Ship Rotated Bounding Box Space for Ship Extraction From High-Resolution Optical Satellite Images With Complex Backgrounds

Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang

***Abstract*—Extracting ships from complex backgrounds is the bottleneck of ship detection in high-resolution optical satellite images. In this letter, we propose a nearly closed-form ship rotated bounding box space used for ship detection and design a method to generate a small number of highly potential candidates based on this space. We first analyze the possibility of accurately covering all ships by labeling rotated bounding boxes. Moreover, to reduce search space, we construct a nearly closed-form ship rotated bounding box space. Then, by scoring for each latent candidate in the space using a two-cascaded linear model followed by binary linear programming, we select a small number of highly potential candidates. Moreover, we also propose a fast version of our method. Experiments on our data set validate the effectiveness of our method and the efficiency of its fast version, which achieves a close detection rate in near real time.**

***Index Terms*—Ship detection, ship extraction from complex backgrounds, ship rotated bounding box space.**

I. Introduction

SHIP detection in remote sensing images is an important issue and has wide applications such as fishery management, vessel traffic services, and naval warfare [1].

The main motivation for this work is to address two typical challenges for ship detection in optical remote sensing images with complex backgrounds. First, an effective and fast method is much in demand when big data meet a platform with limited computation capability. Second, it is difficult to extract ships from complex backgrounds as illustrated in Fig. 2. Ships in remote sensing images are strip-like, but they are often not stand-alone, unlike the objects in natural images. Furthermore, different ships or ships in varying time have amazing varieties of shapes and appearances, which further increases the complication of ship detection.

Existing works on ship detection can be roughly divided into three groups. The first group is based on pixelwise (or pixel-based regions) labeling [2]–[5]. Due to the combinatorial explosion, the search space of this way might be too large, making it infeasible to work on it. To reduce the search space, these methods were designed to perform pixelwise labeling first and then group foreground pixels into regions by incorporating prior knowledge. For example, Yang *et al.* [3] classified each pixel into two classes by sea surface analysis based on statistics. Tang *et al.* [5] grouped pixels by a threshold obtained from a statistical Gaussian model and then connected positive pixels into regions. The second group is based on bounding box (including image patch) labeling [6]–[10]. For example, Cheng *et al.* [8] detected objects (including ships) in remote sensing imagery using a discriminatively trained mixture model based on sliding windows. Han *et al.* [9] detected typical objects in remote sensing images based on square sliding windows in varying sizes by a detector trained using a new weakly supervised learning framework. However, only labeling bounding boxes is not accurate enough for ship localization; thus, it is unsuited for ship classification. The third group is based on rotated bounding box labeling [11], [12]. For example, Liu *et al.* [11] detected ships by shape analysis, including ship head detection and ship body extraction, after water and land segmentation and removed false alarms by labeling rotated bounding box candidates.

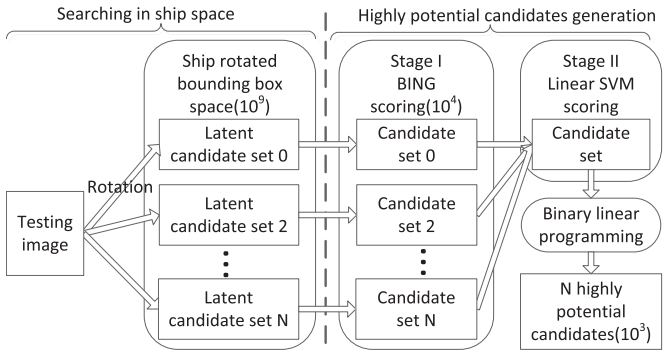


Fig. 1. Flowchart of our method.

Despite some effectiveness, most existing works are unable to tackle the aforementioned challenges. In particular, the traditional methods [11], [12] for ship detection are hand-crafted in a step-by-step manner, which are not robust.

The goal of our work is to robustly generate a small number of candidates that precisely cover ships of any type in complex backgrounds. The framework is demonstrated in Fig. 1. Our contribution is twofold. On the one hand, we construct a nearly closed-form ship rotated bounding box space and analyze the possibility of covering ships by the space. On the other hand, to select a small number of highly potential candidates from

Manuscript received February 17, 2016; revised April 20, 2016; accepted May 7, 2016. This work was supported by the National Natural Science Foundation of China under Grant 91338202.

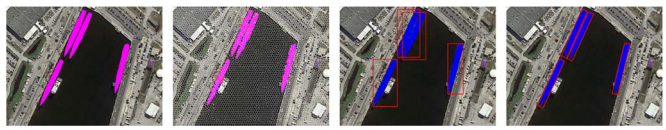
The authors are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lubin.weng@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2016.2565705

(a) (b)

Fig. 2. Various complex backgrounds.



(a) (b) (c) (d)

Fig. 3. Labeling ways. (a) Pixel. (b) Superpixel. (c) Bounding box. (d) Rotated bounding box.

this space, we design a two-cascaded linear model followed by binary linear programming.

II. Ship Rotated Bounding Box Space

In Section I, we list two challenges in ship detection. In fact, complex backgrounds exist in both sea and inshore area as the examples shown in Fig. 2. Things around ships include waves, clouds, islands, other ships, docks, objects on land, etc.

Sea and inshore ship detection face the same bottleneck: ship extraction from complex backgrounds. Sea ship detection can often achieve better performance, mainly because the complex backgrounds are less common in sea.

A. Covering Ships

It is a tough task to hit the target when it is difficult to improve the hit rate. To address this issue, we propose to raise the number of trials. Specifically, to detect a ship o in image I, the usual practice is to try to get a small number of windows or regions (false alarms might also exist) to cover o. We define these windows or regions as candidate set $\mathcal{C}$ for the subsequent strong classifiers. The probabilities of covering a ship with N and M ($M \geq N$) candidates ($\mathcal{C}^N$ and $\mathcal{C}^M$, respectively) are

$$P(\mathcal{C}^N) = 1 - \prod_{i}^{N}(1 - P(\mathcal{C}_i)) \tag{1}$$

$$\begin{aligned} P(\mathcal{C}^M) &= 1 - \prod_{i}^{N}(1 - P(\mathcal{C}_i)) \prod_{i=N+1}^{M}(1 - P(\mathcal{C}_i)) \\ &= P(\mathcal{C}^N) + (1 - P(\mathcal{C}^N))P(\mathcal{C}^{M-N}) \geq P(\mathcal{C}^N) \end{aligned} \tag{2}$$

where $\mathcal{C}^N$ is a subset of $\mathcal{C}^M$, $\mathcal{C}^{M-N} = \mathcal{C}^M - \mathcal{C}^N$, and $P(\mathcal{C}_i)$ is the probability of ith candidate covering o. Equation (2) indicates that, when the candidate set is expanded, it can cover o with higher probability. Here, "a candidate covering o" means that the intersection over union (IoU) metric between them exceeds a fixed value $0 \leq \eta \leq 1$ ($\eta = 0.5$ in this letter).

According to the aforementioned analysis, for the ship detection task, on the one hand, we need to evaluate $P(\mathcal{C}_i)$ for each latent candidate in Ω. This goal can be achieved by scoring for all latent candidates using various strategies. Ideally, a higher overlap leads to a higher score. On the other hand, to raise $P(\mathcal{C}^N)$, we need to select proper N candidates from latent candidate space Ω.

However, before scoring for latent candidates, we need to choose a proper latent candidate space. As will be discussed in the next section, the common bounding box space is not appropriate for ship detection.

TABLE I
Potential Caps of Four Labeling Ways

Labeling way	Recall @ η=0.5	AR	Search space
Pixel	100%	100%	Ω (Too large)
Superpixel	100%	90.8%	Ω (Too large)
Bounding box	31.8%	13.4%	Ω ($\sim 10^{11}$)
Rotated bounding box	100%	75.8%	Ω ($\sim 10^{13}$)

B. Ship Rotated Bounding Box Space Construction

Here, we compare four labeling ways for ship detection, including labeling pixels [2], [3], [5], superpixels [13], bounding boxes [6]–[10], and rotated bounding boxes [11], [12]. All of the ways except labeling superpixels have been introduced in Section I. Similar to labeling pixels, labeling superpixels means classifying various superpixel-based regions and also suffers from a large search space because of combinatorial explosion. To our best knowledge, no successful methods have been reported on classification in the initial search space of the two ways. In Fig. 3, we provide examples to highlight the differences between the four labeling ways. The superpixels are generated using the method in [14].

In Table I, supposing that each candidate is correctly labeled, we evaluate the potential caps of the four labeling ways. It can be seen that pixel-based and superpixel-based ways have high AR but too large search space. In this letter, AR denotes the average recall between IoU 0.5 to 1, which rewards both high recall and good localization and correlates surprisingly well with the performance of the followed detector [15]. Labeling bounding boxes has a proper search space but very low AR. Labeling rotated bounding boxes, the one that we choose in this letter, is a tradeoff between them. To our best knowledge, we are the first to locate ships based on labeling rotated bounding boxes, which is different from [11] and [12], while they locate ships before getting rotated bounding boxes.

By labeling rotated bounding boxes, we can successfully cover all ships in complex backgrounds and avoid the difficulty of extracting objects. However, it is still inefficient to evaluate $P(\mathcal{C}_i)$ for all of the latent candidates in the rotated bounding box space $\Omega_R(O(10^{13}))$. Here, we introduce quantization technology to reduce the search space.

By scale/aspect-ratio quantization, we obtain a nearly closed-form search space, which is beneficial for dealing with scale variance. As shown in Fig. 4, we cluster the sizes of all of the training samples in our data set by k-means (we set $k = 36$, in view of detection rate, AR, and computation time). After linking the origin to two remote points by two lines, respectively, it can be seen that most of the points lie along the two lines, indicating that the size space can be roughly viewed as closed form. For example, if we reduce a 600×90 ship to one third of the original size, the resulting 200×30 version is still close to one of the cluster centers.

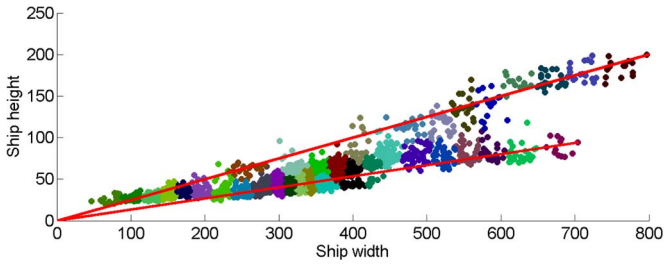


Fig. 4. Nearly closed-form scale quantization. Cluster ship sizes in training data by k-means and parameter $k = 36$.

Besides scale/aspect-ratio quantization, we still need to quantize rotation angles between 0° and 180°. Experimentally, we choose 4° as the rotation interval for our method and 12° for its fast version.

Above all, we get a reduced search space Ω_S of Ω_R, which we call "ship rotated bounding box space." For example, the number of latent candidates in Ω_S for a 1200 × 800 image is about 10^9. As a result, the size of Ω_S makes it possible to use fast scoring methods. Therefore, we can efficiently select a small number of highly potential candidates $\Omega_C(10^3)$ from Ω_S, which enables the usage of complex classifiers followed. This space is specialized for ships in remote sensing images.

III. Candidate Scoring and Selection

Based on ship rotated bounding box space (Ω_S), we evaluate $P(\mathcal{C}_i)$ by scoring for $\mathcal{C}_i$ using a two-cascaded linear model, and then, we can further select N candidates by binary linear programming. Ideally, the more overlap between the candidate and any ship, the higher score we should get.

As shown in Fig. 1, a testing image is rotated with fixed interval between 0° and 180°. Latent candidate sets in ship space are collected. In stage I, using a BING linear model (Section III-A), we score for each candidate in these sets and discard the ones with low scores. Then, we merge the obtained subsets into Ω_I. In stage II, based on the scores obtained in stage I and some additional features, we use a linear model of (3) to score for each candidate in Ω_I. Finally, we select N candidates from Ω_I using binary linear programming.

A. Candidate Evaluation by a Two-Cascaded Linear Model

We score for latent candidates in Ω_S by a two-cascaded linear model

$$s_l = \langle w_{II}, \Phi_l \rangle \tag{3}$$

$$l = (a, i, x, y) \tag{4}$$

$$\Phi_l = [B_l, E_l, W_l, C_l] \tag{5}$$

$$B_l = \langle w_I, g_l \rangle \tag{6}$$

where $w_{II} \in R^9$ and $w_I \in R^{64}$ are model parameters, and s_l, Φ_l, and l are the candidate score, feature vector, and candidate location, respectively. In (4), a, i, x, and y are the rotation angle, size index, and coordinate of the center point. In (5), $B_l \in R$, $E_l \in R^3$, $W_l \in R^4$, and $C_l \in R$ are the BING ship model score, edge distribution feature, rotated bounding box contour feature, and context feature (as will be detailed later). Vector $g_l \in R^{64}$ in (6) is the normed gradient feature.

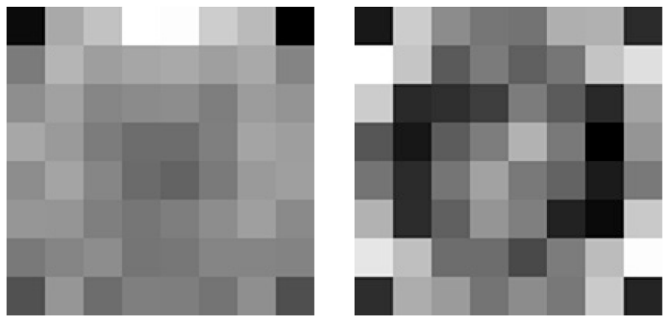


Fig. 5. BING objectness model (left) and our model (right).

Equations (6) and (3) represent the linear models of stages I and II, respectively. They are cascaded into a two-layer model.

In stage I, the linear model is trained in the same way as stage I of BING [16] but on our data set. When resizing ships in remote sensing images to a fixed size (e.g., 8 × 8, chosen as a matter of experience and computation convenience), the norm of the corresponding image gradient exhibits common evidence of ships in such an abstracted view. The learned linear model $w_I \in R^{64}$ in stage I is shown in Fig. 5. Our model is different from BING objectness model (the left one). Compared with stand-alone objects in nature images, most ships have salient contours, islands, robust ship heads, and smooth areas around ship islands. Accordingly, our model has high weights in the four sides and center, some highest values in the left and right sides, but low weights around the center pixels.

In stage II, we use (3) to calculate the final scores. The linear model $w_{II} \in R^9$ is trained by linear support vector machine. However, it is difficult to select good candidates by linearly stretching the 1-D score obtained in stage I as described in [16] because our latent candidates of each scale have different rotation angles and their scores no longer stand in a cluster (see analysis in Section IV-A). Therefore, we further explore some additional features: edge distribution, rotated bounding box contour, and context feature.

Edge distribution (E_l): Edges always distribute differently in the outside, middle, and center areas of a ship. Taking into account efficiency, we approximate edge strength by computing convolution on the 8 × 8 normed gradient map in stage I. We define the edge distribution feature as $E_l = [E_l^{\text{out}}, E_l^{\text{mid}}, E_l^{\text{center}}]$.

Rotated bounding box contour (W_l): We define $W_l = [W_l^0, W_l^1, W_l^2, W_l^3]$, where W_l^0, W_l^1, W_l^2, and W_l^3 are the sum of all of the absolute operator values on the head, tail, left, and right side of the rotated bounding box with location l, respectively, divided by the perimeter of the corresponding side. The operator that we use is [−1, −1, 0, 1, 1], which operates on each pixel of edges of rectangles along the perpendicular direction. In practice, we resize the image to half of its original size in both height and width, and use an integral image for speeding up. This work is related to [11].

Context information (C_l): We define $C_l = S_l^s / S_l^f$, where S_l^s is the sea area in the rotated bounding box and S_l^f is the foreground area. We fix sea pixels by Otsu algorithm in [5].

For stage II, we select candidates in Ω_I as positive training samples whose IoU with any ship is bigger than η, others as negative ones.

B. Highly Potential Candidate Selection

Finally, we design a method to select N highly potential candidates. It is not a good idea to select top N high-score candidates directly. In stage I, we select the rectangles with high scores from local patches by nonmaximal suppression as [16]. However, it cannot suppress candidates between different rotation angles and sizes. To address this issue, we formulate the candidate selection as a new binary linear programming problem, which is different from [11]

$$\begin{aligned} &\max_{x_i} \sum_{i=1}^{H} s_i x_i \\ \text{s.t.} \quad & x_i \in \{0,1\}, \sum_{i=1}^{H} x_i = N, \\ & x_{k1} + x_{k2} + \cdots + x_{kn} \le 1, \\ \text{where} \quad & \{x_{k1}, \ldots, x_{kn}\} = \Omega_k, I(\omega_{ku}, \omega_{kv}) > \eta, \\ & 1 \le u, v \le n, I(\omega_1, \omega_2) = \frac{\omega_1 \cap \omega_2}{\omega_1 \cup \omega_2} \end{aligned} \tag{7}$$

where N is a fixed value, H is the size of Ω_I, and Ω_k is the kth subset of Ω_I in which $I(\omega_{ku}, \omega_{kv})$ of any two candidates ω_{ku} and ω_{kv} is larger than a fixed value η. However, the computation time of (7) is inefficient because we need to do many times of $I(\omega_1, \omega_2)$.

For this reason, we develop two ways to accelerate the computation: first, we approximately compute Ω_k in local grid of location, angle, and area value. Second, we compute $I(\omega_1, \omega_2)$ of two rotated bounding boxes as

$$I(\omega_1, \omega_2) = \frac{\omega_1 \cap \omega_2}{\omega_1 \cup \omega_2} = \frac{\sum_{S_i \in \Delta} S_i}{S_1 + S_1 - \sum_{S_i \in \Delta} S_i} \tag{8}$$

where S_1 and S_2 are the areas of rotated bounding boxes ω_1 and ω_2, respectively, and Δ is a triangle set in which each triangle's vertexes are composed of a fixed center point and two adjacent vertexes of a convex polygon. The vertexes of this polygon are composed of inner vertex points or intersection points of ω_1 and ω_2. Inner points are defined as vertexes of ω_1 in ω_2 (or vice versa). The fixed center point can be any point in the polygon. On a single laptop CPU, our method is about 450 times faster than the one in [17].

C. Algorithm Variants

For the proposed method, we develop two versions. The first one, whose rotation interval is set as 4°, includes stages I and II shown in Fig. 1. The second one called "Fast version," whose rotation interval is set as 12°, is composed of stage I and the "Binary linear programming" module.

IV. Results and Evaluation

We evaluate our methods on our data set HRSC2016.[1] Our data set contains 1070 images collected from Google Earth. The image sizes range from 300 × 300 to 1500 × 900, and most of them are larger than 1000 × 600. In our data set, there are more than 25 types of ships with large varieties of scale, position, rotation, shape, and appearance, which are suitable for evaluating methods of ship detection. After excluding submarines, hovercrafts, and those annotated as "difficult" label, our training, validation, and testing data sets contain 443 images with 1207 samples, 183 images with 541 samples, and 444 images with 1071 samples, respectively.

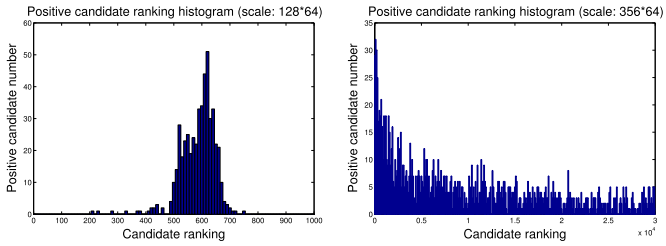


Fig. 6. Score ranking histogram of positive candidates with the same scale in stage I. The results are obtained by BING on VOC2007 (left) and our method (right), respectively.

We compare our methods with the state-of-the-art ship detection approaches[2] [5], [11], BING [16] and its variant "BING based on our ship space." Please note that, for Tang's method [5], we only need to implement the part of initial ship location without removing false alarms. We also evaluate Tang's method on "Sea set" which includes only sea images (28 images and 32 samples) and Liu's method on "Head set" which only contains ships with "V" shape head (247 images and 575 samples). The training work of "BING based on our ship space" is done on our data set. Furthermore, we select top 124 images from our training set to evaluate four labeling ways (Section II-B).

In our experiment, parameters are empirically chosen. In detail, we set $k = 0.5$, $\eta = 0.5$, and $N = 5000$.

A. Analysis for Stage I

Here, we analyze why we need to design our stage II. Fig. 6 shows the score ranking histogram of positive candidates with the same scale in stage I. It can be seen that score rankings generated by BING on VOC2007 [18] are in a cluster and can be easily moved to the front of the queue by linear stretching of the scores using a corresponding linear model in stage II of BING. However, the ones generated in stage I of our method are not in a cluster. It is difficult to improve the rankings of these positive candidates in the same way. Therefore, we need to explore more cues in our stage II.

B. Candidate Quality Evaluation

We evaluate AR and DR-#WIN (detection rate given #WIN candidates) on our testing set. The results are shown in Fig. 7. BING only achieves 20.8% detection rate (DR) and 3.7% AR using 5000 candidates. "BING based on our ship space" gets only 87.2% DR (but much higher than BING) and 43.4% AR. Both Tang's method and Liu's method get very low DR. Tang's

[1] https://sites.google.com/site/hrsc2016/. Now researchers can see samples on this site. The full data set will be released in several months.

[2] We have put the source codes implemented by us on our data set website, which are used to check the correctness of our comparison experiments.

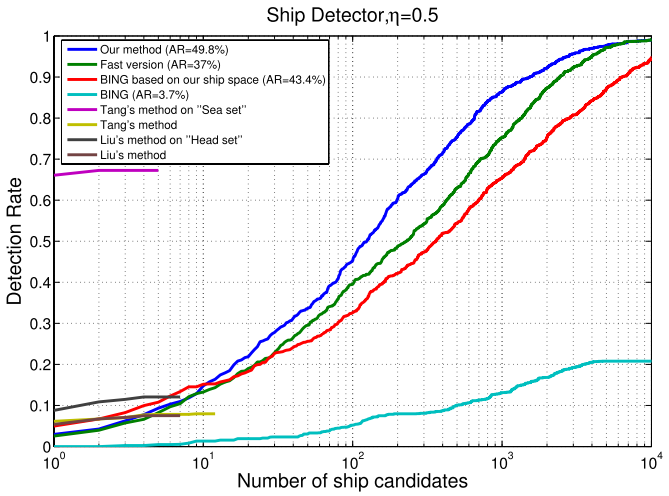


Fig. 7. Comparative experimental results. Here, we calculate AR using 5000 candidates.

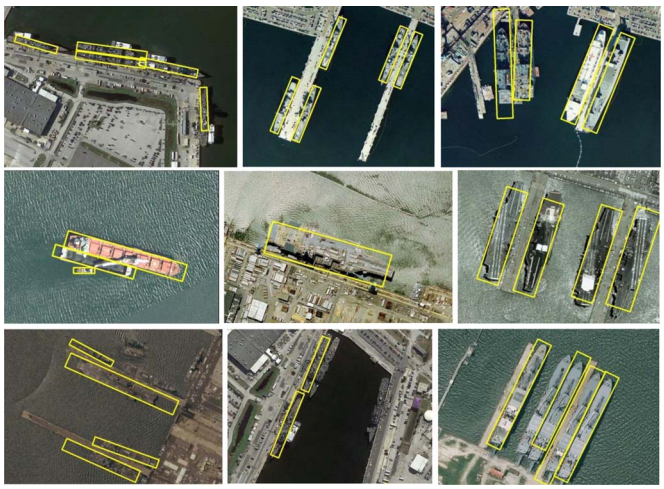


Fig. 8. Candidate samples on HRSC2016: the best fitness in 5000 candidates. Note that the images in the bottom row contain a few false-negative judgments.

method on "Sea set" can achieve 67.3% DR with less than 10 candidates. Our method gives 49.8% AR and 97.6% DR. The fast version can achieve close DR (97.2%) but lower AR (37.0%). It can be seen that our method achieves the best performance.

On the same laptop with an Intel i7-6700K CPU, we evaluate the computation time per image for our method, fast version, "BING based on our ship space", and BING method.

The times for the four methods are 1.26, 0.31, 0.78, and 0.08 s, respectively. The fast version is about four times slower than BING, mainly because our ship rotated bounding box space is much larger (about 5 times for one 1155×820 image) than the window space of BING.

However, the fast version can still meet the demand of near-real-time application.

Fig. 8 shows the ones with best fitness to ground truth in 5000 candidates generated by our method. In the bottom row, there are several ships uncovered by the 5000 candidates. We find that the rankings of the candidates which can cover those uncovered ships are always between 5000 and 20 000.

V. Conclusion

We have presented the possibility of covering ships of any type in complex backgrounds based on a nearly closed-form ship rotated bounding box space. Based on the space, we have designed a two-cascaded linear model to score for latent candidates, followed by binary linear programming to select a small number of highly potential candidates. Experimental results prove the effectiveness of our method.

Here are some possible directions for improvement: designing a new method to construct ship rotated bounding box space instead of k-means, exploring why we get a different distribution of candidate score ranking in stage I of our method, and introducing other additional cues to further reduce the number of candidates while maintaining high DR and AR.

References

[1] 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.

[2] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4511–4523, Aug. 2014.

[3] G. Yang, B. Li, S. Ji, F. Gao, and Q. Xu, "Ship detection from optical satellite images based on sea surface analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 641–645, Mar. 2014.

[4] L. Zhang *et al.*, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Oct. 2015.

[5] J. Tang, C. Deng, G. B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1174–1185, Mar. 2015.

[6] L. Zhang, L. Zhang, D. Tao, and X. Huang, "A multifeature tensor for remote-sensing target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 2, pp. 374–378, Mar. 2011.

[7] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014.

[8] G. Cheng *et al.*, "Object detection in remote sensing imagery using a discriminatively trained mixture model," *ISPRS J. Photogramm. Remote Sens.*, vol. 85, pp. 32–43, Nov. 2013.

[9] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[10] G. Cheng *et al.*, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4238–4249, Aug. 2015.

[11] G. Liu *et al.*, "A new method on inshore ship detection in high-resolution satellite images using shape and context information," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 617–621, Mar. 2014.

[12] J. Lin, X. Yang, S. Xiao, Y. Yu, and C. Jia, *A Line Segment Based Inshore Ship Detection Method*. Berlin, Germany: Springer-Verlag, 2012, pp. 261–269.

[13] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5107–5116.

[14] R. Achanta *et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[15] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 5, pp. 6644–6665, 2015.

[16] M. Cheng, Z. Zhang, W. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3286–3293.

[17] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE ICCVW*, Dec. 2015, pp. 564–586.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.