $k \times k$ window are copied and shifted to form a spectral vector. After the window sliding throughout the whole image, we will finally generate a data cube of the size $H \times W \times k^2$, whose height and width are the same as the original panchromatic image’s. Usually, this three-dimensional data cube is stored as a

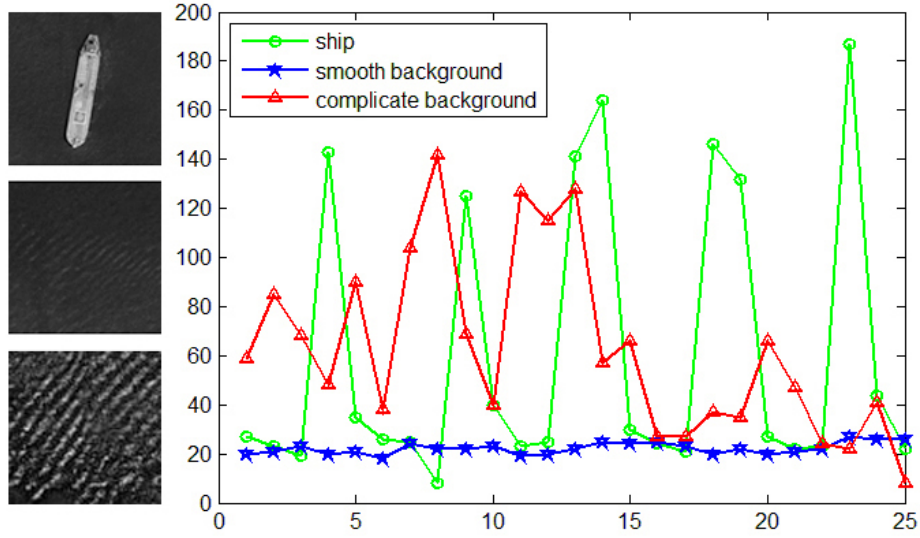


Fig. 4. The graph plots the adjacent pixels in the central areas of the three images on the left. These pixels are plot from left to right, from top to bottom.

$k^2 \times (H \cdot W)$ matrix, in which each column represents a spectral vector.

Simple as it is, the transformation from a panchromatic image to a “fake” hyperspectral one makes sense because the univariate gray level distribution of a panchromatic image is expanded to a multivariate one. More importantly, through this transformation, some shape information and contextual information are imported into each spectral vector in the produced hyperspectral form of data, which makes the ships more distinct from background.

What should be emphasized here is the size of the $k \times k$ window which determines the dimension of the produced spectral vector. Note that the size of the $k \times k$ window will be denoted as “spectral dimension” in the rest. If the spectral dimension is too large, some redundant and unexpected information could be included in the spectral vectors. On the contrary, if the spectral dimension is too small, the information in a spectral vector will be too little to distinguish between ships and background. Generally speaking, the spectral dimension should be selected according to the size of ships in an image and the experiments on it will be presented in section IV.

Another fact should be emphasized and clarified is the potential lying in this “fake” hyperspectral image. Above we have mentioned that the transformation increases the separability between ships and background, however, it does not mean that the produced “fake” hyperspectral image can match with a real hyperspectral image taken by the hyperspectral sensor. The reason is that the information in the “fake” hyperspectral image all comes from the original panchromatic image, accordingly, it is impossible to identify different materials in the “fake” hyperspectral image. But when it comes to distinguishing ships from background by hyperspectral anomaly detector, the “fake” hyperspectral image will be discriminative enough, because what the hyperspectral anomaly detector needs is a separability or difference between ships and background. Considering the fact that the pixels of ships are variable

while those of smooth background have great similarity, the produced spectral vectors of ships will be very different from those of smooth background. Fig. 4 shows some spectral vectors of ships and background. The right graph in Fig. 4 plots the adjacent pixels in the central areas of the left images. As Fig. 4 shows, the spectral vector of ship is fluctuating and turbulent while the spectral vector of smooth background is flat. The two different patterns can be distinguished by the hyperspectral anomaly detector. However, this transformation is not very sophisticated and only smooth background can be separated from ships. Some complicated background such as very strong waves will also produce fluctuating spectral vectors, which will be mixed with ships. Therefore, the hyperspectral anomaly detector in combination with the “fake” hyperspectral image can only be used as a predetection method to extract ship candidates preliminarily. As to the false alarms in ship candidates, they will be processed in the accurate detection stage.

B. Ship Candidates Extraction

After generating the “fake” hyperspectral image, a hyperspectral anomaly detector is used to extract ship candidates preliminarily on the basis of anomaly detection model. Hyperspectral anomaly detectors aim at unsupervisedly distinguishing unusual spectral vectors (also called “anomalies”) from typical and homogenous background. The key of these algorithms lies in extending the separability between anomalies and background by fully exploiting the spectral differences acquired from the hyperspectral image. For example, the random-selection-based anomaly detector [11] performs the random selections repeatedly so as to achieve more accurate background statistics and to increase the aforementioned separability. The kernel-based target-constrained-interference-minimized filter [39] utilizes nonlinear kernel functions to map the hyperspectral data into a high-dimensional space in which the anomalies are assumed to be more separable from background. Besides, transfer learning is employed in [12] to enhance the separability between anomalies and background by unsupervisedly constructing a subspace based detector.

The hyperspectral anomaly detector used in our approach is named as “Reed-Xiaoli” (RX) [36], an unsupervised benchmark algorithm for hyperspectral anomaly detection [18], [29], [37], [38]. RX algorithm regards the pixels in a hyperspectral image as two categories: background and target. The spectral vectors of background are common and usually take a large proportion in an image. On the contrary, the spectral vectors of targets are abnormal and only take a tiny proportion. In order to distinguish target from background, RX first gives an estimate of the background and then calculates a kind of distance away from the estimated background for each pixel. Clearly, the spectral vectors with longer distances will be identified as targets. In our problem, RX is suitable in that the ships and the complicated clutters can be viewed as targets, while the smooth areas of sea can be viewed as background.

Now the principle of RX will be introduced in details. In a produced “fake” hyperspectral image, the spectral vector of each pixel can be denoted as a P -dimensional vector of the form \mathbf{x} . Then a hyperspectral image with N pixels in total can be denoted as a $P \times N$ matrix $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(N)]$, in which each column denotes a spectral vector.

Given a spectral vector $\mathbf{x}(n)$ ($n = 1, 2, \dots, N$), RX will distinguish between the two hypotheses below [25]:

$$\begin{aligned}
H_0 : \mathbf{x}(n) &= \mathbf{n} \\
H_1 : \mathbf{x}(n) &= \alpha \mathbf{s},
\end{aligned} \tag{1}$$

where \mathbf{n} is a vector denoting background and noise, α is a positive constant, and \mathbf{s} denotes the target patterns. H_0 models the background as a Gaussian distribution $N(\boldsymbol{\mu}_b, \mathbf{C}_b)$ while H_1 models the target patterns as a Gaussian distribution $N(\mathbf{s}, \mathbf{C}_b)$. It can be seen that the two distributions have the same covariance matrix but different mean values. Notice that $\boldsymbol{\mu}_b$, \mathbf{s} and \mathbf{C}_b are all assumed to be unknown but they can be estimated locally or globally from the hyperspectral image.

In order to distinguish between the two hypotheses, RX algorithm supposes that all spectral vectors are observations from the two Gaussian distributions, with the same covariance matrix \mathbf{C}_b but different mean values [30]. Then, for each spectral vector $\mathbf{x}(n)$, $n = 1, 2, \dots, N$, a measure $\delta(\mathbf{x}(n))$ is computed as [25]:

$$\delta(\mathbf{x}(n)) = (\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b)^T (\hat{\mathbf{C}}_b)^{-1} (\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b), \tag{2}$$

where $\hat{\boldsymbol{\mu}}_b$ and $\hat{\mathbf{C}}_b$ are the maximal likelihood estimates of $\boldsymbol{\mu}_b$ and \mathbf{C}_b respectively:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_b &= \frac{1}{N} \sum_{n=1}^N \mathbf{x}(n) \\
\hat{\mathbf{C}}_b &= \frac{1}{N} \sum_{n=1}^N (\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b)(\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b)^T.
\end{aligned} \tag{3}$$

Before analyzing formula (2), one thing about formula (3) should be clarified first. In (3), we want to estimate the mean value and covariance matrix of background: $\boldsymbol{\mu}_b$ and \mathbf{C}_b , but all pixels $\mathbf{x}(n)$, $n = 1, 2, \dots, N$ including targets and background in the image are used. That is to say, to estimate background properly, formula (3) makes sense only when the background takes a dominant proportion of the image while the targets take a tiny one so that the distributional differences between background and the whole image will be negligible.

Now a further insight will be given into (2). It can be seen from (2) that $\delta(\mathbf{x}(n))$ is, in essence, the square of the Mahalanobis distance [27] between $\mathbf{x}(n)$ and the estimated distribution of background $N(\hat{\boldsymbol{\mu}}_b, \hat{\mathbf{C}}_b)$. In other words, a larger $\delta(\mathbf{x}(n))$ means a longer Mahalanobis distance to $N(\hat{\boldsymbol{\mu}}_b, \hat{\mathbf{C}}_b)$. According to the characteristic of Mahalanobis distance [26], the target patterns are more likely to give larger values of $\delta(\mathbf{x}(n))$ since they are usually abnormal and have low probabilities of occurrence. While the common areas of background are liable to give smaller values of $\delta(\mathbf{x}(n))$. Therefore, in a scene of sea and ships, the ships as well as some complex clutters will generate larger values of $\delta(\mathbf{x}(n))$ while the smooth areas of sea will generate smaller values of $\delta(\mathbf{x}(n))$. From this perspective, the ships and the complex clutters will be projected, while the smooth areas of sea will be suppressed. Accordingly, ship candidates can be extracted by a threshold. In our approach, Otsu's method [31] is applied to select the threshold automatically. Otsu's method is suitable in that the outputs of RX can be roughly divided into two classes (target and background), which accords with the assumption of Otsu's method [31].

Otsu's method aims at dichotomizing the pixels in an image into two classes by finding a threshold which maximizes the separability between the two dichotomized classes. Now suppose that the pixels in an image is divided into two classes C_0 and C_1 by a threshold k , where C_0 includes the gray levels $[1, \dots, k]$ and C_1 includes the gray levels $[k+1, \dots, L]$ (if there are L gray levels in total). Then the variances of the two classes can be given by $\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 p_i / \omega_0$ and $\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 p_i / \omega_1$, where p_i is the probability of gray level i ; $\omega_0 = \sum_{i=1}^k p_i$ and $\omega_1 = \sum_{i=k+1}^L p_i$ are the probabilities of class occurrence; $\mu_0 = \sum_{i=1}^k i p_i / \omega_0$ and $\mu_1 = \sum_{i=k+1}^L i p_i / \omega_1$ are the class mean values. Also, we can give the the mean value of the original image $\mu_T = \sum_{i=1}^L i p_i$. For any choice of k , we can easily verify the following equations:

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T, \quad \omega_0 + \omega_1 = 1. \quad (4)$$

Given (4), the separability σ_B^2 between the two classes are given by:

$$\begin{aligned} \sigma_B^2 &= \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \\ &= \omega_0 \omega_1 (\mu_0 - \mu_1)^2. \end{aligned} \quad (5)$$

Notice that σ_B^2 is a function with respect to the threshold k so σ_B^2 can also be denoted as $\sigma_B^2(k)$. The target threshold k^* is the one which maximizes $\sigma_B^2(k)$:

$$k^* = \arg \max_{1 \leq k \leq L} \sigma_B^2(k). \quad (6)$$

Since the value of k is selected from 1 to L , we can calculate all $\sigma_B^2(k)$, ($k = 1, \dots, L$) and then find out k^* which maximizes $\sigma_B^2(k)$.

In order to apply RX to panchromatic imagery more properly, another two techniques are employed: the sliding window and the regularization term.

1) *RX Algorithm with Sliding Window*: the first one is the sliding window, mainly to guarantee the gaussianity of background. Specifically, RX is deduced on the condition that the background in an image follows Gaussian distribution. However, all pixels in a large image can hardly satisfy this condition due to the complexity and variation of the image. Therefore, if the image is very large, say 5000×5000 , applying RX directly will be inappropriate and a kind of sliding window will be of great necessity [37]. Obviously, the size of this sliding window should not be too large on account of the gaussianity condition. And it should not be too small either. The reason can be revealed by formula (3). In light of the aforementioned analysis about (3), we want to estimate the distribution of background, but all pixels are used. Therefore, to make the estimation accurate, the background in the image should take a very large proportion. From this perspective, the sliding window should be large enough to guarantee the dominant amount of background. In Fig. 3, the white parallelogram on the produced "fake" hyperspectral image is this sliding window. The experiments on the size of the sliding window will be shown in section IV.

2) *RX Algorithm with Regularization Term*: another technique employed in our approach is the regularization term which is applied for fear of singular $\hat{\mathbf{C}}_b$. As is shown in (2), RX needs to compute the inverse of $\hat{\mathbf{C}}_b$, but $\hat{\mathbf{C}}_b$

is not ensured to be nonsingular all the time. When encountering singular $\hat{\mathbf{C}}_b$, RX will have poor performance. In fact, this problem can be improved by adding a scaled identity matrix to $\hat{\mathbf{C}}_b$ before inverting, because the inverse is strongly affected by the small eigenvalues (considering the inverse matrix in terms of its eigenvalue decomposition). Thus, adding a scaled identity can decrease the influence of the small eigenvalues and consequently making a more stable inverse of $\hat{\mathbf{C}}_b$. Furthermore, it has been proven that adding a scaled identity matrix to $\hat{\mathbf{C}}_b$ before inverting is equivalent to including a regularization term to the cost function when designing RX [30]. More detailed information about the proof can be seen in [30]. Thus, the regularized RX can be given by:

$$\delta_{\text{RX}}(\mathbf{x}(n)) = (\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b)^T (\hat{\mathbf{C}}_b + \beta \cdot \mathbf{I})^{-1} (\mathbf{x}(n) - \hat{\boldsymbol{\mu}}_b), \quad (7)$$

where β is a small positive constant, and \mathbf{I} is an identity matrix.

To be honest, for the “fake” hyperspectral image, the number of pixels will greatly surpass the dimension of the spectral vector and the covariance matrix is liable to be nonsingular. However, it still has the possibility that the covariance matrix is singular and in such cases RX with the regularization term will produce better results than without it. Notice that β is selected empirically. Intuitively, the value of β is related to the spectral dimension and the size of the sliding window since β is used to amend the ill-rank covariance matrix. In our experiment, we find that when the size of the window is smaller than or equal to the spectral dimension, β will affect the results of RX strongly. Generally speaking, a relatively small β (for example, 10^{-5}) will produce a lot of false alarms. When β increases, the false alarms will decrease greatly. However, when the size of the window is much bigger than the spectral dimension (for example, the size of the window is four times of the spectral dimension), the value of β will not influence the results of RX so greatly. Specifically, when β is relatively small (for example, 10^{-5}), the results with the regularization term will be very similar to the results without this term. When β gets bigger (for example, 10^{-3}), the performance of RX will improve to some degree. However, when β continues to increase, (for example, 10^{-2} or 10^{-1}), the results of RX do not have great differences.

Besides, we also find that when the size of the window is much bigger than the spectral dimension, the elements of the covariance matrix will have similar values (with the same order of magnitude). However, when the size of the window is smaller than or equal to the spectral dimension, the covariance matrix will be approximate to a diagonal matrix. This phenomenon also explains why the relatively small size of the window will produce poor performance of RX, since the Mahalanobis distance will reduce to the Euclidean distance when the covariance matrix is an identity matrix.

III. ACCURATE DETECTION

After the process of “Reed-Xiaoli” (RX), several spectral vectors including the ships and complex clutters are obtained. Then all these detected spectral vectors are remapped into the panchromatic image to serve as the ship candidates. Our current task is to give a further insight into the extracted ship candidates and to find out real ships accurately. In our approach, local feature descriptors in combination with Adaboost classifier are used to achieve this goal. What we need here is a robust feature set which is able to distinguish ships cleanly while maintaining

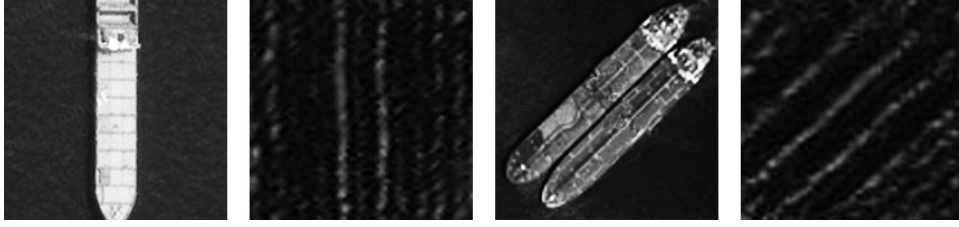


Fig. 5. The samples of ships and waves which are similar to ships.

its invariance to certain image variations. As we know, Histograms of Oriented Gradients (HOG) [8] is a very robust feature which can reliably grasp the shape information of local objects. HOG is based on the idea that the appearance and shape of local object can be characterized rather well by the distribution of edge directions or gradients [8]. Practically, for an sample image, its gradient image is first generated by computing the difference. Then the gradient image is divided into small regions (also called “cells”) and, for each cell, a 1-D histogram of gradient directions over the pixels in the cell is calculated. Furthermore, each 1-D histogram is normalized by the energy over somewhat larger regions (also called “blocks”) for better invariance to illumination. Finally, the combination of the normalized histogram entries forms the final feature vector, namely, the HOG descriptor.

Obviously, HOG is extracted only based on the gradients of an image, so we put some extra information into HOG for further improvement. This improvement is essential due to the kind of clutters which are very similar to ships, as shown in Fig. 5. The first and third images in Fig. 5 are two samples of ships and the second and fourth images are the clutters similar to ships. Generally speaking, the shape of a ship is a little simple because its most prominent characteristic is the two parallel borderlines which are very similar to some waves in the image. Therefore only by gradient information which is closely related to the edges in an image, ships can hardly be distinguished cleanly from such clutters.

Therefore, in our approach, an extra feature is provided with HOG in terms of the shape of ships. This extra feature does not focus on the gradients of an image, but rather on the pattern of gray values of ships. Furthermore, this feature is extracted based on the pixels along a circle over the ship, rather than the closely adjacent pixels on which HOG is extracted. So this feature can produce some information for ships which HOG is deficient in. Then the extra feature will be introduced in details.

A. CF Feature

The extra feature is named as “Circle-Frequency” (CF) feature which is generated on the basis of Circle-Frequency (CF) filter [22] and CF filter has been applied to several applications [20], [21], [23], [33]. The principle of CF filter is as follows.

In an image, ships will be either brighter or darker than the background in their vicinities. We first consider the situation of brighter ships. Now suppose that there is a hypothetical circle centered in the center of the ship, and its diameter is longer than the width while shorter than the length of the ship, as is shown on the left side of Fig. 6.

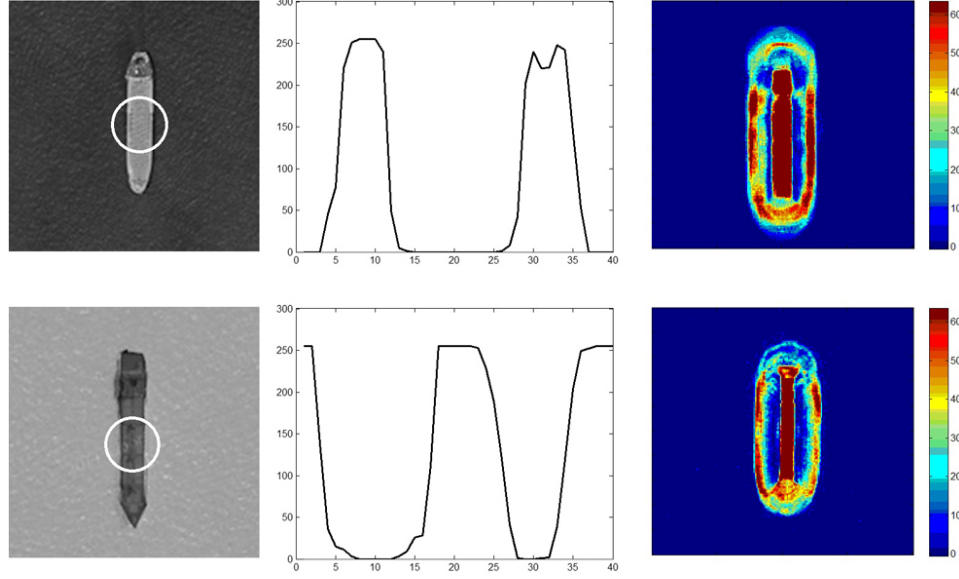


Fig. 6. The pixel values along the circle and the output of CF-filter. The left column shows the ships. The upper one is a bright ship while the lower one is a dark ship. The middle column shows the values along the circle over each ship. The horizontal axis denotes the serial number of the 40 sampled points and the vertical axis denotes their gray values. The right column shows the ships' outputs of Circle-Frequency filter which are shown using pseudo-colour.

Then the gray values along this circle trend to be dark-bright-dark-bright if we start from the left point. In a word, the gray values will have two peaks and two valleys, which resembles a 2-cycle sinusoidal signal. The upper row of Fig. 6 presents a brighter ship and the graph in the middle plots the gray values along the circle. This pattern is special for ship and can be used to generate an efficient feature.

To utilize this pattern, the Discrete Fourier Transform (DFT) of the gray values are computed. Specifically, let $f_k (k = 0, 1, \dots, N - 1)$ denote the N pixel values along a circle of radius r , centered at (i, j) , then, the DFT of the series f_k is computed as the formula below, and we denote the result of pixel (i, j) as $DFT_{(i,j)}$:

$$DFT_{(i,j)} = \frac{1}{N} \sqrt{\left(\sum_{k=0}^{N-1} f_k \cos \frac{ck\pi}{N} \right)^2 + \left(\sum_{k=0}^{N-1} f_k \sin \frac{ck\pi}{N} \right)^2}, \quad (8)$$

where N is the number of the points along the circle, c is a coefficient determining the frequency of the sine and cosine function in the process of DFT. Each pixel (except the pixels near the boundary) in the image will produce an output $DFT_{(i,j)}$. Finally, an output image is produced. The right part of Fig. 6 shows the output image of CF filter. Because the circles over the ships are all 2-cycle signals, then the parameter c should be set to 4 so that the body of the ships will give stronger responses while other parts the smaller ones. If c changes, the output image will be different. The experiment on c will be presented in section IV.

Above, we have only concerned the brighter ships. As to the darker ships whose gray values are lower than the background in their vicinities, the situation is a little different, because the gray values along the circle will be bright-dark-bright-dark starting from the left point. The second row in Fig. 6 shows the situation of darker ships.

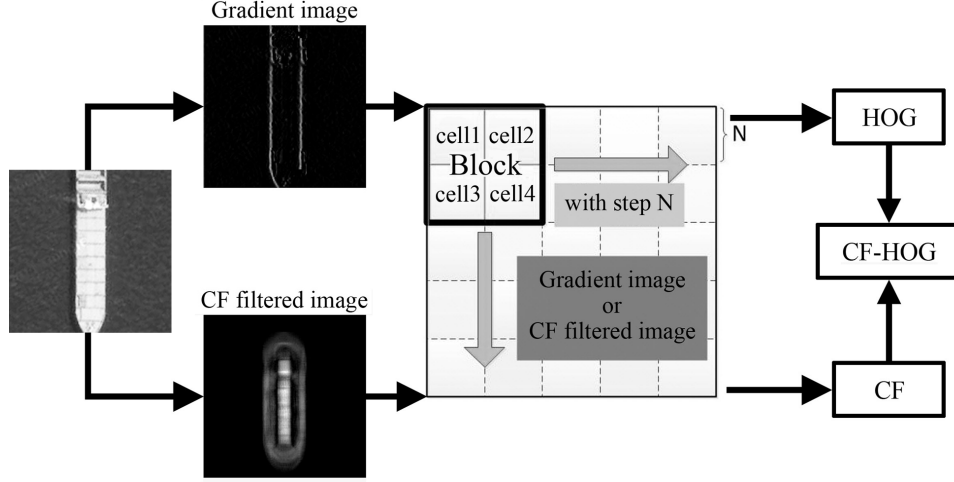


Fig. 7. The process of CF-HOG extraction, including the gradient image, the circle-frequency filtered image as well as the cells and blocks.

Obviously, the gray values along the circle is also a 2-cycle signal but its phase changes. According to DFT, the change in phase will not affect its output. Thus the bright ships and dark ships will give similar outputs.

Then, in order to form a feature vector, the output image is quantified and its histograms are computed. Just like what is used in Histograms of Oriented Gradients (HOG), some cells and blocks are also applied to grasp the shape information extensively. Finally, all histograms are combined together to form the CF feature. The detailed process of CF feature extraction will be presented in the extraction of CF-HOG.

B. CF-HOG Feature

The extracted Circle-Frequency (CF) feature is then combined with Histograms of Oriented Gradients (HOG) [8], making a new feature named as “CF-HOG”. In this paper, the training samples are all 40×40 images. The cells are 8×8 and the blocks are 16×16 . For the CF filtered image, the gray values are quantified into 9 gray bins (the experiment on the gray bin will be presented in section IV). Fig. 7 shows the extraction of CF-HOG for a 40×40 input image **I** and the details are as follows:

- 1) Generate the Gradient image [8] and the CF filtered image of the input image **I**.
- 2) Choose the 16×16 subimage in the upper left corner of the Gradient image and Circle-Frequency filtered image as the first block, which is separated into 4 subblocks named as cells. For each cell, we get a 9-dimensional histogram from the Gradient image and a 9-dimensional histogram from the CF filtered image. Then, histograms from the 4 cells are combined together, and normalized by the energy density of the block, and out come the $4 \times 9 \times 2 = 72$ dimensional feature representation of this block, which can be denoted as F_0 .
- 3) The block is set to traverse throughout the image with the step of 8 pixels, and for each step, we can obtain a 72-dimensional feature vector $F_i (i = 0, 1, \dots, 15)$. Finally, the combination of these F_i forms the final

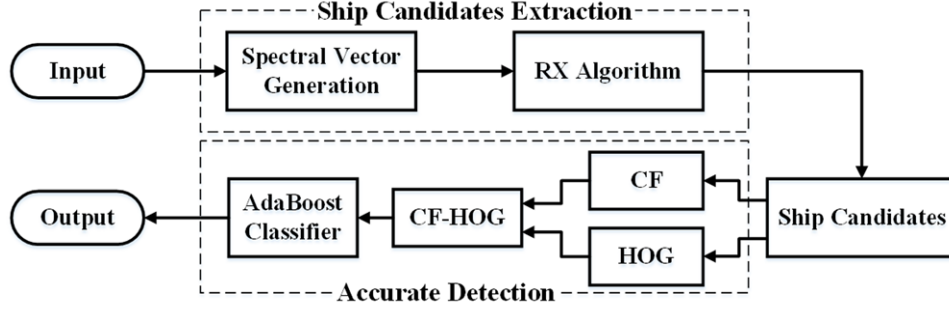


Fig. 8. The detailed outline of our method, including Spectral Vector Generation, RX, CF-HOG feature and AdaBoost classifier.

$16 \times 72 = 1152$ dimensional CF-HOG feature.

C. AdaBoost Algorithm

After feature extraction, AdaBoost algorithm [14], [15], [16] is applied to generate the hypotheses for ships. The principle of AdaBoost algorithm will be introduced briefly here. AdaBoost algorithm is aimed at boosting a weak learner into an arbitrarily accurate “strong” learning algorithm [17] by maintaining a set of weights over the training set. These weights are updated repeatedly so that more attention will be paid to the hard examples in the training set. After several iterations, a weighted majority vote of the weak hypotheses is generated, making the the final hypothesis. In many experiments, Adaboost algorithm has shown good performance and the ability to avoid overfitting [2], [9], [10], [24].

So far the whole process of our approach is introduced completely and the detailed outline of our approach is shown in Fig. 8.

IV. EXPERIMENTS

In this section, the experiments on the whole process of our approach are presented.

A. Dataset

Dataset1: the first dataset includes the training samples used to train the classifier. It totally contains 8400 samples, of which 5400 are positive ones and 3000 are negative ones. Notice that the positive samples contain ships in four orientations because Circle Frequency-Histograms of Oriented Gradients (CF-HOG) feature is not invariant to orientation and accordingly four classifiers are generated to detect ships in different orientations respectively. Fig. 9 shows some positive and negative samples. All these samples are 40×40 images, with the same size as the detecting window in classification. In fact the sizes of these samples and the detecting window are selected according to the smallest ships in the images. Since the small ships are about 40×10 , we choose 40×40 so that these ships can be covered by the detecting window. As to the larger ships, we choose to downsample the original image so that they can also be covered by 40×40 detecting window.

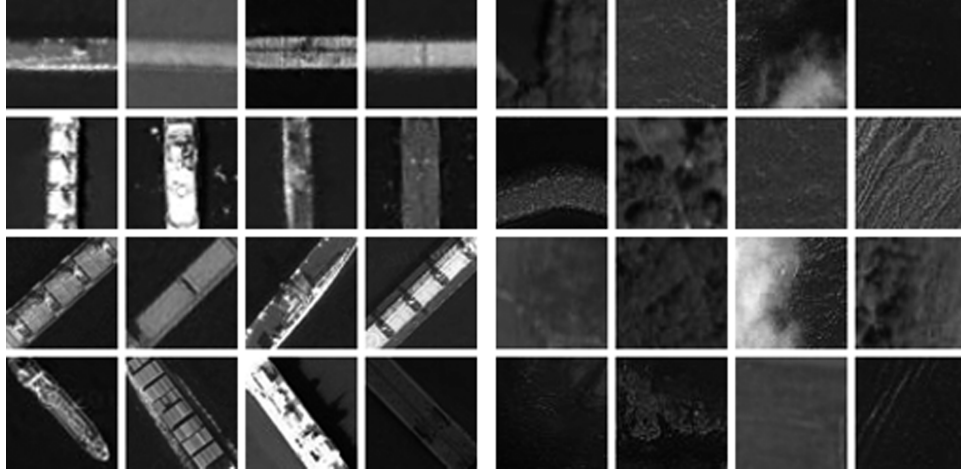


Fig. 9. The samples used to train the classifiers. The left columns show the positive samples in four orientations. The right columns show the negative samples.

Dataset2: dataset2 is essentially the same as Dataset1 but it is divided into two parts: 4200 training samples and 4200 testing samples. Dataset2 is mainly used to evaluate the performance of CF-HOG as well as some other feature sets such as Histograms of Oriented Gradients (HOG), Local Binary Pattern (LBP) and so forth. The 4200 training samples are first used to generate the classifiers. Then these produced classifiers are implemented to the 4200 testing samples so that the detection rate and the false alarm rate can be computed to evaluate the classifiers respectively.

Dataset3: dataset3 includes 50 large images from Google-Earth to evaluate our approach comprehensively. The sizes of these images range from 3000×3000 to 5000×5000 and their resolutions are all 1 m. In these images, the smallest ships are about 40×10 and the biggest ships are about 300×50 . The images are under different illuminations and contain different clutters such as clouds, waves and small islands.

B. Parameter Selection for RX

In this part, the experiments on the parameters of “Reed-Xiaoli” (RX) are presented. In fact, the original RX does not have any parameters. Here the parameters refer to the spectral dimension when generating spectral vectors and the size of the sliding window.

1) *Spectral Dimension*: spectral dimension is the dimension of the spectral vector which is produced by shifting the pixels in a $k \times k$ window. According to the analysis in section II, too large or too small spectral dimension will violate the conditions of RX and could cause poor performance. To select a proper spectral dimension, we test this parameter from 1×1 to 10×10 . The resultant images of the size 700×700 are presented in Fig. 10.

In Fig. 10, the two images in the left column are the original images and those in the right columns are the results of RX with different spectral dimensions: 3×3 , 5×5 , 9×9 , from left to right. When the spectral dimension is 3×3 , as is shown in the second column of Fig. 10, the performance is not good because the middle areas of

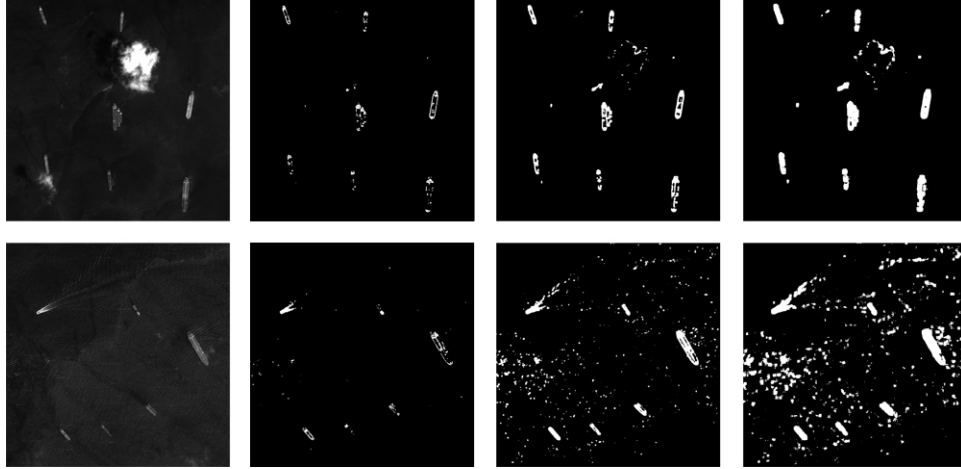


Fig. 10. The results of RX with different spectral dimensions. The two images in the left column are the original images and those in the right columns are the results of RX with different spectral dimensions: they are 3×3 , 5×5 , 9×9 , from left to right.

bigger ships are removed and only their edges are preserved. When the spectral dimension goes up to 5×5 , the performance gets better. When the spectral dimension is 9×9 , the upper image also gives good result. However the lower image produces a lot of false alarms. In fact, the lower image is more complicated and the waves in it are stronger. So when the spectral dimension is 9×9 , much noise is imported into the spectral vectors, which leads to the poor performance. In our approach, the spectral dimension is finally set to 5×5 .

2) *Sliding Window*: the sliding window is used mainly to guarantee the gaussianity of the data being processed by RX and its size will also affect the results of RX. The size of sliding window is tested from 300×300 to 1200×1200 . Fig. 11 shows the results of RX with different sliding windows. In Fig. 11, the two images on the left are the original images and those in the right columns are the results with different sizes of sliding windows: 300×300 , 700×700 and 1200×1200 , from left to right.

It can be seen from Fig. 11 that the results achieved are similar to those in Fig. 10. That is to say the size of the sliding window has little impact on the first image, in which the background is relatively simple. However, as to the second one, the size of the sliding window has great influence on it. Specifically speaking, for the second image, when the size of the sliding window is 300×300 , some ships are missed. But when it is 1200×1200 , too much false alarms are detected. This phenomenon also validates the analysis in section II that too small or too large sliding window will lead to poor performance. Finally this parameter is set to 700×700 in our approach.

C. RX Evaluation

In this part, the results of “Reed-Xiaoli” (RX) when applied to low contrast images and the overall performance of RX are presented.

1) *Ships with Low Contrast*: in the experiments, we test our approach on ships with low contrast because such ships are usually hard to detect. Besides, according to the appearances of ships and background, it seems that some

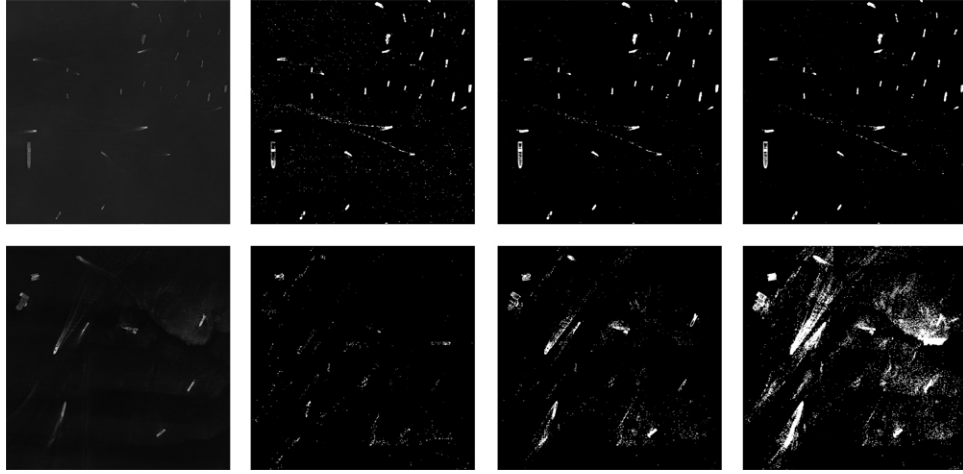


Fig. 11. The results of RX with different sliding windows. The two images in the left are the original images and those in the right columns are the results with different sliding windows. The sizes of these windows are 300×300 , 700×700 and 1200×1200 , from left to right.

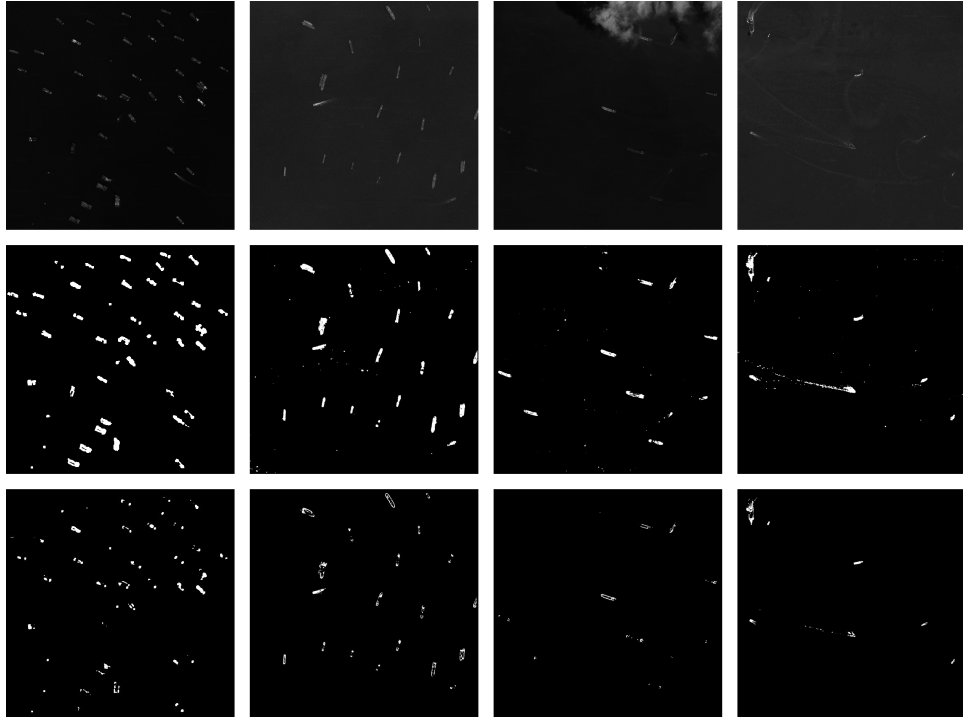


Fig. 12. The results of ships with low contrast. The upper row shows the original images; the middle row shows the results of RX and the lower row shows the results of the variance based method.

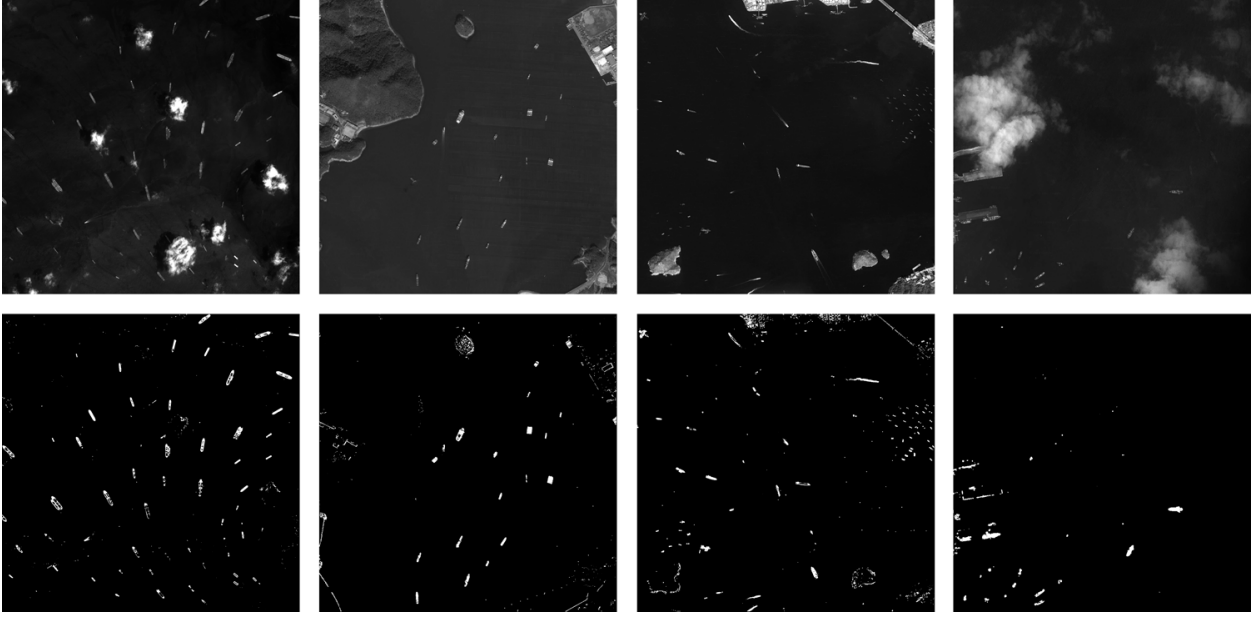


Fig. 13. The results of RX algorithm. The first row includes four 5000×5000 panchromatic images and the second one includes their results of RX.

ships can be easily distinguished from background by the variances of the pixels near ships. Therefore, we also compute the variance of each spectral vector to make a comparison with RX algorithm and this method will be named as “variance based method”. Fig. 12 presents the results of RX as well as the variance based method. In Fig. 12, the upper row shows the original images; the middle row shows the results of RX and the lower row shows the results of the variance based method. It can be seen from Fig. 12 that, for RX algorithm, these ships can also be detected despite of their low contrast. However, for the variance based method, the results are not so good and some ships are missed. In fact, for the clear ships, variance can indeed distinguish them from background. However, when dealing with the low-contrast ships, the variance will not be discriminative enough, especially for the low-contrast ships near some high-contrast objects (for example, a bright ship), because such low-contrast ships will be dwarfed or suppressed by the high-contrast objects during the normalization procedure (the procedure which normalizes the resultant image to 0-255). The reason why RX can outperform the variance based method could be that: by computing the statistics of background (the covariance matrix and the mean), RX essentially takes some global information of the window into consideration, which diminishes the suppression of the low-contrast ships while enhancing the separability between ships and background. While the variance is computed over a single spectral vector, thus, it only contains the local information.

2) *Overall Performance of RX*: in this part, the overall performance of RX is presented in Fig. 13. The upper row of Fig. 13 includes four 5000×5000 panchromatic images and the lower row includes the results of RX.

From Fig. 13, we can see that most ships can be detected by RX in the four images. Furthermore, in the first and the fourth image, the areas of cloud can be removed but their edges are liable to be preserved. As to the second

and the third one, some islands can be removed but others may not. Also, some areas on shore can be eliminated but some others can not. Generally, simple and smooth areas tend to be removed while complex areas such as the edges of clouds and islands, the artificial objects on shore are more likely to be preserved. However, from a global perspective, RX can detect ships efficiently while eliminating large areas of background. This can greatly accelerate the whole process and reduce pressure for the subsequent accurate detection stage.

D. Parameter Selection for CF-HOG

In this section, the details of the Circle Frequency-Histograms of Oriented Gradients (CF-HOG) feature as well as the impact of the parameters are presented. The parameters include the radius of the circle, the frequency bins (the constant c in (8)) in the process of Discrete Fourier Transform (DFT) as well as the gray bins when quantifying the filtered image.

In the experiments, the precision-recall graph [34] is used to visualize the performance of CF-HOG. Precision-recall graph can visualize and compare the performance of the classifiers and, indirectly, visualize and compare the performance of the feature sets. Precision and Recall (also known as sensitivity) are defined as:

$$\begin{aligned} \text{Precision} &= \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \\ \text{Recall} &= \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}. \end{aligned} \tag{9}$$

In (9), true positives refer to the positive samples which have been classified as positive correctly. False positives refer to the negative samples which have been classified as positive erroneously. False negatives refer to the positive samples which have been classified as negative erroneously. In brief, Precision reflects how many correctly classified samples are there in all samples which have been classified as positive. While, Recall reflects how many correctly classified samples are there in all positive samples. Therefore, we hope that Precision and Recall both have large values and, accordingly, the upper-right curve in a graph is the best. More detailed information on precision-recall graph can be found in [34]. Then the details of CF-HOG will be presented. Notice that we use the positive samples in one orientation as well as the negative samples in Dataset2 to draw the precision-recall graphs.

1) *Radius of Circle Frequency Filter*: The radius of Circle Frequency filter refers to the radius of the circle in Fig. 6. Since the ships in the training samples are about 40×10 , we test the radius from 5 to 15. The results are presented in Fig. 14.

From Fig. 14, it can be seen that when the radius is 7 the performance is the best. This can be explained as follows: if we assume that the width of a ship is 10, then the circle with a radius of $5 \times \sqrt{2} \approx 7$ can be equally divided into four parts by the two borderlines of the ship. Consequently, the series along this circle will be most approximate to the 2-cycle sinusoidal signal. Therefore, this radius does best in grasping the shape information and its performance is the best.

2) *Frequency Bin*: Frequency bin refers to the constant c in (8) in the process of DFT. The analysis in section III reveals that this parameter should be set to 4 in theory and we test it from 2 to 8. Fig. 15 shows the results.

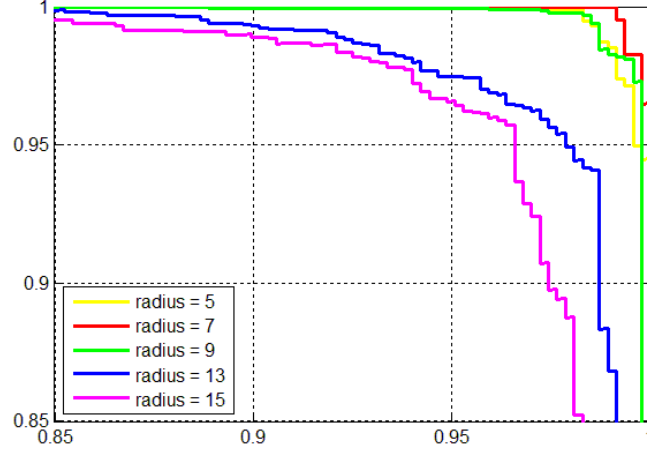


Fig. 14. The results with radiuses ranging from 5 to 15.

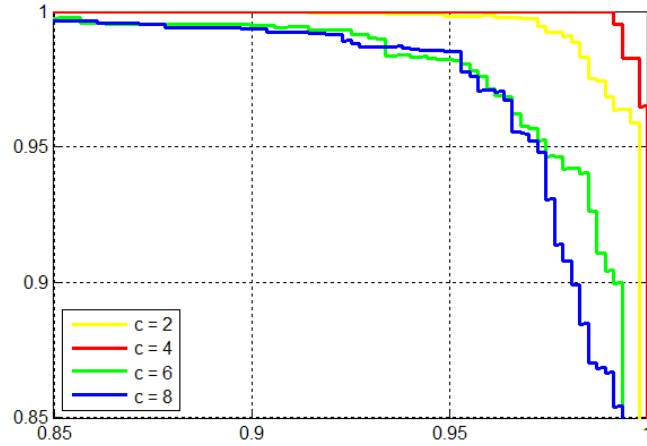


Fig. 15. The results of different frequency bins ranging from 2 to 8.

In Fig. 15, it can be seen that when $c = 4$ the performance is the best, which accords with our theoretical analysis. Frequency bin determines the outputs of CF filter greatly. When $c = 4$, the body areas of a ship will give larger outputs while other areas will give the smaller ones. When $c = 2$, as Fig. 15 shows, the performance is also not bad. One reason could be that when $c = 2$, the larger outputs come from the two borderlines of the ship, which can also grasp the shape information to some degree. However, when $c = 2$, the information in CF feature is similar to what is in HOG, so the CF-HOG contains redundant information and the performance is not the best. Notice that $c = 4$ is only suitable for ships due to their special shape. If the targets to be detected are other objects such as planes, the parameter could be changed according to the shape.

3) *Quantification for Filtered Image*: To extract the histogram of the filtered image, some quantification techniques must be applied to divide the gray values into several gray bins since the outputs of CF filter span a wide range. The number of the gray bins could affect the performance greatly. A larger number of the gray bins will

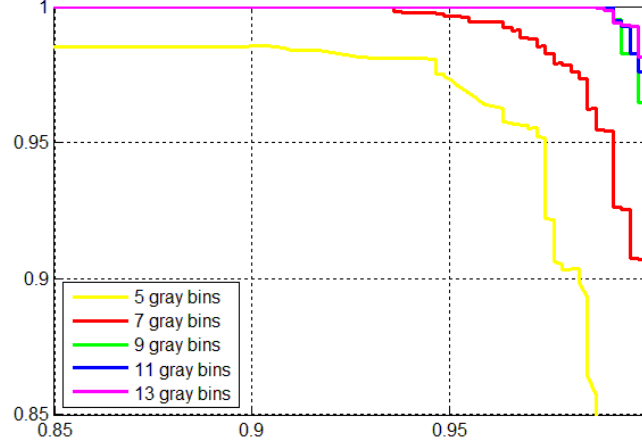


Fig. 16. The results of RX with different gray bins. The gray bins are 5, 7, 9, 11, 13.

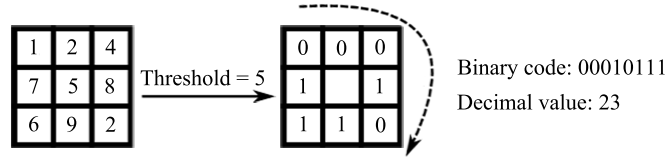


Fig. 17. The extraction of LBP. The left part is a pixel and its neighbors in an image and the right one is the binary image dichotomized by a threshold.

make the produced histogram more sophisticated but will result in a higher dimension of the feature set, while a smaller number of the gray bins will do the opposite. To select a proper number of the gray bins, this parameter is tested from 5 to 13 as is shown in Fig. 16.

As Fig. 16 shows, when the number of gray bins is smaller than 9, the performance is bad. As it increases, the performance is improved greatly. However, when the number of gray bins is larger than 9, the performance almost remains the same. Therefore, considering the dimension of the feature set, the number of the gray bins is set to 9.

E. Feature sets comparison

In this section, some other feature sets such as Histograms of Oriented Gradients (HOG) [8], Local Binary Pattern (LBP) [32] and their various extensions are compared with the Circle Frequency-Histograms of Oriented Gradients (CF-HOG). Both of HOG and LBP have many applications in the field of object detection [1], [8], [19], [43]. HOG is a combined histogram of the first order derivatives of an image. With the help of some special normalization techniques as well as some cells and blocks, HOG is invariant to illuminations and tiny shape changes. If we extract the first and second order derivatives of an image, then we get a new extension: R2-HOG. Furthermore, if we replace the square cells with circular cells, then we get another extension: C-HOG.

LBP is also compared in the experiments. LBP is a kind of coding of the spatially adjacent pixels. Fig. 17 shows the procedure of LBP extraction. First, for one pixel in the image, its neighbors are dichotomized into 0 and 1 by

TABLE I
THE RESULTS OF DIFFERENT FEATURES

Feature	Detection rate(%)	False alarm rate(10^{-6})
CF-HOG	96.07	2.449
HOG	95.32	3.126
C-HOG	95.21	2.729
R2-HOG	95.47	2.762
LBP _{riu2}	90.91	9.970
LBP _{u2}	91.33	7.319
LTP _{riu2}	93.07	7.174
LTP _{u2}	94.04	5.937

the gray level of this pixel. Then these 0 and 1 series are coded into a binary code, for example, 00010111, as shown in Fig. 17. Thus the decimal value of the binary code can reflect the spatial information to some degree. In this way, for each pixel in the image, a decimal value can be calculated and the histogram of the resultant image forms the feature vector. Furthermore, if there are two thresholds, the pixels in the neighborhood will be cut into 0, 1 and 2 and we get Local Ternary Pattern (LTP). LBP as well as LTP have some extensions: U2-LBP and U2RI-LBP, U2-LTP and U2RI-LTP. U2-LBP is extracted by attributing the binary codes with more than two times' 0/1 changes to one class and forcing their decimal values to be a same value because the binary codes which change frequently (for example, 01010101) have tiny probabilities of occurrence. And U2RI-LBP is extracted by shifting the binary code for 8 times and selecting the smallest decimal value so that the similar spatial patterns with different orientations will give the same output, which makes U2RI-LBP invariant to orientation. So do U2-LTP and U2RI-LTP. More detailed information about LBP can be found in [32].

In the experiments, we first use Dataset1 to train classifiers based on different features. The produced classifiers are then implemented to Dataset3 to detect ships in 1-m resolution images of the sizes ranging from 3000×3000 to 5000×5000 . Afterwards, the numbers of the detected ships, missed ships and detected false alarms are counted to compute the detection rate and the false alarm rate (also called false positive rate). They are computed as the formulas below:

$$\begin{aligned}
 \text{detection rate} &= \frac{\text{number of correctly detected ships}}{\text{number of all ships}} \\
 \text{false alarm rate} &= \frac{\text{number of detected false alarms}}{\text{size of the image}}.
 \end{aligned} \tag{10}$$

Table I shows the results of different features. It can be seen from Table I that CF-HOG outperforms other features in both detection rate and false alarm rate. The performances of C-HOG and R2-HOG are almost the same as HOG. As to LBP and its extensions, U2-LTP achieves the best performance.

To make a further glimpse of these features, we draw the precision-recall curves of all feature sets. Because U2RI-

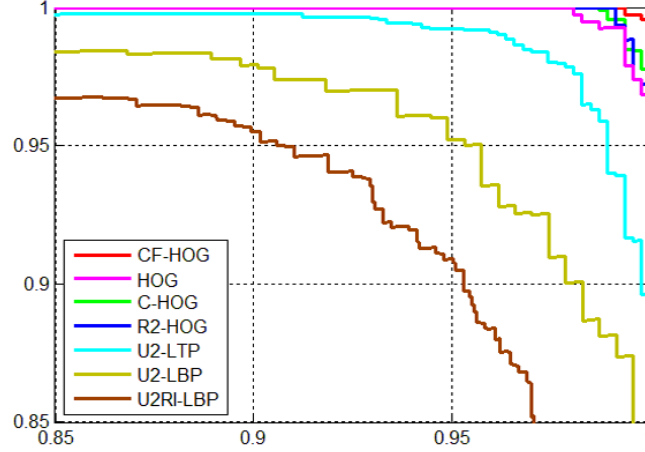


Fig. 18. The detailed precision-recall curves of all features.

LBP and U2RI-LTP are invariant to orientation, we use all positive samples of different orientations in Dataset2 to train the classifiers. For U2RI-LBP and U2RI-LTP, each of them generates only one classifier which can detect ships of different orientations. However, as to other features, we train four classifiers for each of them so that ships of different orientations can be detected respectively by the four classifiers. Fig. 18 shows the precision-recall curves of different features.

It can be seen from Fig. 18 that the results achieved here accord with the results in Table I. CF-HOG is better than HOG, LBP and their extensions. The results of HOG, C-HOG and R2-HOG are similar and their performances are better than LBP and its extensions. Among LBP, LTP and other extensions, U2-LTP achieves the best performance. The results indicate that Circle-Frequency (CF) feature can really reinforce HOG to some degree, at least in the realm of ship detection.

As to LBP, LTP and their extensions, U2-LTP is the best. Its detection rate is a little lower than that of CF-HOG but the false alarm rate is a little high. This higher false alarm rate is probably because of the limitation in a parameter of LTP. The parameter is a radius which determines the coverage of the local descriptor. Theoretically, the radius can be arbitrarily large, while in practice, the radius longer than 3 and the number of sampling points more than 24 will produce an original LTP feature with a quite large dimension (larger than 3^{24}), which makes it a very difficult or even impossible task for our computer to generate the U2-LTP as well as other extensions. In [42] and [43], the radiuses used to extract LTP are all smaller or equal to 3. Therefore, due to the limitation in computer memory, the radiuses are also smaller or equal to 3 in our experiments. This limitation in the radius makes LTP can only grasp the information in a small neighbourhood. For 5 to 10-m resolution images tested in [42], LTP can effectively discriminate between ships and false alarms. However, in our experiments, the images are all with 1-m resolution and the ships in them are much bigger.

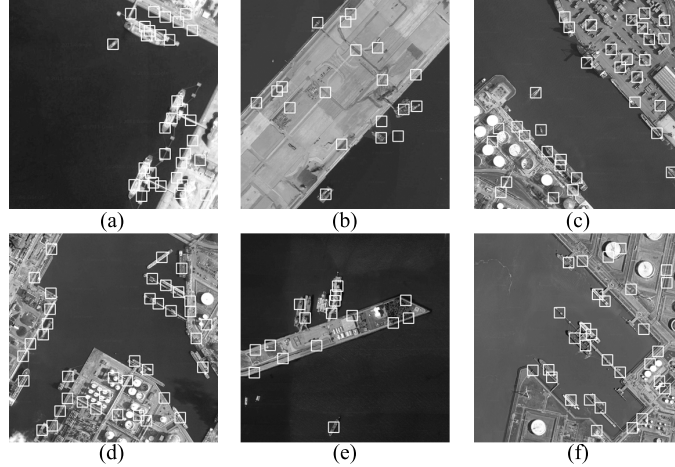


Fig. 19. The results of ships near land. The six scenes are all near land and contain lots of artificial objects.

F. Ships Adjoining Land

Besides the work mentioned above, we also did some experiments focusing on the ships adjoining land, which are usually difficult to detect. In this special situation, there is no doubt that the straight borderlines will have an effect on the Circle Frequency-Histograms of Oriented Gradients (CF-HOG) descriptor, which may make our method fail. Fig. 19 shows the results of our method. It can be seen from Fig. 19 that most of the ships near land can also be detected successfully but the false alarm rate is extremely high. There are two reasons. The first one is that the objects on shore are so complex that they can hardly be removed by the predetection stage. Another one is that some artificial objects are very similar to ships and CF-HOG can not distinguish them from ships. This phenomenon also indicates that our approach is very susceptible to the artificial objects on shore.

V. CONCLUSION

After all the analysis above, the conclusion of this paper will be summarized as follows.

We investigate the problem of ship detection in high resolution optical panchromatic imagery and employ an approach involving a predetection stage and an accurate detection stage to detect ships in a coarse to fine manner.

In the predetection stage, the panchromatic optical image is first converted into a hyperspectral form and ship candidates are extracted by a hyperspectral algorithm, “Reed-Xiaoli” (RX). The transformation rearranges the adjacent pixels and turn the shape information and contextual information into a hyperspectral form, which increases the separability between ships and background. Then, through RX algorithm, the separability is further increased by projecting the ships while suppressing the background. The experiments show that the predetection stage can detect most ships while removing large areas of background, even for the ships with low contrast.

Afterwards, in the accurate detection stage, Circle-Frequency (CF) feature is provided with Histograms of Oriented Gradients (HOG) to detect ships accurately. This extra feature pays more attention to the gray values rather than

the gradients of an image. Furthermore, this feature is generated by the pixels along a circle rather than the closely adjacent pixels. Therefore, it contains some information different from HOG and can reinforce HOG to some degree. The experiments indicate that the combined feature is better than some other features such as HOG, LBP and their extensions in the field of ship detection.

Besides the good results achieved, some issues also exist to be further investigated. For example, when dealing with ships near land, our approach will give poor performance. Our further work will focus on these problems to strive for further improvement in this field.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and anonymous reviewers for their patience in reviewing this paper and their invaluable suggestions and comments.

REFERENCES

- [1] T. Ahonen, A. Hadid and M. Pietikainen, "Face recognition with local binary patterns," *M.S. thesis, Mach. Vis. Group, Infotech Oulu, Univ. Oulu, Oulu, Finland*, 2004.
- [2] R. E. Banfield, L. O. Hall, K. W. Bowyer and W.P. Kegelmeyer, "A Comparison of Decision Tree Ensemble Creation Techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 137-180, Jan. 2007.
- [3] F. K. Bi, B. C. Zhu, L. N. Gao, and M. M. Bian, "A Visual Search Inspired Computational Model for Ship Detection in Optical Satellite Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 4, pp. 749-753, July. 2012.
- [4] D. W. Burgess, "Automatic ship detection in satellite multispectral imagery," *Photogramm. Eng. Remote Sens.*, vol. 59, no. 2, pp. 229-237, 1993.
- [5] C. Corbane, F. Marre, and M. Petit, "Using SPOT-5 HRG data in panchromatic mode for operational detection of small ships in tropical area," *Sensors*, vol. 8, no. 5, pp. 2959-2973, May 2008.
- [6] C. Corbane, E. Pecoul, L. Demagistri, and M. Petit, "Fully automated procedure for ship detection using optical satellite imagery," in *Proc. SPIE-Remote Sensing of Inland, Coastal, and Oceanic Waters*, vol. 7150, 2008.
- [7] C. Corbane, L. Najman, E. Pecoul, L. Demagistri and M. Petit, "A complete processing chain for ship detection using optical satellite imagery," *Int. J. Remote Sens.*, vol. 31, no. 22, pp. 5837-5854, 2010.
- [8] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2005, pp. 886-893.
- [9] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 135-157, 2000.
- [10] H. Drucker and C. Cortes, "Boosting decision trees," in *Advances in Neural Information Processing Systems*, pp. 479-485, 1996.
- [11] B. Du and L. P. Zhang, "Random-selection-based anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 5, pp. 1578-1589, 2011.
- [12] B. Du, L. P. Zhang, D. C. Tao and D. Y. Zhang, "Unsupervised transfer learning for target detection from hyperspectral images," *Neurocomputing*, 2013.
- [13] K. Eldhuset, "An automatic ship and ship wake detection system for spaceborne SAR Images in coastal regions," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 4, pp. 1010-1019, 1996.
- [14] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation*, vol. 121, no. 2, pp. 256-285, 1995.
- [15] Y. Freund and R.E. Schapire, "Game theory, on-line prediction and boosting," in *Proc. 9th Annu. Conf. Computational Learning Theory (COLT)*, pp. 325-332, 1996.
- [16] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. System Sci.*, vol. 55, no. 1, pp. 119-139, 1997.

- [17] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771-780, 1999.
- [18] H. Goldberg and N. M. Nasrabadi, "A comparative study of linear and nonlinear anomaly detectors for hyperspectral imagery," in *Proc. SPIE.*, vol. 6565, 2007.
- [19] G. Heusch, Y. Rodriguez and S. Marcel, "Local binary patterns as an image preprocessing for face authentication," in *Proc. 7th Int. Conf. Autom. Face Gesture Recog. (FGR)*, 2006, pp. 9-14.
- [20] S. Kawato and J. Ohya, "Two-step approach for real-time eye tracking with a new filtering technique," in *Proc. Int. Conf. On System, Man and Cybernetics*, 2000, pp. 1366-1371.
- [21] S. Kawato and J. Ohya, "Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes," in *Proc. 4th IEEE. Int. Conf. on Automatic Face and Gesture Recognition.*, Mar. 2000, pp. 40-45.
- [22] S. Kawato and N. Tetsutani, "Circle-Frequency Filter And Its Applation," *Int. Workshop on Advanced Image Technology*, vol. 100, no. 616, pp.217-222, Feb. 2001.
- [23] S. Kawato and N. Tetsutani, "Real-time Detection of Between-the-Eyes with a Circle Frequency Filter", in *Proc 5th Asian Conference on Computer Vision. (ACCV)*, 2002, pp. 521-526.
- [24] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern Anal. Appl.*, vol. 1, no. 1, pp. 18-27, Mar. 1998.
- [25] H. Kwon, N. Nasrabadi, and S. Der, "Adaptive anomaly detection using subspace separation for hyperspectral imagery," *Optical Engineering.*, vol. 42, pp. 3342-3351, Nov. 2003.
- [26] D. Maesschalck, J. R. Delphine, and M. Desirel, "The Mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, pp.1-18, 2000.
- [27] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proc of the National Institute of Sciences of India*, vol. 2, no. 1, pp. 49-55, 1936.
- [28] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," in *IEEE. Singal Processing Magazine*, vol. 19, no. 1, pp. 29-43, 2002.
- [29] S. Matteoli, M. Diani and G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images," *IEEE. Aerospace and Electronic Systems.*, vol. 25, no. 7, pp. 5-28, 2010.
- [30] N. Nasrabadi, "Regularization for spectral matched filter and RX anomaly detector," in *Proc. SPIE-Multispectral, Hyperspectral, and Ultraspectral Imagery.*, vol. 6966, 2008.
- [31] N. Otsu, "A threshold selection method from gray-level histograms," in *Proc IEEE Trans. Sys., Man., Cyber.*, vol. 9, no. 1, pp. 62-66, 1979.
- [32] M. Pietikainen, "Image analysis with local binary patterns", in *Proc. SCIA*, vol. 3540, pp. 115-118, 2005.
- [33] C. H. Ping and G. Z. Wei, "A New Method to Detect Airplanes in Remote Sensing Image," *Signal Processing*, vol. 23, no. 4, pp. 539-543, Aug. 2007.
- [34] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness correlation," *School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE*, vol. 7, no. 1, 2007.
- [35] N. Proia and V. Pag, "Characterization of a bayesian ship detection method in optical satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 2, pp. 226-230, Apr. 2010.
- [36] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 10, pp. 1760-1770, Oct. 1990.
- [37] Z. W. Shi, W. Jun, Y. Shuo and J. Z. guo, "RX and its variants for anomaly detection in hyperspectral images," *Infrared and Laser Engineering.*, vol. 42, no. 3, pp. 796-802, Mar. 2012.
- [38] D. Stein, S. G. Beaven and L. E. Hoff, "Anomaly detection from hyperspectral imagery," *IEEE. Signal Processing Magazine.*, vol. 19, no. 1, pp. 58-69, 2002.
- [39] T. Wang, B. Du, and L. P. Zhang, "A kernel-based target-constrained interference-minimized filter for hyperspectral sub-pixel target detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 626-637, 2013.
- [40] Y. Wang and A. Liu, "A hierarchical ship detection scheme for high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 4173-4184, 2012.

- [41] Y. Xia, S. H. Wan and L. H. Yue, "A novel algorithm for ship detection based on dynamic fusion model of multi-feature and support vector machine," in *6th International Conference on Image and Graphics. (ICIG)*, 2011, pp. 521-526.
- [42] C. Zhu, H. Zhou, R. Wang, and J. Guo, "A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 9, pp. 3446-3456, Sep. 2010.
- [43] C. R. Zhu and R. S. Wang, "Local multiple patterns based multiresolution gray-scale and rotation-invariant texture classification," *Pattern Recognit.*, vol. 187, pp. 93-108, 2012.