



Tensorized Generalized Hough Transform for Object Detection in Remote Sensing Images

Hao Chen , *Member, IEEE*, Tong Gao, Guodong Qian, Wen Chen, *Student Member, IEEE*,
and Ye Zhang , *Member, IEEE*

Abstract—To avoid using a large 4D-Hough counting space (HCS) and complex invariant features of generalized Hough transform (GHT) or its extensions when detecting objects in remote sensing image (RSI), a tensorized GHT (TGHT) is proposed to extract object contour by simple gradient angle feature in a 2D-HCS using a single training sample. Considering that tensor can record the structure relationship of object contour, tensor representation R -table is constructed to record the contour information of template. For slice centered at each position of RSI, the tensor-space-based voting mechanism is presented to use the tensor that records the contour information of slice to gather votes at the same entry of 2D-HCS. Furthermore, a multiorder binary-tree-based searching method is presented to accelerate voting by searching the index numbers of elements in tensors. In addition, by solving the tensor-space-based optimization problem that is used to determine the candidates objects, the cause of false alarms (FAs) caused by interferences with complex contour and FAs caused by interferences that are partial-similar to objects is revealed, and the matching rate and matching sparsity-based strategies are then proposed to remove these FAs. Using public RSI datasets with different scenes, experimental results demonstrate that TGHT reduces nearly 99% storage requirement compared with GHT for RSI with size exceeding 1000×1000 under small time consumption, and outperforms the well-known contour extraction methods and state-of-the-art deep-learning-based methods in terms of precision and recall.

Index Terms—Multiorder binary-tree-based searching method, object detection, tensor-space-based contour extraction, tensor-space-based false alarms (FAs) removal, tensorized generalized Hough transform (TGHT).

I. INTRODUCTION

WITH the development of imaging sensor technology, there is a growing interest in various applications for remote sensing information processes, such as object detection [1]–[3], unmixing [4], and hyperspectral image classification [5]. Of these, object detection is considered a fundamental application and challenge task for remote sensing image (RSI) analysis and processing. Recently, owing to the advantage of powerful feature representation, various deep-learning-based

methods have been developed for different object detection tasks. In [6], the weakly supervised learning method [7] and high-level feature learning technology were combined to construct an object detection framework for RSIs. In [8], by utilizing the two-stream pyramid module and an encode–decode module, the LV-Net was built to achieve salient object detection [9] in RSIs. Although these deep-learning-based methods can be used to detect different types of objects, they rely heavily on many training samples. In comparison, contour extraction based methods are similarly effective for object detection, but use much fewer training samples [10]–[12]. Contour extraction methods can be categorized into two groups: analytical shape-oriented methods and nonanalytical shape-oriented methods. Representative analytical shape-oriented contour extraction methods, such as line segments extraction, circle extraction, and rectangle extraction, can be used to detect objects with a corresponding analytical shape. Since airport runways can be described as parallel line segments with a certain length, the geometrical features of line segments are constructed as a saliency map to detect the airport [13]. By applying the circular Hough transform, the contours of above-ground circular storage structures are extracted in RSIs for complex industrial environments [14]. In [15], a rectangle extraction method was used to detect building rooftops.

As more objects (e.g., airplanes and ships) in RSIs have complex shapes (i.e., nonanalytical shapes), it is difficult to detect these objects using analytical shape-oriented methods. Therefore, contour extraction methods for nonanalytical shapes have wider application potential. As a typical representative, the generalized Hough transform (GHT) [16] adopts a reference table (i.e., R -table) to record edge information of a template image by defining a mapping from the orientation of contour points (CPs; calculated by gradient angle feature) to a reference point (RP) in arbitrary shapes. It then detects a specific shape in the test image according to the constructed R -table. Considering typical objects (e.g., airplanes and ships) present with unknown orientations and sizes in RSIs, the existing contour extraction methods consume a large parameter space or use complex invariant features to cover all potential objects.

In the GHT, the positions of votes are calculated under all possible rotation angles and scales when the objects to be detected present different orientations and sizes. Votes are gathered in a large parameter space [i.e., 4D-Hough counting space (4D-HCS)], where the value of each cell in 4D-HCS indicates the number of votes for the object with the corresponding horizontal position, vertical position, rotation angle, and scale. To reduce

Manuscript received March 3, 2020; revised April 25, 2020 and May 27, 2020; accepted June 14, 2020. Date of publication June 17, 2020; date of current version July 2, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61771170 and Grant 61871150. (Corresponding author: Hao Chen.)

Hao Chen, Tong Gao, Wen Chen, and Ye Zhang are with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: hit_hao@hit.edu.cn; 872095644@qq.com; 1152878181@qq.com; zhyc@hit.edu.cn).

Guodong Qian is with the Beijing Institute of Remote Sensing Information, Beijing 100192, China (e-mail: qianguodong@163.com).

Digital Object Identifier 10.1109/JSTARS.2020.3003137

the large storage requirement caused by 4D-HCS, some GHT extensions replace 4D-HCS with 2D-HCS (i.e., eliminate two degrees of freedom in the parameter space) at the expense of using complex rotation-scale invariant features. In invariant GHT (IGHT), features based on pairs of CPs are constructed to extract object shapes in 2D-HCS. One weakness of IGHT is that the pair-CPs-based feature is susceptible to occlusion and interference compared to the gradient angle feature [17]. Lin *et al.* [18] proposed a rotation-invariant feature called radial-gradient angle (RGA) to search for potential objects in 2D-HCS. The RGA of the same CP was calculated multiple times during the process of object detection. More complex invariant features (e.g., Fourier-based descriptors and the local triangle feature) were adopted in local-IGHT [19] and polygon-IGHT [20] to detect objects with different orientations and sizes in 2D-HCS, but using these features had a higher computational load than using the gradient angle feature. In [21], by integrating different channel features, feature learning technology, and an ensemble classifier, an optical remote sensing imagery detector was established to locate objects with different orientations and sizes in RSIs. In [22], a detection framework was constructed by using a rotation-invariant Fourier representation for a histogram of gradient orientation (HoG) feature to detect objects with unknown orientations and sizes. Although Fourier representation for an HoG feature is invariant to object rotation, it requires multiple calculations for Fourier transform and convolution operations.

In practice, since object detection in RSIs is susceptible to interference from other objects with similar characteristics, false alarms (FAs) often occur for RSIs containing complex backgrounds. Researchers have developed different strategies to reduce FAs for contour extraction methods. For example, weighted voting and outline continuity factor-based strategies were designed to reduce FAs caused by shape-similar distractors in RSIs [23]. An iterative training-based IGHT was proposed to reduce FAs by gradually increasing the number of votes for objects and decreasing the number of votes for interferences [24]. To improve the contour extraction results of a ship head, contour refinement strategies and a Gini coefficient-based criterion were presented to remove nonship-head contours with large curvature or large Gini coefficients [25]. However, few strategies that are used to reduce FAs for contour extraction based object detection in RSIs explore the causes of FAs in detail.

The limitations of existing methods motivate us to consider two interesting problems.

- 1) Design a contour extraction method that can extract potential objects with unknown orientations and sizes using a simple gradient angle feature in 2D-HCS.
- 2) Analyze the cause of FAs for contour extraction and construct effective strategies to remove these FAs.

To address these problems, we replace the GHT voting mechanism that traverses CPs to accumulate votes in a large 4D-HCS with the novel approach of using the contour information of slices centered at arbitrary positions in the RSI to calculate the number of votes at the corresponding position. Based on this idea and considering that a tensor can record the structure relationship of an object contour, it makes sense to exploit the tensor to describe the contour information of a slice to thus develop a contour

extraction method and analyze the cause of FAs. Therefore, we propose the tensored GHT (TGHT) to extract object contours with a single sample in 2D-HCS by using a simple gradient angle feature. TGHT is a unified tensor-space-based contour extraction scheme, including three parts, i.e., tensor-space-based contour representation, a tensor-space-based voting mechanism, and tensor-space-based FA removal. Specifically, for a template image containing a certain object, the contour information is described as a tensor representation R -table (TR- R -table). To detect potential objects in an RSI, combined with a constructed TR- R -table, we establish a tensor space-based voting mechanism that traverses positions of the RSI instead of the GHT voting mechanism that traverses CPs in RSI. In addition, to reduce storage requirements and time consumption for tensor operations in TGHT, we propose a multiorder binary tree (BT)-based searching method to accelerate voting by efficiently calculating the inner product between tensors. Furthermore, to reduce FAs caused by various interferences in RSIs, the process of object detection is converted to a tensor-space-based maximization of the inner product to reveal the cause of two types of FAs, and two strategies [i.e., matching rate (MR) and matching sparsity (MS)] are then presented to remove these FAs, respectively. The experiments are conducted on RSIs with different scenes collected from the publicly available NWPU-VHR-10 [1] dataset to examine the effect of TGHT in terms of storage requirement, time consumption, and detection results compared to well-known contour extraction methods.

The contributions of this article are summarized as follows.

- 1) Compared to GHT and existing GHT extensions utilizing a 4D-HCS or complex invariant features to detect objects with unknown orientations and sizes in RSI, TGHT exploits a novel tensor space voting mechanism to detect objects with unknown orientations and sizes by avoiding using either a large Hough counting space (4D-HCS) or complex invariant features of other GHT extensions.
- 2) TGHT provides an analytic expression to calculate the number of votes at each entry of 2D-HCS, which can be utilized to reveal the cause of FAs. Reducing FAs significantly improves object detection results. Although existing contour extraction methods [13]–[25] have some specific strategies to reduce FAs, they do not explore the cause of FAs in detail. In comparison, by virtue of the analytic expression for TGHT, the process of object detection is converted to the process of solving the tensor-space-based optimization problem, whose solution can be utilized to analyze the cause of FAs. This allows us to construct two strategies to remove these FAs.
- 3) TGHT can obtain accuracy votes for potential objects, which ensures the effectiveness of the detection results. A conventional GHT and its extensions utilize the tabular function to accumulate votes, whereas the quantization coordinates and fuzzy voting strategy cause a single CP vote for the same position multiple times. This indicates that the number of votes for GHT and existing GHT extensions are not accurate. By contrast, by using the tensor-space-based voting strategy for TGHT, the same CP will vote for the same position no more than once.

Thus, TGHT can obtain more accurate votes and better detection results.

The remainder of our article is organized as follows. Section II presents a brief review of the conventional GHT and then presents TGHT in detail. Section III proposes a multiorder BT-based searching method to reduce the storage requirements and time consumption of TGHT. By virtue of the analytic expression for TGHT, Section IV reveals the cause of two types of FAs and proposes two respective strategies to remove these FAs. Section V provides the experimental results. Section V-A discusses the impact of parameter setting of TGHT on detection results, Section V-B verifies the generality of two types of interferences, Section V-C compares the storage requirements and time consumption of TGHT and GHT, and Section V-D compares the performance of TGHT with well-known contour extraction methods and state-of-the-art deep learning methods using publicly available datasets. Section VI concludes this article.

II. TENSORED GHT

A brief introduction to the notation used in this article and the conventional GHT is given in the following sections.

A. Notations and Operations

According to the conventional notations in [26], the lowercase letter (e.g., s), the lowercase boldface letter (e.g., \mathbf{v}), the uppercase boldface letter (e.g., \mathbf{M}), and the Euler script letter (e.g., \mathcal{X}) denote scalar, vector, matrices, and tensors, respectively. For the n -order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$, its (i_1, i_2, \dots, i_n) th entry is denoted $\mathcal{X}(i_1, i_2, \dots, i_n)$, where d_i denotes the dimension of the i th mode for the tensor. The definitions of basic tensor operations used in this article are summarized as follows.

Definition 1 (Mode- k product): Let $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$ be an n -order tensor and $\mathbf{M} \in \mathbb{R}^{d_k \times d'_k}$ be a matrix. The mode- k product of \mathcal{X} with \mathbf{M} is denoted as $\mathcal{X} \times_d \mathbf{M}$, whose result is an n -order tensor of dimension $d_1 \times \dots \times d_{k-1} \times d'_k \times d_{k+1} \times \dots \times d_n$ with its $(i_1, \dots, i_{k-1}, i'_k, i_{k+1}, \dots, i_n)$ entry given by

$$\begin{aligned} & (\mathcal{X} \times_d \mathbf{M})_{(i_1, \dots, i_{k-1}, i'_k, i_{k+1}, \dots, i_n)} \\ &= \sum_i \mathcal{X}(i_1, \dots, i_{k-1}, i, i_{k+1}, \dots, i_n) \times \mathbf{M}(i, i'_k). \end{aligned} \quad (1)$$

Definition 2 (Inner product of tensors): Let $\mathcal{X}_1 \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$ and $\mathcal{X}_2 \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$ be n -order tensors. The inner product of \mathcal{X}_1 and \mathcal{X}_2 is given by

$$\langle \mathcal{X}_1, \mathcal{X}_2 \rangle = \sum_{i_1, i_2, \dots, i_n} \mathcal{X}_1(i_1, i_2, \dots, i_n) \times \mathcal{X}_2(i_1, i_2, \dots, i_n). \quad (2)$$

The operation of extracting the subtensor from the given tensor is applied frequently in this article. For convenience, the MATLAB notation is introduced to denote the subtensor of a given tensor. For example, the notation $\mathcal{X}(:, :, :, i_4, i_5)$ denotes the three-order subtensor of $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_5}$ and

is calculated by

$$\begin{aligned} \mathcal{X}(:, :, :, i_4, i_5) &= \mathcal{X} \times_4 \mathbf{v}_1 \times_5 \mathbf{v}_2 \mathbf{v}_1 \in \mathbb{R}^{d_4} \mathbf{v}_2 \in \mathbb{R}^{d_5} \\ \mathbf{v}_1(i) &= \begin{cases} 1, & \text{if } i = i_4 \\ 0, & \text{otherwise} \end{cases} \\ \mathbf{v}_2(j) &= \begin{cases} 1, & \text{if } j = i_5 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (3)$$

B. Brief Introduction to the GHT

GHT was introduced by Ballard [16] and is generalized from the Hough transform to address contour extraction for nonanalytic shapes. As illustrated in Fig. 1, the RP (see the notion (x_r, y_r) in Fig. 1) is selected in the template image, and the gradient angles θ are calculated quantized to i_θ for all the CPs. Then, the R -table extracted from the template is constructed to describe the mapping from CPs to RP by recording all the reference vectors $R(\theta)$ (i.e., the displacement from CP to RP) and the corresponding index numbers of gradient angles (i.e., i_θ) as entries in the offline phase. In the online phase, considering that there are objects with different orientations and sizes in RSIs, 4D-HCS is set up over the domain of the parameters, where each finite cell of the HCS corresponds to a certain range of positions, orientations, and scales of the object in the RSI. To extract the potential objects in the RSI, the gradient angle θ for each CP in the RSI is extracted and quantized to i_θ , and the position of vote $[x_r, y_r]$ is calculated according to the reference vectors $R(\theta)$ recorded in entry i_θ of the R -table under all potential rotation angles and scales. The generated votes are gathered at the corresponding entry $[x_r, y_r, \alpha, s]$ of 4D-HCS, as shown in

$$[x_r, y_r]^T = [x(\theta), y(\theta)]^T + s \cdot \beta(\alpha) \cdot R(\theta) + [\tau, \tau]^T \quad (\tau \leq \delta) \quad (4)$$

where $[x_r, y_r]$, θ , α , s , $[x(\theta), y(\theta)]$, $\beta(\alpha)$, $R(\theta)$ and δ , respectively, denote the position of the vote in the RSI, the gradient angle, the rotation angle (i.e., the difference between the orientation of the object in the template image and that in the RSI), the scale (i.e., the ratio between the size of the object in the template image and that in the RSI), the position of CPs, the rotation matrix, the reference vector, and the length of step for the fuzzy voting strategy. For convenience, α and s hereinafter represent the rotation angle and scale, respectively. After traversing all the CPs in the RSI, the accuracy position, orientation, and size of the potential objects are determined by the index number of maxima in 4D-HCS.

The procedure for implementation of GHT is concluded as follows.

- 1) For a given RSI, extract the contour image by using an edge detection operator.
- 2) Traverse CPs in the RSI and calculate the index number of the gradient angle for the corresponding CP.
- 3) Look up the reference vectors in the entry of the R -table and increment the corresponding entries of 4D-HCS.
- 4) Select entries with values exceeding the threshold in 4D-HCS as the RPs of candidate objects.

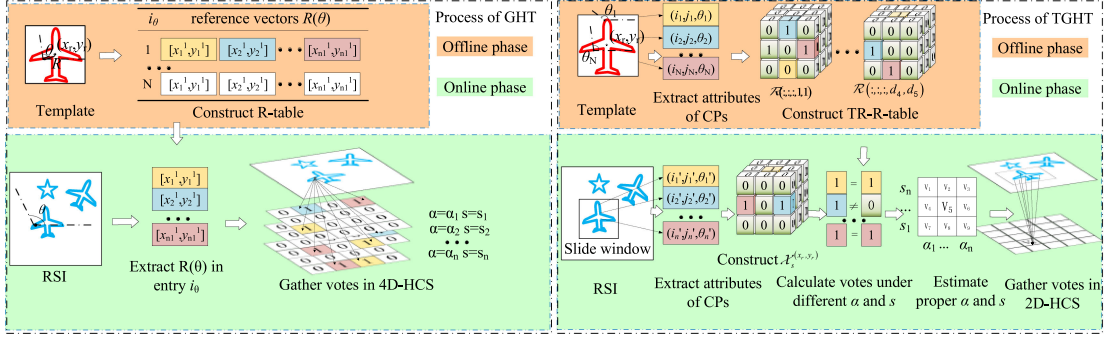


Fig. 1. Illustration of object detection in RSIs using GHT and TGHT.

C. Extend GHT to TGHT

As discussed, 4D-HCS as used in GHT has a large storage requirement. To reduce the dimensionality of 4D-HCS to 2D and avoid using the complex invariant feature of other GHT extensions, we propose the TGHT, as illustrated in Fig. 1. In the offline phase, by utilizing the tensor-space-based contour representation, the object contour in the template image is described and recorded as a TR-R-table. In the online phase, to extract potential objects in RSIs, we apply the tensor-space-based voting mechanism instead of the GHT voting mechanism to gather votes in 2D-HCS with the constructed TR-R-table.

1) *Offline Phase of TGHT*: In the offline phase of TGHT, the contour information of the template image is described and recorded as a TR-R-table for detecting objects with specific shapes in the RSI (denoted as $I_{RSI} \in R^{M' \times N'}$). Before constructing the TR-R-table, the edge image $D_{tem} \in R^{M \times N}$ is extracted by an edge detection operator [27] from the given template image (denoted $I_{tem} \in R^{M \times N}$), where $D_{tem}(i, j) = 1$ denotes that the CP exists in position (i, j) of the template image. According to the process of R-table construction for GHT, the R-table describes the object shape by recording reference vectors and corresponding gradient angles as entries. In other words, the CPs for GHT have three attributes (i.e., horizontal position, vertical position, and gradient angle). Since the potential objects in the RSI may present different poses (i.e., different orientations and sizes) relative to that in the template image, the same CP in objects with different poses will have different positions and gradient angles. An example of object rotation and scaling is shown in Fig. 2. It is observed in Fig. 2 that after undergoing rotation and scaling, the position of the CP changes from (i, j) to (i', j') , and the gradient angle of the CP changes from θ to θ' . This indicates that the attributes of CPs for an object with a single orientation and size cannot be utilized directly to extract an object contour with an unknown orientation and size in an RSI. Therefore, it is necessary to calculate the attributes of CPs in I_{tem} under different rotation angles and scales to cover all potential objects in RSIs.

Proper steps $\Delta\alpha$ and Δs are used to traverse possible ranges of rotation angle α and scale s , respectively, i.e., $\alpha \in [0, 2\pi]$ and $s \in [s_{min}, s_{max}]$, where s_{min} and s_{max} denote the possible minimum and maximum scale, respectively. For current $\alpha = (i_\alpha - 1) \times \Delta\alpha$ and $s = (i_s - 1) \times \Delta s + s_{min}$, where i_α and

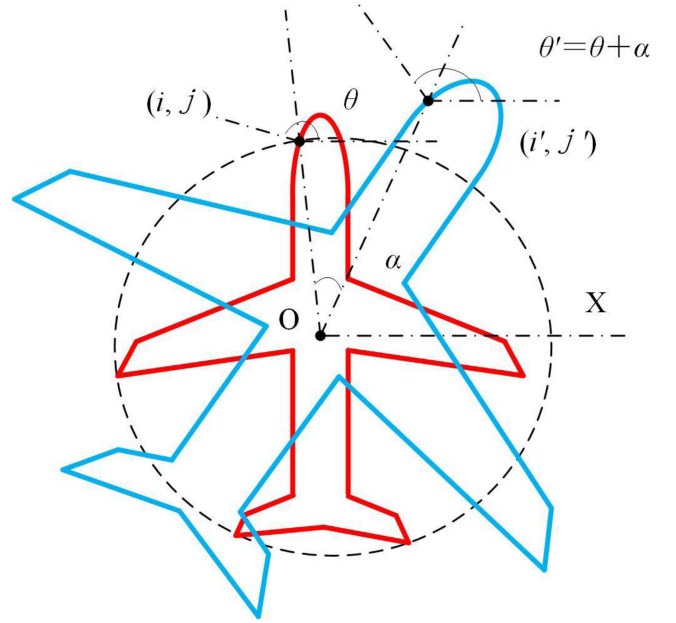


Fig. 2. Illustration of the changes of attributes for object rotating and scaling.

i_s denote the index number of the rotation angle and the index number of the scale, respectively, the new position of the CP after rotation and scaling can be calculated by (5). Since the gradient angle changes in the same ratio as the rotation of the object [17], the new gradient angle of the CP can be calculated by (6) as follows:

$$[i', j']^T = s \cdot \beta(\alpha) [i - x_r, j - y_r]^T + [x_r, y_r]^T \quad (5)$$

$$\theta' = \text{mod}(\theta + \alpha, 2\pi) \quad (6)$$

where $[i', j']$, $[i, j]$, $[x_r, y_r]$, θ' , θ , and $\text{mod}(\cdot)$, respectively, denote the position of the CP after rotation and scaling, the position of the CP, the position of the RP in I_{tem} , the gradient angle of the CP after rotation and scaling, the gradient angle of the CP, and the remainder operator. To reduce the slight difference in the gradient angle for the same CP caused by different imaging conditions, the gradient angle is separated into i_θ levels with the interval of $\Delta\theta$, i.e., $i_\theta = \lceil \theta / \Delta\theta \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling operator. Thus, the CP in the template image has five attributes: horizontal position, vertical position, gradient

Algorithm 1: Process of TR-R-Table Construction.

Input: template image $I_{tem} \in R^{M \times N}$
Initialization: set proper values to $\Delta\alpha, \Delta s, \Delta\theta, s_{min}, s_{max}$, and construct a zero tensor $\mathcal{R} \in R^{M \times N \times d_3 \times d_4 \times d_5}$
Step1: Extract CPs in I_{tem} and calculate the corresponding gradient angle using edge detection operator.
for $i_\alpha = [1 : d_4]$
{for $i_s = [1 : d_5]$
{for each CP in I_{tem}
{Step2: Calculate the position (i, j) and the index number of gradient angle θ for CP under rotation angle $\alpha = (i_\alpha - 1) \times \Delta\alpha$ and scale $s = (i_s - 1) \times \Delta s + s_{min}$ by using (5) and (6), respectively.
Step3: Recording the CP indexed by attributes $(i, j, i_\theta, i_\alpha, i_s)$ to TR-R-table, i.e., $\mathcal{R}(i, j, i_\theta, i_\alpha, i_s) = 1$.}}
Output: \mathcal{R}

angle, rotation angle, and scale. Considering that a tensor can record the structure relationship of multiattribute data, it is suitable to describe the object shape by recording the CPs with different indices of attributes. Therefore, the R -table in GHT is extended to the TR-R-table by constructing a five-order tensor (i.e., $\mathcal{R} \in R^{M \times N \times d_3 \times d_4 \times d_5}$) to record the contour information of the object, where the first, second, third, fourth, and fifth orders, respectively, represent the horizontal spatial domain, the vertical spatial domain, the gradient angle domain, the rotation angle domain, and the scale domain. According to the obtained (i', j', i_θ) under indices i_α and i_s , if the CP indexed by different attributes $(i, j, i_\theta, i_\alpha, i_s)$ exists, the corresponding element in the TR-R-table is set to 1, i.e., $\mathcal{R}(i, j, i_\theta, i_\alpha, i_s) = 1$; otherwise, it is set to 0. After the attributes of each CP for all possible α and s are calculated and recorded in \mathcal{R} , the construction of the TR-R-table is complete. The detailed process of TR-R-table construction is concluded as shown in Algorithm 1.

Analogous to the fuzzy voting strategy in GHT, to prevent the slight deformation of an object impacting contour extraction caused by occlusion and imaging condition in RSIs, we adopt the fuzzy strategy to tolerate the slight deformation of the object contour, i.e., the position of CPs in the template image are allowed to vary within a small neighborhood. To record the fuzzy positions of CPs in the TR-R-table, we use two band matrices, $F_1 \in R^{M \times M}$ and $F_2 \in R^{N \times N}$, to generate the fuzzy TR-R-table, as per

$$R_F = \text{binary} \left(R \prod_{k=1}^2 \times_k F_k \right)$$

$$F_k(i, j) = \begin{cases} 1, & \text{if } |i - j| \leq \delta \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where \times_k , $\text{binary}(\cdot)$, and \mathcal{R}_F , respectively, denote the mode- k product, the binarization operator that converts the input tensor

to a binary tensor, and the fuzzy TR-R-table. The detailed operation of $\text{binary}(\cdot)$ is given in

$$\text{binary}(R_F)(i, j, i_\theta, i_\alpha, i_s) = \begin{cases} 1 & \text{if } R_F(i, j, i_\theta, i_\alpha, i_s) \geq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

For convenience, the fuzzy TR-R-table is hereinafter denoted as TR-R-table (i.e., \mathcal{R}).

2) *Online Phase of TGHT:* In the online phase, the potential objects are detected for a given RSI (denoted as $I_{RSI} \in R^{M' \times N'}$) by using the constructed TR-R-table in the offline phase. First, the CPs in I_{RSI} are extracted by an edge detection operator, and the attributes of CPs are calculated, including the positions and gradient angles. Subsequently, it is essential to construct an effective voting mechanism to locate potential objects. Since the position, rotation angle, and scale for potential objects are unknown, the GHT voting mechanism requires the construction of a 4D-HCS to store generated votes to locate potential objects. To reduce the dimensionality of HCS from 4-D to 2-D, the slice centered at each position in the RSI is operated to calculate the number of votes at the corresponding position and store these results in the same entry of 2D-HCS.

In detail, a window with the same size as the template image is adopted to scan each position (x_r, y_r) in the RSI to obtain a series of slices. For each slice (denoted I_s), the gradient angle of the CPs is separated into i_θ levels with the interval of $\Delta\theta$ just as in the process of the offline phase. To describe these CPs with different attributes (i.e., the position and gradient angle) in the slice, the attributes of CPs in the slice are recorded into the tensor as part of the TR-R-table construction. Note that the orientation and size of the object in the template image are known, whereas those of the potential object in the slice are unknown. Therefore, by reducing the rotation angle domain and scale domain of the TR-R-table, only a three-order tensor, denoted $\mathcal{X}_s^{(x_r, y_r)} \in R^{M \times N \times d_3}$, is required to describe the contour information of the slice. Here, the first, second, and third orders represent the horizontal spatial domain, vertical spatial domain, and gradient angle domain, respectively. Similar to the TR-R-table construction, if the CP indexed by attributes (i.e., the position (i, j) and i_θ) exists in the slice, the corresponding element of $\mathcal{X}_s^{(x_r, y_r)}$ is set to 1, i.e., $\mathcal{X}_s^{(x_r, y_r)}(i, j, i_\theta) = 1$; otherwise, it is set to 0.

Next, the number of votes at the candidate RP (x_r, y_r) of the RSI is calculated according to the constructed TR-R-table and $\mathcal{X}_s^{(x_r, y_r)}$. Note that if both $\mathcal{R}(i, j, i_\theta, i_\alpha, i_s)$ and $\mathcal{X}_s^{(x_r, y_r)}(i, j, i_\theta)$ are equal to 1, i.e., there is a CP with the same attributes (i, j, i_θ) in both the template image and the slice, the CP in the slice will vote for the candidate RP (x_r, y_r) under the specific index number i_α of rotation angle and index number i_s of scale. Therefore, $\langle \mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(:, :, :, i_\alpha, i_s) \rangle$ can be used to calculate the number of votes at a candidate RP (x_r, y_r) under a corresponding index number of rotation angle (i.e., i_α) and index number of scale (i.e., i_s), where $\langle \cdot \rangle$ denotes the inner product operation. For specific i_α and i_s , larger $\langle \mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(:, :, :, i_\alpha, i_s) \rangle$ means more votes are gathered at the candidate RP under i_α and i_s . Thus, the accurate rotation angle and scale of the potential

object are determined by the maximum votes index number i_α and i_s for $\langle \mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(:, :, :, i_\alpha, i_s) \rangle \forall i_\alpha, i_s$. Additionally, since there is an approximately linear relationship between the number of CPs and the scale of the object, to prevent the scale from impacting the number of votes, the scale ($s_{\min} + i_s \times \Delta s$) corresponding to i_s for the potential object is used to normalize the number of votes. The final number of votes for a candidate RP (x_r, y_r) is calculated by (9) and recorded in the corresponding entry of 2D-HCS

$$HCS(x_r, y_r) = \max_{i_\alpha, i_s} \frac{\langle \mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(:, :, :, i_\alpha, i_s) \rangle}{(s_{\min} + i_s \times \Delta s)}. \quad (9)$$

After calculating the number of votes at each position of the RSI, the positions of candidate objects can be extracted by searching the maxima in entries of 2D-HCS. Note that the elements in $\mathcal{X}_s^{(x_r, y_r)}$ and $\mathcal{R}(:, :, :, i_\alpha, i_s)$ are binary, so a single CP in the RSI generates no more than one vote at the specific position of the RSI for TGHT. By contrast, in the GHT voting mechanism, which incorporates a fuzzy voting strategy and coordinate quantification, by using different reference vectors $R(\theta)$ recorded in the R -table, a single CP $[x(\theta), y(\theta)]$ may increment the same entry of 4D-HCS multiple times [see (4)], i.e., a single CP in the RSI may generate more than one vote at the specific position of the RSI for GHT. Thus, TGHT appears to output more accurate votes than GHT.

III. EFFECTIVE IMPLEMENTATION OF TGHT

To reduce the time consumption and storage requirement for tensor operations in TGHT, we propose an effective implementation of TGHT based on an index number searching strategy.

TGHT utilizes the inner product of two tensors (i.e., $\mathcal{R}, \mathcal{X}_s^{(x_r, y_r)}$) to calculate the number of votes at position (x_r, y_r) of the RSI [see (9)], and store them into the corresponding entry (x_r, y_r) of 2D-HCS. In this way, it will consume the store space of the entire tensor $\mathcal{R} \in R^{M \times N \times d_3 \times d_4 \times d_5}$ and $\mathcal{X}_s^{(x_r, y_r)} \in R^{M \times N \times d_3}$, and some dynamically allocated storage space caused by accumulating votes. Therefore, the spatial complexity of using tensor operations directly is $O(M \times N \times d_3 \times d_4 \times d_5)$. Note that the number of multiplications for calculating the number of votes at entry (x_r, y_r) in (9) is equal to $M \times N \times d_3 \times d_4 \times d_5 + d_4 \times d_5$. Therefore, the time complexity of using tensor operations directly is $O(M \times N \times d_3 \times d_4 \times d_5)$. According to the construction process of $\mathcal{X}_s^{(x_r, y_r)}$ and \mathcal{R} , if CPs with specific attributes exist in the template image or slice of the RSI, the elements in the corresponding entries of $\mathcal{X}_s^{(x_r, y_r)}$ or \mathcal{R} are equal to 1; otherwise, they are equal to 0. This means that $\mathcal{X}_s^{(x_r, y_r)}$ and \mathcal{R} are binary. Moreover, since the number of CPs in the template image or slice of the RSI is much smaller than the number of elements in $\mathcal{X}_s^{(x_r, y_r)}$ or \mathcal{R} , $\mathcal{X}_s^{(x_r, y_r)}$ and \mathcal{R} are sparse (i.e., only a small number of elements are 1). From the definition of the inner product between tensors, the result of the inner product is not related to the 0 elements. Therefore, it only needs to store the

$p \times M \times N \times d_3 \times d_4 \times d_5$ index numbers of elements corresponding to a 1 value in \mathcal{R} instead of storing entire tensor, where p denotes the proportion of the elements corresponding to a 1 value in all the elements for \mathcal{R} . Thus, the storage load caused by $\mathcal{X}_s^{(x_r, y_r)}$ and \mathcal{R} is significantly reduced. Based on these index numbers, an index number searching method is constructed to calculate the number of votes at the entry of 2D-HCS, according to the following steps.

For each index number that corresponds to an element with a 1 value in $\mathcal{X}_s^{(x_r, y_r)}$, search the same index number among all the index numbers that correspond to elements with a 1 value in $\mathcal{R}(:, :, :, i_\alpha, i_s)$. If the same index number exists, the number of votes under i_α and i_s will be accumulated by 1. The final number of votes at each entry of 2D-HCS is determined by the maxima of the number of votes under different i_α and i_s .

Note that the time consumption of the index number searching method is related to the searching strategy. As an example, for exhaustive searching (i.e., compare the index number (i, j, i_θ) of elements equal to 1 in $\mathcal{X}_s^{(x_r, y_r)}$ with the index number of all elements equal to 1 in \mathcal{R} until the same index number is found), the largest number of comparison operations for searching the index number (i, j, i_θ) is equal to $p \times M \times N \times d_3 \times d_4 \times d_5$. To reduce the time consumption of voting for TGHT, an efficient searching strategy is required.

Inspired by the search tree [31] method that is widely used in data searching, we propose a multiorder BT-based searching method to accelerate obtaining the number of votes at each entry of 2D-HCS by reducing comparison operations for index number searching. Compared with the typical search tree methods, including the binary search tree, 2-3-4 tree, and k-d tree, the multiorder BT is constructed for specific applications (e.g., to accelerate voting) and is more suitable for matching indices of multiorder tensor data. Fig. 3 illustrates the procedure of the multiorder BT-based searching method.

As can be seen in Fig. 3, the multiorder BT-based searching method traverses each index number (i, j, i_θ) that corresponds to an element with a 1 value in $\mathcal{X}_s^{(x_r, y_r)}$ and accumulates the generated votes under different i_α and i_s in accumulator \mathbf{P} , where the resulting $\mathbf{P}(i_\alpha, i_s)$ denotes the number of votes under rotation angle i_α and scale i_s . The final number of votes in $HCS(x_r, y_r)$ is determined by the maxima in \mathbf{P} . From Fig. 3, the multiorder BT-based searching method consists of the four stages detailed ahead.

To conveniently describe the detailed process for the multiorder BT-based searching method, we define the sets $S_{\mathcal{X}}^{(x_r, y_r)} = \{(i, j, i_\theta) | \mathcal{X}_s^{(x_r, y_r)}(i, j, i_\theta) = 1, \forall i, j, i_\theta\}$ and $S_{\mathcal{R}} = \{(i', j', i'_\theta, i'_\alpha, i'_s) | \mathcal{R}(i', j', i'_\theta, i'_\alpha, i'_s) = 1, \forall i', j', i'_\theta, i'_\alpha, i'_s\}$, where $(i, j, i_\theta) \in S_{\mathcal{X}}^{(x_r, y_r)}$ and $(i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}$, respectively, denote the index numbers that correspond to elements with a 1 value in $\mathcal{X}_s^{(x_r, y_r)}$ and the index numbers that correspond to elements with a 1 value in \mathcal{R} .

In the first stage of the searching method, the index number $(i, j, i_\theta) \in S_{\mathcal{X}}^{(x_r, y_r)}$ along the first order (e.g., i) is searched among all the index numbers $(i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}$ along the

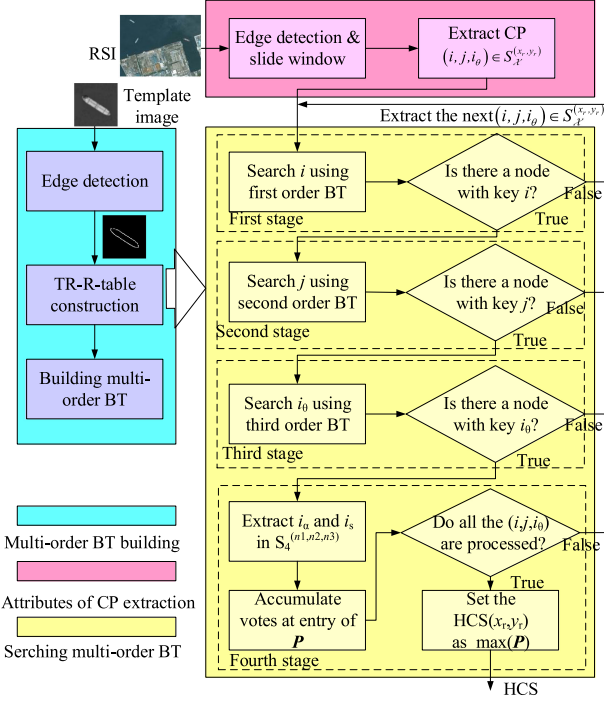


Fig. 3. Procedure of using multiorder BT-based searching method to calculate the number of votes in 2D-HCS.

first order (i.e., horizontal spatial domain). To build the first-order BT, all the index numbers $(i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}$ along the first order are extracted using

$$\mathbf{v}_1 = \mathcal{R} \prod_{k=2}^5 \times_k \mathbf{1}_{d_k}$$

$$S_1 = \{s | \mathbf{v}(s) > 0, 1 \leq s \leq M\} \quad (10)$$

where $\mathbf{1}_{d_k} = \underbrace{[1, \dots, 1]}_{d_k}^T$, and S_1 denotes a set containing the index numbers of elements equal to 1 in \mathcal{R} along the first order. Subsequently, the first-order BT is constructed according to the following steps to record these index numbers in set S_1 . For the index numbers of elements equal to 1 in \mathcal{R} along the first order recorded in set S_1 , choose the median index number of all the index numbers as the key of the root node, and partition the other index numbers into two approximately equal-sized sets, where the first set contains all the index numbers whose values are less than this median, and the second set contains the remaining index numbers. The medians of the first subpartition set and the second subpartition set are regarded as the keys of the root node for the left and right subtrees, respectively. The tree is built recursively until all the index numbers recorded in S_1 are assigned to the keys of nodes. Fig. 4 presents an example of constructing a first-order BT.

Note that all the first-order index numbers for elements equal to 1 in \mathcal{R} , i.e., $\{i' | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i', j', i'_\theta, i'_\alpha, i'_s\}$, are recorded into corresponding nodes of the first-order BT. For the first index number i to be searched, the first-order BT can be used to search the same index number from the keys of nodes

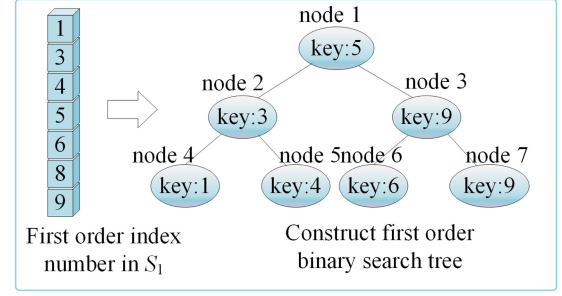


Fig. 4. Constructing first-order BT.

using the following steps. Compare i with the key of the root node for the first-order BT. If i is smaller, the left subtree is searched. If i is larger, the right subtree is searched. If i is equal to the key of root node, the search is successful. The process is repeated until i is found or the remaining subtree is *null*. If the search is successful, i.e., there is a same first-order index number i among all $\{(i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i', j', i'_\theta, i'_\alpha, i'_s\}$, the second index number (e.g., j) is searched later. If there is no node with a key with i , the search is stopped because the corresponding elements in $\mathcal{X}_s^{(x_r, y_r)}$ will not generate votes.

Similar to the first stage, the second-order index number j is searched among all the $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, \forall j', i'_\theta, i'_\alpha, i'_s\}$ along the second order (i.e., vertical spatial domain). To build the second-order BT, the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, \forall j', i'_\theta, i'_\alpha, i'_s\}$ along the second order (i.e., vertical spatial domain) are extracted using

$$\mathbf{M}_2 = \mathcal{R} \prod_{k=3}^5 \times_k \mathbf{1}_{d_k}$$

$$S_2^{(i)} = \{s | \mathbf{M}_2(i, s) > 0, 1 \leq s \leq N\} \quad (11)$$

where $S_2^{(i)}$ denotes a set containing the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, \forall j', i'_\theta, i'_\alpha, i'_s\}$ along the second order. By using the extracted index number recorded in $S_2^{(i)}$, the corresponding second-order BT is built in the same way as the first-order BT.

The constructed second-order BT is searched with the same approach as the first-order BT, but for j . If the search is successful, the third-order index number i_θ is searched later. If there is no node containing a key with j in the second-order BT, the search is stopped.

Similar to the first and second stages, the third-order index number i_θ is searched among all the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, j' = j, \forall i'_\theta, i'_\alpha, i'_s\}$ along the third order (i.e., gradient angle domain). To build the third-order BT, the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, j' = j, \forall i'_\theta, i'_\alpha, i'_s\}$ along the third order are extracted using

$$\mathcal{T}_3 = \mathcal{R} \prod_{k=4}^5 \times_k \mathbf{1}_{d_k}$$

$$S_3^{(i, j)} = \{s | \mathcal{T}_3(i, j, s) > 0, 1 \leq s \leq d_3\} \quad (12)$$

where $S_3^{(i,j)}$ denotes a set containing the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, j' = j, \forall i'_\theta, i'_\alpha, i'_s\}$ along the third order. Similar to the first and second-order BTs, the third-order BT is built to record the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, j' = j, \forall i'_\theta, i'_\alpha, i'_s\}$ along the third order.

By using the constructed third-order BT, the third-order index number i_θ is searched in the same way again. If the search is successful, the votes are generated and accumulated in the fourth stage. If there is no node containing a key with i_θ in the third-order BT, the search is stopped.

In the fourth stage, the index numbers $\{(i', j', i'_\theta, i'_\alpha, i'_s) | (i', j', i'_\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, i' = i, j' = j, i'_\theta = i_\theta, \forall i'_\alpha, i'_s\}$ along the fourth and fifth orders are extracted and recoded into set $S_4^{(i,j,i_\theta)}$ to generate votes under different rotation angles and scales by using

$$\begin{aligned} S_4^{(i,j,i_\theta)} &= \{(i'_\alpha, i'_s) | \mathcal{R}(i, j, i_\theta, i'_\alpha, i'_s) \\ &= 1, 1 \leq i'_\alpha \leq d_4, 1 \leq i'_s \leq d_5\} \end{aligned} \quad (13)$$

where $S_4^{(i,j,i_\theta)}$ denotes a set containing the index numbers of rotation angles and of scales for votes. If the search is successful for the first three stages, the votes under different rotation angles and scales are generated according to the (i'_α, i'_s) recorded in $S_4^{(i,j,i_\theta)}$, and the corresponding entries (i'_α, i'_s) of $\mathbf{P} \in R^{d_4 \times d_5}$ are accumulated. After all the index numbers $(i, j, i_\theta) \in S_{\mathcal{X}}^{(x_r, y_r)}$ are processed using the multiorder BT-based searching method, the number of votes at $HCS(x_r, y_r)$ is determined by the maxima of \mathbf{P} entries. The multiorder BT is thus complete.

Since the largest number of comparison operations between the index number i and the keys in the BT is equal to the height of the corresponding BT, the largest number of comparison operations for searching the index number $(i, j, i_\theta) \in S_{\mathcal{X}}^{(x_r, y_r)}$ is approximately $\log(M \times N \times d_3)$. Compared with the largest number of comparison operations (i.e., $p \times M \times N \times d_3 \times d_4 \times d_5$) for exhaustive searching, the multiorder BT-based searching method significantly reduces the comparison operations to accelerate voting. Therefore, the time complexity of using multiorder BT is $O(N_{cp} \times \log(M \times N \times d_3))$, where N_{cp} denotes the number of CPs in slice centered at (x_r, y_r) . By comparing the time complexity between using tensor operations directly (i.e., $O(M \times N \times d_3 \times d_4 \times d_5)$) and using multiorder BT (i.e., $O(N_{cp} \times \log(M \times N \times d_3))$), it can be found that the multiorder BT will obtain smaller time complexity when the slice contains few CPs or the \mathcal{R} presents a large size.

The process for calculating the number of votes by using the multiorder BT-based searching method is concluded as shown in Algorithm 2.

By using the proposed multiorder BT-based searching method, the procedure to implement TGHT is briefly described as follows.

- 1) For a given RSI, extract the contour image using an edge detection operator.
- 2) Traverse positions (x_r, y_r) in the RSI, and extract the slice centered at the corresponding position.

Algorithm 2: Process of Calculating the Number of Votes by Using Multiorder BT-Based Searching Method.

Input: RSI $I_{RSI} \in R^{M' \times N'}$, multiorder BT

for: $x_r = 1: M'$

for: $y_r = 1: N'$

Initialization: accumulator $\mathbf{P} \in R^{d_4 \times d_5}$

Step 1: Construct $\mathcal{X}_s^{(x_r, y_r)}$ according to the slice centered at (x_r, y_r) in RSI

Step 2: for: each $(i, j, i_\theta) \in S_{\mathcal{X}}^{(x_r, y_r)}$.

Step 3: Search first-order index number i using first-order BT. If the search is successful, perform step.4, otherwise, perform step.2

Step 4: Search second-order index number j using second-order BT. If the search is successful, perform step.5, otherwise, perform step.2

Step 5: Search third-order index number i_θ using third-order BT. If the search is successful, perform step.6, otherwise, perform step.2

Step 6: according to all the index numbers of rotation angle and scale recorded in $S_4^{(i,j,i_\theta)}$, accumulate votes to \mathbf{P}

end

Step 7: Let $HCS(x_r, y_r) = \max_{i_\alpha, i_s} \mathbf{P}(i_\alpha, i_s)$

end

end

Output: 2D-HCS

- 3) For CPs with attributes in the slice, apply the multiorder BT-based searching method to obtain the number of votes at entry (x_r, y_r) of 2D-HCS.
- 4) Select entries with values exceeding the threshold in 2D-HCS as the RPs of candidate objects.

IV. FURTHER IMPROVEMENT TO TGHT FOR FA REMOVAL

Since the contour extraction results are susceptible to other objects with similar characteristics, FAs occur frequently in RSI object detection. Before constructing an effective strategy to reduce FAs, the cause of FAs for contour extraction must be understood. Unlike other contour extraction methods that utilize the discrete tabular function to gather votes, TGHT provides an analytic expression [see (9)] to calculate the number of votes at an arbitrary entry of 2D-HCS, i.e., it utilizes $\frac{(\mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(\cdot, \cdot, \cdot, i_\alpha^*, i_s^*))}{(s_{\min} + i_s^* \times \Delta s)}$ to calculate the number of votes at a specific entry of 2D-HCS, where i_α^* and i_s^* , respectively, denote the optimal index number of the rotation angle and the scale for the potential object. Note that the contour extraction method searches the entries with maximum votes in HCS as the RPs of candidate objects. That is, searching the maxima in entries of 2D-HCS as RPs of candidate objects can be converted to a tensor-space-based optimization problem, as shown in

$$\begin{aligned} \arg \max_{(x_r, y_r)} & \left\langle \mathcal{X}_s^{(x_r, y_r)}, \mathcal{R}(\cdot, \cdot, \cdot, i_\alpha^*, i_s^*) \right\rangle / (s_{\min} + i_s^* \times \Delta s) \\ \text{s.t. } & \mathcal{X}_s^{(x_r, y_r)}(i, j, k) \in \{0, 1\} \forall i, j, k. \end{aligned} \quad (14)$$

For a specific RSI, the optimal (x_r^*, y_r^*) denotes the position of candidate objects. To obtain $\mathcal{X}_s^{(x_r^*, y_r^*)}$ of candidate objects to analyze FA causes, the optimal $\mathcal{X}_s^{(x_r, y_r)}$ that can maximize (14) must be solved. Although typical optimization algorithms (e.g., the simplex algorithm [28]) can be used to solve (14), it is complex and difficult to obtain all the solutions when the solutions are not unique. To solve the optimization problem effectively, we observe from (14) that the objective function belongs to the simple inner product operation, and the feasible region of each element in $\mathcal{X}_s^{(x_r, y_r)}$ is $\{0, 1\}$. Therefore, the optimal solution of $\mathcal{X}_s^{(x_r, y_r)}$ can be conveniently formulated using the following logical analysis.

According to (14), since $(s_{\min} + i_s^* \times \Delta s) > 0$ and \mathcal{R} are binary, the value for each element of $\mathcal{R}(:, :, :, i_\alpha^*, i_s^*) / (s_{\min} + i_s^* \times \Delta s)$ is nonnegative. If the elements in $\mathcal{R}(:, :, :, i_\alpha^*, i_s^*) / (s_{\min} + i_s^* \times \Delta s)$ are positive, the corresponding elements in optimal $\mathcal{X}_s^{(x_r, y_r)}$ are set to 1 to maximize the objective function [i.e., (14)]. If the elements in $\mathcal{R}(:, :, :, i_\alpha^*, i_s^*) / (s_{\min} + i_s^* \times \Delta s)$ are equal to 0, the value of the objective function is not affected by the value of the corresponding elements in optimal $\mathcal{X}_s^{(x_r, y_r)}$. Therefore, the optimal $\mathcal{X}_s^{(x_r, y_r)}$ can be written as

$$\mathcal{X}_s^{(x_r, y_r)}(i, j, i_\theta) = \begin{cases} 1, & \text{if } \mathcal{R}(i, j, i_\theta, i_\alpha^*, i_s^*) = 1 \\ t & \forall t \in \{0, 1\}, \text{ otherwise} \end{cases} \quad (15)$$

where $\mathcal{X}_s^{(x_r, y_r)}$ is the optimal $\mathcal{X}_s^{(x_r, y_r)}$. According to (15), $\mathcal{X}_s^{(x_r, y_r)}$ only depends on the value of elements in entry $\{(i, j, i_\theta) | \mathcal{R}(i, j, i_\theta, i_\alpha^*, i_s^*) = 1 \forall (i, j, i_\theta)\}$. This indicates that the decision criterion that searches the entries with maximum votes as positions of candidate objects only needs to focus on CPs in the slice with attributes (i, j, i_θ) , where the template image containing CPs with the same attributes (i, j, i_θ) (i.e., $\mathcal{R}(i, j, i_\theta, i_\alpha^*, i_s^*) = 1$), and it can ignore CPs in slice with attributes (i, j, i_θ) such that $\mathcal{R}(i, j, i_\theta, i_\alpha^*, i_s^*) = 0$. In other words, the decision criterion focuses solely on the number of CPs in the slice that match the template image, and it ignores the CPs in the slice that do not match the template image and the structure relationship of matched CPs. In the following, according to the weakness of the decision criterion, two FA removal strategies are proposed to reduce FAs caused by two representative types of interference respectively.

A. Removal of FAs Caused by Interference With Complex Contour

For a noninteresting object with a large number of CPs, there are more elements equal to 1 in the corresponding $\mathcal{X}_s^{(x_r, y_r)}$. Thus, more elements are coincidentally matched by the elements equal to 1 in $\mathcal{R}(:, :, :, i_\alpha^*, i_s^*)$ to generate enough votes to be mistaken as objects using the contour extraction method, causing FAs. In our article, this type of interference is defined as noninteresting objects with more CPs than objects to be detected and is called interference with complex contour (ICC). To further illustrate this concept, some representative examples obtained using the template image of an airplane [see Fig. 6(a)] and the RSI collected from the NWPU-VHR 10 dataset are displayed in

Fig. 5, which shows the objects to be detected in the first two rows and ICCs in the middle two rows. It can be seen that ICCs have more CPs than objects, some of which are coincidentally matched by template images, so it is easy for them to be mistaken as objects. Therefore, the decision criterion that focuses on the number of votes cannot be used to distinguish between objects and ICCs. Notably, although both an object and an ICC present various CPs matched by the template image, the ICC has more unmatched CPs than an object, i.e., only a small proportion of CPs for the ICC are matched by a template image. Therefore, we define the MR score to distinguish between objects and FAs caused by ICCs in

$$V_{MR}(x_r, y_r) = HCS(x_r, y_r) / \text{sum}(\mathcal{X}_s^{(x_r, y_r)}) \quad (16)$$

where $V_{MR}(\cdot)$ and $\text{sum}(\cdot)$ denote the MR score of a candidate object and the summation operator, respectively. Note that there are enough matched CPs (i.e., enough votes) for both ICCs and objects, whereas the $\text{sum}(\mathcal{X}_s^{(x_r, y_r)})$ that corresponds to ICCs is much larger than the $\text{sum}(\mathcal{X}_s^{(x_r, y_r)})$ that corresponds to objects. Therefore, the $V_{MR}(\cdot)$ of ICCs is much smaller than that of objects. For the examples in Fig. 5, the MR scores for the objects in the first and second rows are 0.118 and 0.109, respectively, whereas the MR scores for the ICCs in the third and fourth rows are only 0.058 and 0.058, respectively. Therefore, candidate RPs with a small MR are considered to be FAs and are removed.

B. Removal of FAs Caused by Interferences that are Partially Similar to Objects

For a noninteresting object that is partially similar to an object, parts of CPs are easily matched to the corresponding CPs in the template image because the CPs in this type of noninteresting object and an object have similar attributes. For convenience, this type of interference is defined as interferences that are partially similar to objects (IPSSs). Representative examples of IPSSs can be seen in last two rows of Fig. 5, where the fifth and sixth rows show the boarding bridge and stripe on the ground, respectively. Since part of the contour for this type of interference (see Fig. 5) is similar to the partial contour of the object, these interferences generate enough matched CPs to be mistaken as objects.

Unlike ICCs, FAs caused by IPS are difficult to remove with the MR criterion because the number of CPs for IPS is not large enough (see the last rows of Fig. 5). Therefore, it is necessary to develop another strategy to remove FAs caused by IPS.

To distinguish between objects and IPSSs, we need to construct an effective strategy to quantitatively describe the characteristic that the matched CPs for IPS are concentrated on the partial contour of an object. To this aim, we first count the histograms for the objects and IPSSs (see the last column of Fig. 5), where each bin denotes the proportion between the number of matched CPs with corresponding i_θ (i.e., the index number of the gradient angle) and the $\text{sum}(\mathcal{R}(:, :, i_\theta, i_\alpha^*, i_s^*))$. Compared with the histogram obtained by an object (see Fig. 5), the matched CPs in an IPS concentrate on parts of objects, so only a few bins in the histogram have large values and the others have very small values. In other words, the histogram for an IPS is sparser

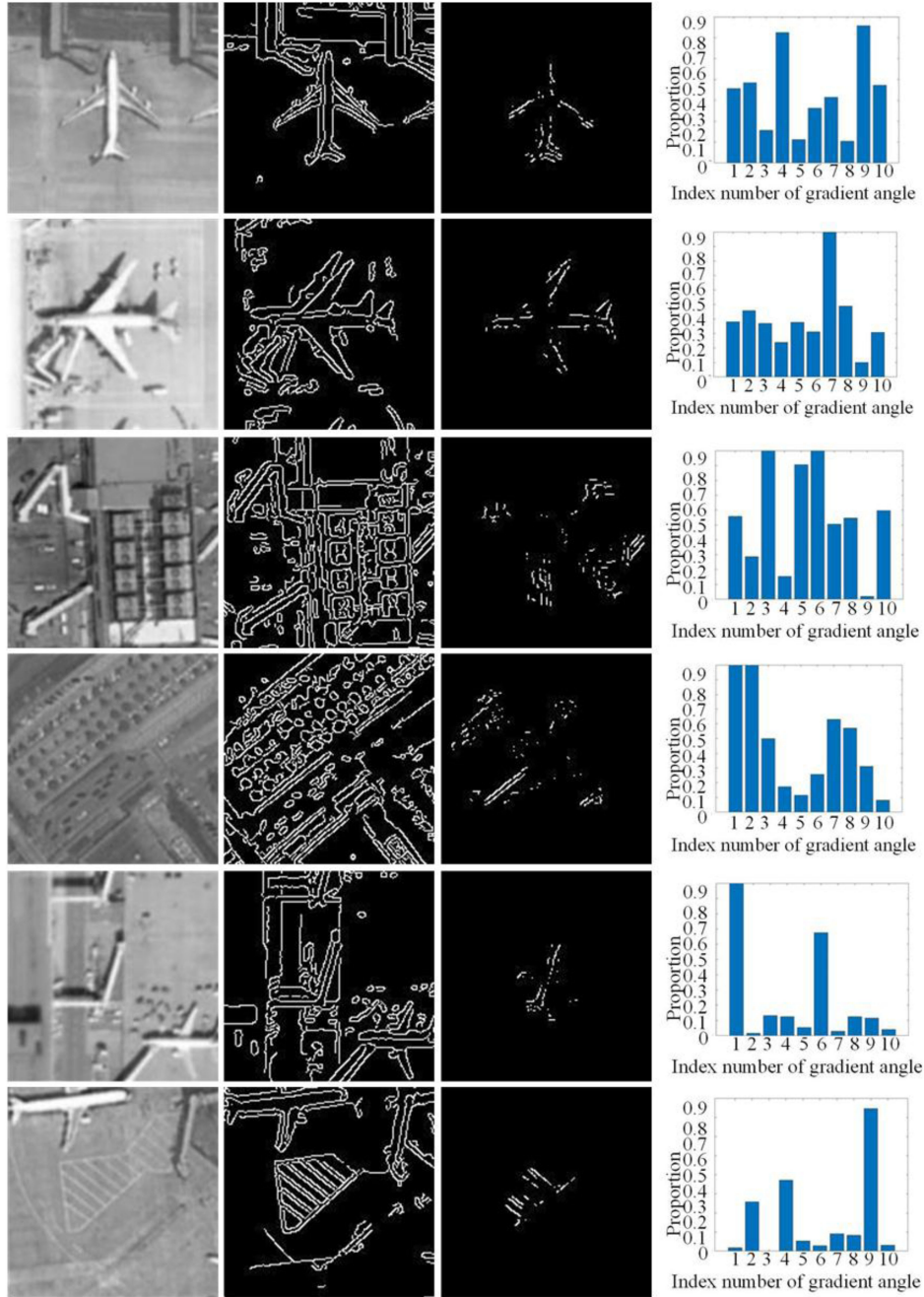


Fig. 5. Representative examples of objects and false alarms. The first two rows are examples of the objects, the middle two rows are examples of interference with complex contour (ICC), and the last two rows are examples of interference with partial-similarity (IPS). The columns from left to right show an input slice, the edge detection results of the slice, matched CPs in the slice, and the histogram of matched CPs.

than the histogram for an object. To capture this characteristic, we introduce a sparsity measure [29] to calculate the matching sparsity score (MS) as follows:

$$\begin{aligned}
 V_{MS}(x_r, y_r) &= \frac{\|\mathbf{y}\|_1 - \sqrt{d_3}\|\mathbf{y}\|_2}{\|\mathbf{y}\|_2 - \sqrt{d_3}\|\mathbf{y}\|_1} \mathbf{y}(n) = \frac{\mathbf{Z}(n)}{\mathbf{Z}_{\text{total}}(n)} \\
 \mathbf{Z} &= \left(\mathcal{X}_s^{(x_r, y_r)} \& \mathcal{R}(:, :, :, i_\alpha^*, i_s^*) \right) \prod_{k \neq 3} \times_k 1_{d_k} \mathbf{Z}_{\text{total}} \\
 &= \mathcal{R}(:, :, :, i_\alpha^*, i_s^*) \prod_{k \neq 3} \times_k 1_{d_k} \quad (17)
 \end{aligned}$$

where $1_{d_k} = \underbrace{[1, \dots, 1]}_{d_k}$, the operator $\&$ denotes the *logic and* operation, $V_{MS}(\cdot)$ denotes the MS, and $d_1 = M$, $d_2 = N$. Here, \mathbf{Z} is a vector, where the value in the i_θ th entry indicates the number of matched CPs with index number i_θ of the gradient angle. $\mathbf{Z}_{\text{total}}$ is a vector, where the value in the i_θ th entry indicates the number of CPs in the template image (undergoing rotation and scale transformation) with index number i_θ of the gradient angle. The value in the entry i_θ of \mathbf{y} (i.e., the value in i_θ th bin of histogram) denotes the proportion between the number

of matched CPs with index number i_θ of the gradient angle and the total number of CPs in the template with index number i_θ of the gradient angle. The MS score is a scale ranging from 0 to 1. The sparser is y , the larger is the MS score. Since the matched CPs of an IPS concentrate part of the contour, the y of an IPS is sparser than that of an object. This indicates that the IPS will obtain a larger MS than objects. For the examples in Fig. 5, the MS scores for the objects in the first and second rows are 0.179 and 0.185, respectively, whereas the MS scores for the IPSs in the fifth and sixth rows of Fig. 5 are 0.598 and 0.584, respectively.

Combined with the constructed MR score, the MS score and the number of votes, the final decision criteria in TGHT are given in

$$C(x_r, y_r) = \begin{cases} 1, & \text{if } HCS(x_r, y_r) \geq V_{th} \text{ and} \\ & V_{MR}(x_r, y_r) \geq V_{MR}^{th} \text{ and} \\ & V_{MS}(x_r, y_r) \leq V_{MS}^{th} \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where V_{th} , V_{MR}^{th} , and V_{MS}^{th} denote the threshold for the number of votes, the threshold of MR score, and the threshold of MS score, respectively. If $C(\cdot)$ is equal to 1, the object exists in the corresponding RP; otherwise, the object does not exist.

Note that there is a distance between different objects in RSIs. To prevent the detected objects from overlapping, we adopt the nonmaximum suppression method [30] to remove those RPs (x_r, y_r) whose number of votes is not the maxima in the HCS neighborhood of (x_r, y_r) . The final RPs are treated as the positions of the detected objects, and the bounding boxes with the proper size are used to mark the detection results.

V. EXPERIMENTS AND ANALYSIS

In the experiments, two datasets containing RSIs with different scenes were utilized to evaluate the performance of TGHT in terms of storage load, time consumption, and detection results. The detailed information of the two datasets are given as follows.

- 1) *Dataset 1*: This dataset contains 30 RSIs with different scenes collected from the publicly available Northwestern Polytechnical University very high resolution-10 (NWPUVHR-10) dataset [1]. These RSIs contain three types of objects with different orientations and sizes. Among these, 10 RSIs contain 94 airplanes, 10 RSIs include 227 oil tanks, and the remaining 10 RSIs contain 31 ships.
- 2) *Dataset 2*: This is the NWPU-2 dataset. There are 41 harbor images for the training and test sets, which, respectively, consist of 322 ships and 151 ships. Furthermore, these RSIs are extended by ten times using data augmentation, i.e., scale transform and rotation transform. Detailed information about NWPU-2 dataset can be found in [35].

Representative RSIs are displayed in Fig. 11, of which the three RSIs [see Fig. 11(a), (f), and (l)] are used to extract slices (see Fig. 6) as the three types of template image. The contours of the template images are generated by edge detection and interference removal.

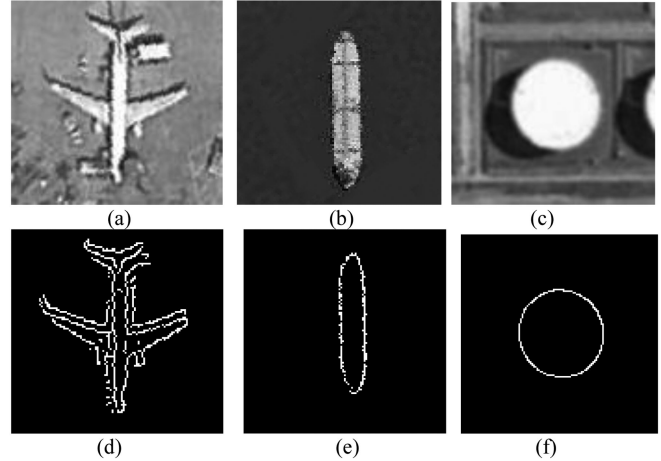


Fig. 6. Template image and contour: (a) template image of airplane, (b) template image of ship, (c) template image of oil tank, (d) contour of (a), (e) contour of (b), and (f) contour of (c).

The experiments consist of four parts. The impact of parameter setting on TGHT is detailed in Section V-A, and the generality of the defined two types of FAs is verified in Section V-B. In Section V-C, the main storage requirement and time consumption for TGHT are discussed and compared with those of the conventional GHT. In Section V-D, the performance of TGHT in terms of object detection is evaluated and compared with three representative contour extraction based object detection methods (i.e., GHT, IGHT, and RGA) and some state-of-the-art deep-learning-based object detection methods (i.e., HSF-Net, Fast R-CNN, and Faster R-CNN).

All the simulations are performed running on an i7-7700 Intel processor at 3.6 GHz and 8 GB memory with a Windows 10 system. The MATLAB interpreter consumes extra time to translate the algorithm, impacting the performance evaluation in terms of time consumption. Thus, to effectively evaluate the time consumption of TGHT and GHT, the simulations in Section V-C are implemented using Visual Studio 2017 (C++ programming language). However, the simulations in the other sections that are not related to time consumption are implemented using MATLAB 2018b.

A. Analyze the Impact of Parameter Setting on TGHT

The main parameters of TGHT are s_{\min} , s_{\max} , d_3 , d_4 , d_5 , V_{th} , and δ , which, respectively, denote the minimum possible scale of objects to be detected in RSIs, the maximum possible scale of objects to be detected in RSIs, the dimension of the gradient angle domain for the TR-R-table, the dimension of the rotation angle domain for the TR-R-table, the dimension of the scale domain for the TR-R-table, the threshold of the number of votes, and the threshold of the fuzzy strategy, as described in Section II. Since the parameters s_{\min} and s_{\max} can be determined by the size of objects to be detected in RSIs and the size of the object in the template, in our experiments, s_{\min} and s_{\max} were set to 0.5 and 2, respectively.

TABLE I
DETECTION RESULTS WITH VARIOUS d_3

d_3	5	8	10	12	15
$R_r(\%)$	83.4	76.6	66.4	52.6	36.2
$R_p(\%)$	38.3	63.7	86.2	91.5	93.1

TABLE II
DETECTION RESULTS WITH VARIOUS d_4

d_4	20	25	30	35	40
$R_r(\%)$	46.2	53.1	66.4	66.6	67.3
$R_p(\%)$	82.1	88.1	86.2	86.1	86.3

TABLE III
DETECTION RESULTS WITH VARIOUS d_5

d_5	5	8	10	12	15
$R_r(\%)$	39.6	56.3	66.4	66.5	67.0
$R_p(\%)$	88.1	89.2	86.2	85.3	85.9

TABLE IV
DETECTION RESULTS WITH VARIOUS V_{th}

V_{th}/N_t	0.36	0.39	0.42	0.45	0.48
$R_r(\%)$	94.7	86.2	66.4	45.7	30.9
$R_p(\%)$	46.6	66.4	86.2	95.6	96.7

To demonstrate the impact of parameter setting on TGHT and to select the optimal parameter setting for subsequent experiments, the detection results of 10 RSIs with airplanes under different parameter settings (see Tables I–V) are discussed in detail. Here, $R_r(\%)$, $R_p(\%)$, and N_t , respectively, denote the recall, precision, and number of CPs in the template. Since the gradient angle of CPs in the object of the RSI is slightly different from that of CPs in the object of the template image, a proper interval of gradient angle (i.e., $\Delta\theta$) ensures the CPs of the object in the template image and the RSI obtain the same index number of gradient angles. In other words, a proper d_3 ensures the corresponding CPs in objects of the RSI and the template image have the same attribute of gradient angle. If d_3 is too large, the corresponding CPs in the object of the RSI and the object of the template image will have different gradient angle attributes, and potential objects may be missed. From Table I, TGHT has low recall when $d_3 > 12$. Furthermore, Tables II and III show the precision and recall under different numbers of intervals for rotation angles (i.e., d_4) and scales (i.e., d_5), respectively. For a large d_4 or d_5 , the contour information of the object under different poses with slight differences is recorded in the TR-R-table, which means that potential objects with different poses are more easily detected. It is observed that as d_4 or d_5 increases, recall improves. Since the entries of 2D-HCS with values larger than V_{th} are considered the positions of objects, a larger V_{th} indicates fewer FAs and detected objects. Thus, as V_{th} decreases, the recall of TGHT improves, but the precision of TGHT decreases. Because δ determines the threshold of the fuzzy strategy, a proper δ ensures TGHT is robust to the object shape with slight deformation. However, too large δ causes interferences with similar shapes to be mistaken as potential objects.

TABLE V
DETECTION RESULTS WITH VARIOUS δ

δ	0	1	2	3	4
$R_r(\%)$	57.8	66.4	68.4	69.1	69.0
$R_p(\%)$	91.3	86.2	73.9	62.1	56.1

TABLE VI
OPTIMAL PARAMETER SETTING OF TGHT

Parameter	d_3	d_4	d_5	V_{th}/N_t	δ
Value	10	30	10	0.42	1

From Table V, precision is reduced under large δ . Based on the abovementioned analysis, the determined optimal parameters (see Table VI) are adopted in the subsequent experiments.

Next, we consider the essential preprocessing step of edge detection. Considering that deep-learning-based edge detection methods can obtain better performance than the conventional Canny operator, we select a representative deep-learning-based method [i.e., the holistically-nested edge detection method (HED)] instead of the Canny operator to evaluate the detection results of TGHT. Here, the HED networks are considered image-to-image edge detection technology by means of a deep learning model that leverages fully convolutional neural networks and deep supervised nets. The HED networks have obtained excellent edge detection results for natural images (e.g., 0.782 ODS F -score on the BSD500 dataset), and the source code and the pretrained models are available in [34].

To demonstrate the impact of different edge detection methods on the performance of TGHT, the Canny operator and the HED-network-based edge detection method were utilized. Fig. 7 displays the edge detection results for different methods.

Compared to the Canny operator, the HED method can reduce the edges of interferences with small sizes by using multiscale and multilevel feature learning. As shown in Fig. 7, the HED method obtains a more complete contour of the object and fewer contours of interferences than the Canny operator. This indicates that the HED method can be used to effectively prevent interference from being determined as objects. To verify this interpretation, the optimal parameters of TGHT (see Table VI) were adopted to evaluate TGHT with the Canny operator and with the HED method. Table VII presents the corresponding detection results. Notably, TGHT with the HED method outperformed TGHT with the Canny operator in terms of recall and precision. However, the HED method requires many training samples, whereas the Canny operator does not need any training samples. To ensure the proposed TGHT can be used to detect objects by using limited samples, TGHT with the Canny operator is adopted in our experiments.

B. Verify the Generality of FAs Caused by ICC and IPS in RSI

In Section IV, the representative examples of FAs caused by ICC and IPS are given in Fig. 5. In this section, more experiments are conducted on 30 RSIs to verify the generality of FAs caused by ICC and IPS.

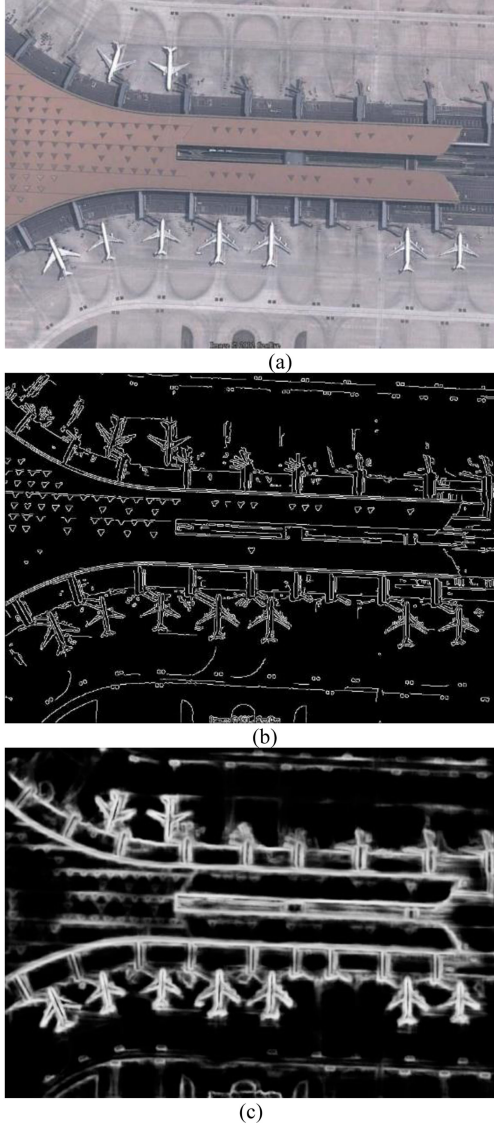


Fig. 7. Edge detection results for different methods. (a) RSI, (b) edge detection results of the Canny operator, (c) edge detection results of the HED method.

TABLE VII
COMPARISON OF OBJECT DETECTION RESULTS FOR DIFFERENT
EDGE DETECTION METHODS

Method	Canny	HED
$R_d(\%)$	68.4	75.6
$R_p(\%)$	73.9	78.2

1) *Verify the Generality of FAs Caused by ICC*: According to the definition of ICC, i.e., the noninteresting objects with more CPs than objects to be detected, the number of CPs for slices with objects (denoted SO) and slices with noninteresting objects (denoted SNO) in RSIs are counted to quantitatively illustrate the generality of ICCs. In detail, for 30 RSIs, we extract a slice centered at each position of these RSIs to count the number of CPs, and then obtain the statistical results for different object detection tasks, as shown in Table VIII. Here N_{cp}^o , N_{cp}^u , and P^u , respectively, denote the largest number of CPs for all the SOs,

TABLE VIII
STATISTICAL RESULTS FOR THE NUMBER OF CPs FOR OBJECTS TO BE
DETECTED AND NONINTERESTING OBJECTS

	N_{cp}^o	N_{cp}^u	P^u
Airplane detection	3225	5341	12.61%
Oil tank detection	533	2456	21.63%
Ship detection	2611	3892	5.36%

TABLE IX
STATISTICAL RESULTS ABOUT THE MS SCORES FOR OBJECTS
TO BE DETECTED AND FAS

	V_{MS}^O	V_{MS}^F	P_{MS}^U
Airplane detection	0.44	0.71	21.56%
Oil tank detection	0.32	0.69	45.5 %
Ship detection	0.51	0.99	69.4 %

the largest number of CPs for all the SNOs, and the proportion of SNOs containing more than N_{cp}^o CPs in all the SNOs. From Table VIII, we see that N_{cp}^o for different object detection tasks are different because the SOs have different numbers of CPs for different object detection tasks. For the different object detection tasks, N_{cp}^u is larger than the corresponding N_{cp}^o , which indicates that there are always some SNOs (i.e., ICC) with more CPs than SOs. In particular, it was observed that P^u for airplane detection, ship detection, and oil tank detection were 12.61%, 21.63%, and 5.36%, respectively. This means that the ICCs occupy a moderate proportion of detected items for different object detection tasks.

These ICCs have a large number of CPs, some of which are coincidentally matched by template image. Therefore, the ICCs can easily obtain sufficient votes to be mistaken as objects. To validate this assumption, the relationship between the average number of votes and the number of CPs in SNOs are counted and displayed in Fig. 8. It is seen from Fig. 8 that as the number of CPs in an SNO increases, the corresponding votes increase, demonstrating that ICCs easily obtain enough votes to be mistaken as objects, as assumed. Therefore, we conclude that the FAs caused by ICCs are widespread for contour extraction based object detection methods.

2) *Verify the Generality of FAs Caused by IPS*: The IPS in our article is defined as the interference that is partially similar to objects to be detected. To quantify this, IPS is defined as interference with a large MS score. To demonstrate the generality of FAs caused by IPSs, different candidate objects containing objects and FAs whose number of votes and MR score exceed thresholds are collected from 30 RSIs to calculate the corresponding MS scores. Table IX shows the statistical results for different object detection tasks. Here, V_{MS}^O , V_{MS}^F , and P_{MS}^U , respectively, denote the largest MS score for all the true objects, the largest MS score for all the FAs, and the proportion of FAs (i.e., IPS) with MS score larger than V_{MS}^O of all the FAs. It can be seen that V_{MS}^F is larger than V_{MS}^O for different object detection tasks,

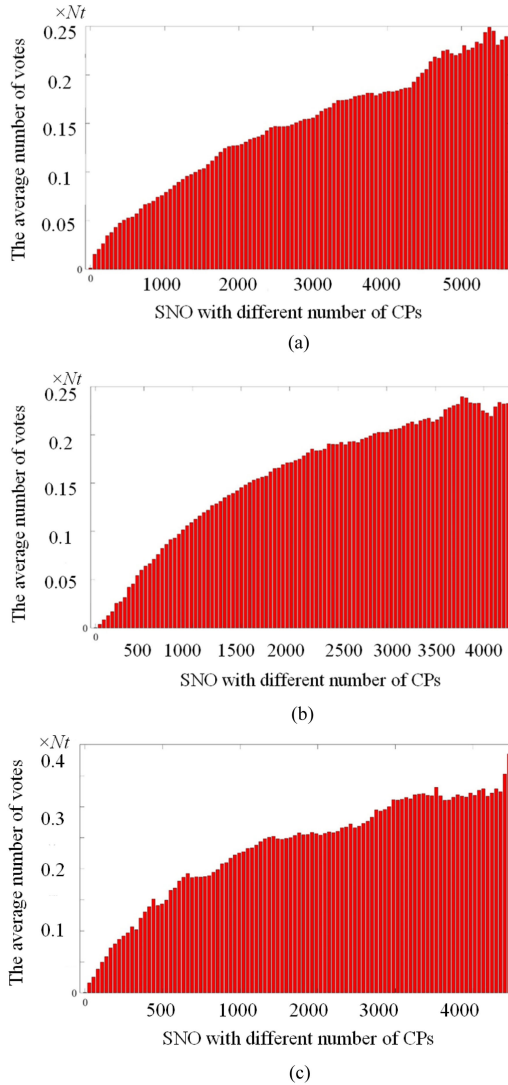


Fig. 8. The average number of votes for SNOs under different object detection tasks. (a) The average number of votes for SNO with different numbers of CPs for airplane detection. (b) The average number of votes for SNO with different numbers of CPs for oil tank detection. (c) The average number of votes for SNO with different numbers of CPs for ship detection.

which indicates that there are always some interferences (i.e., IPSs) with larger MS scores than some objects. In particular, the proportion of FAs caused by IPSs in all the FAs for airplane detection, ship detection, and oil tank detection were 21.56%, 45.5%, and 69.4%, respectively. These results indicate that FAs caused by IPSs are widespread for different object detection tasks.

C. Evaluation of Main Storage Requirements and Time Consumption for TGHT

One of the main advantages for TGHT is that, in contrast to GHT, it utilizes 2D-HCS instead of 4D-HCS without using a complex invariant feature. To verify this advantage, the storage requirement and time consumption for TGHT and GHT are analyzed in detail.

TABLE X
STORAGE REQUIREMENTS FOR MULTIORDER BT

	N_{node}^1	N_{node}^2	N_{node}^3	N_{index}^4 (N_{index}^5)	N_{total}
Airplane	161	20329	145872	961015	2088392
Oil tank	85	5144	8622	216252	446355
Ship	101	9732	45847	243005	541690

1) *Evaluate the Main Storage Requirement of TGHT*: According to the procedure of implementing GHT (see Section II-B) and the procedure of implementing TGHT (see Section III), the main storage load for TGHT is generated by 2D-HCS and the multiorder BT, whereas the main storage load for GHT is generated by 4D-HCS and the R -table.

To evaluate the main storage requirements of TGHT and GHT, the storage requirements for the multiorder BT, 2D-HCS, R -table, and 4D-HCS are discussed.

Note that the sizes of the multiorder BT and the R -table relate to the specific template image. Therefore, template images with three types of objects (see Fig. 5) are used to extract contour information and generate the corresponding multiorder BT and R -table for TGHT and GHT, respectively. For TGHT, the corresponding parameter setting is shown in Table VI. To ensure a fair comparison, the number of entries indexed by gradient angles for the R -table in GHT was set to 10, and the number of intervals for the rotation angle domain and for the scale domain in 4D-HCS were set to 30 and 10, respectively.

The multiorder BT consists of four parts, i.e., first-order BT, second-order BT, third-order BT, and the index numbers for the rotation angles and scales recorded in $\{S_4^{(i,j,i\theta)} | (i, j, i\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i'_\alpha, i'_s\}$. To evaluate the storage requirement for the multiorder BT built according to different template images, the number of nodes for the first-order BT, the number of nodes for the second-order BT, the number of nodes for the third-order BT, and the number of index numbers recorded in $\{S_4^{(i,j,i\theta)} | (i, j, i\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i'_\alpha, i'_s\}$ are counted, as shown in Table X. Here, N_{node}^1 , N_{node}^2 , N_{node}^3 , N_{index}^4 , N_{index}^5 , and N_{total} , respectively, denote the number of nodes in the first-order BT, the number of nodes in the second-order BT, the number of nodes in the third-order BT, the number of index numbers for the rotation angles recorded in $\{S_4^{(i,j,i\theta)} | (i, j, i\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i'_\alpha, i'_s\}$, the number of index numbers for the scales recorded in $\{S_4^{(i,j,i\theta)} | (i, j, i\theta, i'_\alpha, i'_s) \in S_{\mathcal{R}}, \forall i'_\alpha, i'_s\}$, and the total number of required storage cells. Since each node in the multiorder BT records the corresponding key, each node requires a storage cell (without considering the data structure). Note that both the element of 4D-HCS and the node in the multiorder BT are considered as floating point variables for implementation on the computer. In our article, the size of the storage cell is defined as the size of a storage space for a floating point value (i.e., 4 B). Therefore, the total number of required storage cells can be calculated by $N_{total} = N_{node}^1 + N_{node}^2 + N_{node}^3 + N_{index}^4 + N_{index}^5$, as shown in Table X.

The R -table used in GHT records all the reference vectors and corresponding index numbers of the gradient angles as

TABLE XI
STORAGE REQUIREMENTS FOR THE R -TABLE

	N_e	N_{rp}	N_{total}
Airplane	10	662	1324
Oil tank	10	193	396
Ship	10	108	226

TABLE XII
STORAGE REQUIREMENT FOR GHT AND TGHT UNDER DIFFERENT OBJECT DETECTION TASKS (UNITS: MILLION STORAGE CELLS)

	Airplane	Oil tank	Ship
GHT	202	202	202
TGHT	2.76	1.12	1.21

entries. Therefore, the main storage requirement of the R -table includes the number of entries and the number of reference vectors. Table XI shows detailed storage requirements of the R -table for different template images. Here, N_e , N_{rp} , and N_{total} , respectively, denote the number of entries for the R -table, the number of reference vectors, and the total number of required storage cells. Note that each entry of the R -table consumes a storage cell to record the corresponding index number of the gradient angle, and each reference vector requires two storage cells. Therefore, the total number of required storage cells for the R -table is calculated by $N_{total} = N_e + 2 \times N_{rp}$, as shown in Table XI.

Next, compare the storage requirements of 4D-HCS and 2D-HCS, which relate to the size of the RSI. An example RSI [see Fig. 11(a)] of size 1039×649 was used to calculate the storage requirement of 4D-HCS and 2D-HCS. According to the parameter setting, the number of intervals for the rotation angle domain in 4D-HCS was equal to 30, and the number of intervals for the scale domain in 4D-HCS was equal to 10. Therefore, the number of storage cells for $4D-HCS \in R^{1039 \times 649 \times 30 \times 10}$ was nearly 202 million as calculated by $1039 \times 649 \times 30 \times 10$. For $2D-HCS \in R^{1039 \times 649}$, the number of storage cells was nearly 0.67 million as calculated by 1039×649 .

Through the abovementioned analysis, the comparison of the main storage requirements of TGHT (i.e., the storage cells required by the multiorder BT and 2D-HCS) and GHT (i.e., the storage cells required by the R -table and 4D-HCS) are summarized in Table XII.

As seen from Table XII, the main storage requirement for TGHT is much smaller than that of GHT. For RSIs with larger size, 4D-HCS requires much more storage cells than 2D-HCS. Therefore, the storage requirement of GHT is much larger than that of TGHT. Fig. 9 displays a detailed comparison of main storage requirements of TGHT and GHT for RSIs with different sizes. It is observed that as the RSI size increases, the advantage of TGHT in terms of storage requirement becomes increasingly obvious.

2) *Evaluate Time Consumption of TGHT*: The performance of TGHT is evaluated in terms of time consumption compared to the conventional GHT. To ensure a fair comparison, both GHT and TGHT are implemented with C++ programming language

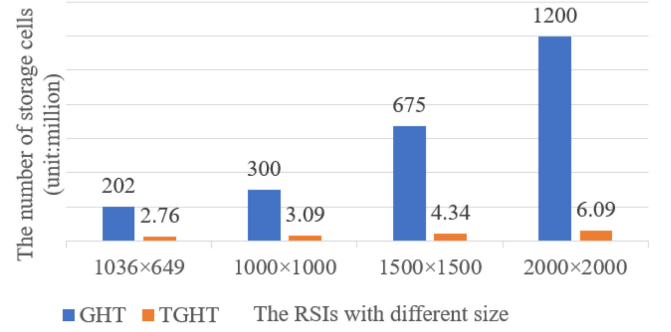


Fig. 9. The number of storage cells required by TGHT and GHT under RSI with different sizes for airplane detection.

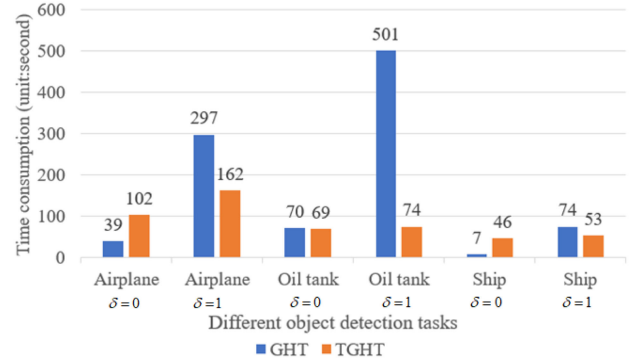


Fig. 10. The comparison of time consumption for GHT and TGHT.

under the same conditions (including the same RSI, the same template image, and the same edge detection operator). Fig. 10 shows the time consumption for TGHT and GHT under different object detection tasks and different thresholds of fuzzy voting (i.e., δ).

Three RSIs [see Fig. 11(c), (e), and (i)] were selected for airplane detection, oil tank detection, and ship detection, respectively. From Fig. 10, the different object detection tasks resulted in different time consumptions. Since the RSI with ships [see Fig. 11(i)] contains the fewest CPs, the corresponding time consumption is lowest. Note that the fuzzy voting strategy will bring more votes. From Fig. 10, as δ increases, the time consumption of both TGHT and GHT increases. In addition, GHT may bring repeat votes when applying the fuzzy voting strategy, whereas TGHT avoids repeat votes by using the binary TR- R -table. Therefore, the total number of votes for TGHT will be smaller than that of GHT when $\delta > 0$ (i.e., the fuzzy voting strategy is used). Furthermore, as seen in Fig. 10, the time consumption of TGHT is lower than for GHT when $\delta > 0$. For $\delta = 0$, the time consumption of TGHT is larger than that of GHT. Thus, the difference in time consumption between TGHT and GHT is small, indicating that TGHT hardly increases the time consumption by using tensor operations.

D. Comparison With Other Representative Methods

1) *Comparison With Well-Known Contour Extraction Methods*: To highlight the superiority of TGHT in terms of detection performance, 30 RSIs with three types of objects were applied to

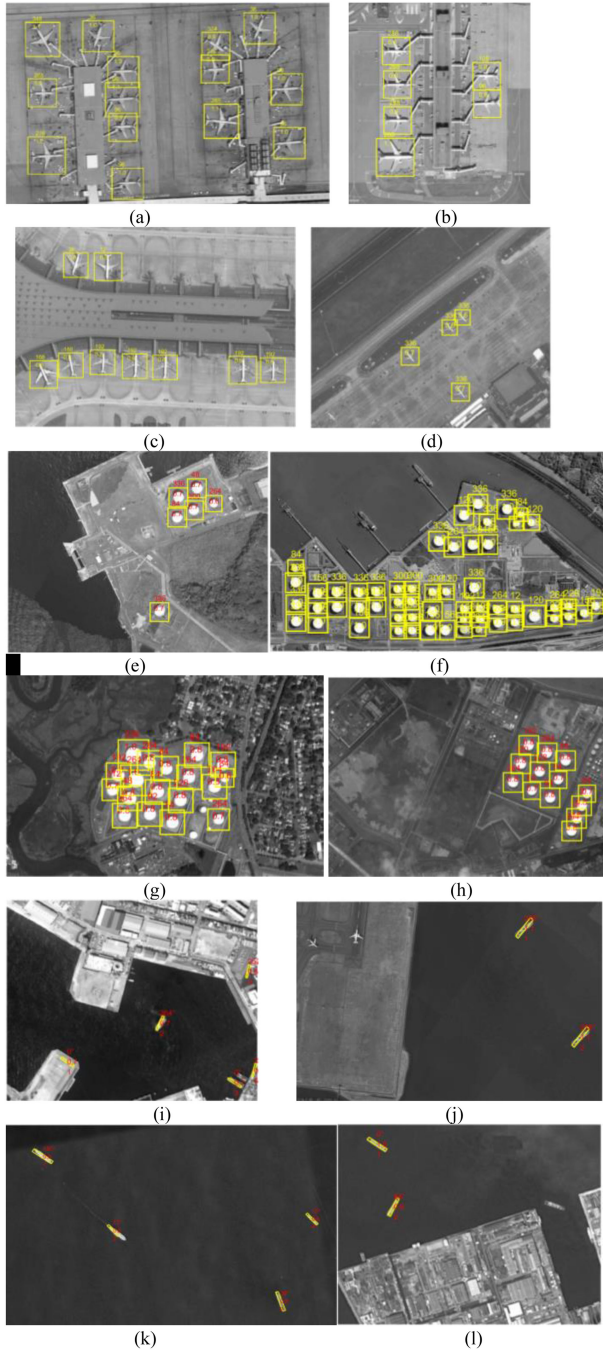


Fig. 11. Representative detection results for TGHT: (a–d) detection results of airplane, (e–h) detection results of oil tank, (i–l) detection results of ship.

compare TGHT with representative contour extraction methods (i.e., IGHT, RGA, and GHT). Fig. 11 shows part of the detection results obtained by TGHT. In detail, TGHT and competitors with a template image of an airplane were tested on RSIs containing airplanes, oil tanks, and ships.

For the contour extraction methods, the threshold of the number of votes (i.e., V_{th}) determined the number of detected objects and the number of FAs. To evaluate the performance of different contour extraction methods, V_{th} for all methods were regulated to obtain different numbers of detected objects and

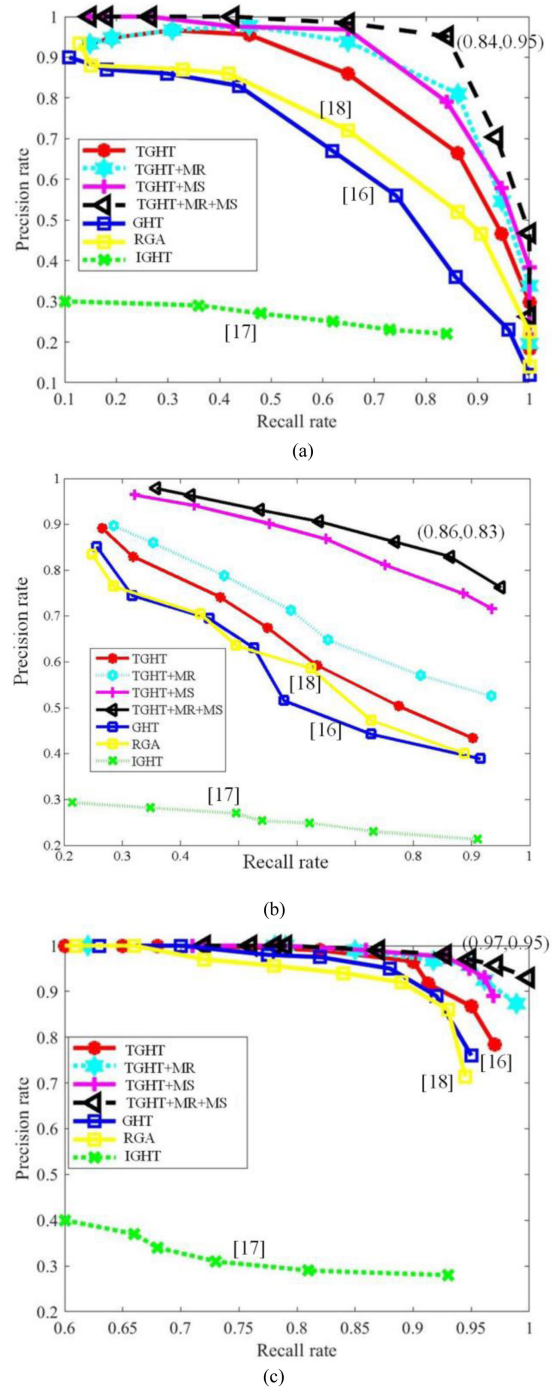


Fig. 12. PR curve of TGHT and contour extraction based methods: (a) PR curve for airplane detection, (b) PR curve for ship detection, (c) PR curve for oil tank detection.

FAs. Fig. 12 shows the generated precision–recall (PR) curves [32], [33].

As shown in Fig. 12, IGHT has the worst detection results because the pairwise-point-based feature is susceptible to interference, resulting in false pairwise-points weakening the detection results. Since the circular structure of the oil tank was distinctive in RSIs, almost all the methods (except IGHT) obtained excellent detection results. For airplane and ship detection, TGHT

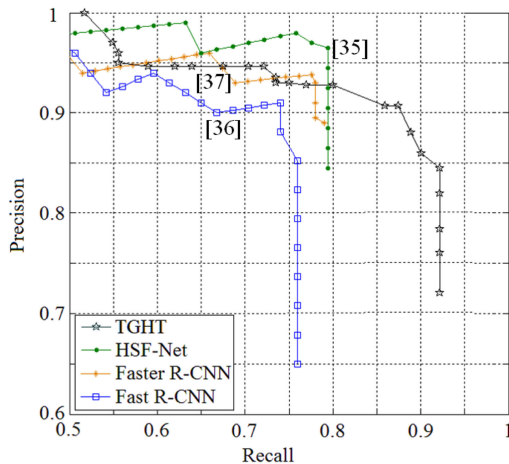


Fig. 13. PR curve of TGHT and deep-learning-based methods.

obtained slightly better detection results than GHT and RGA because TGHT can obtain a more accurate number of votes for objects. The MR-criterion-based strategy ($V_{MR}^{th} = 0.05$) and MS-criterion-based strategy ($V_{MS}^{th} = 0.4$) were effective in further improving the detection results. The best detection results were obtained when using both MR and MS criteria based strategies. In particular, for airplane detection, the precision and recall for TGHT with two FA removal strategies simultaneously reached 0.95 and 0.84, respectively. For ship detection, they reached 0.83 and 0.86, respectively. For airplane detection, they reached 0.95 and 0.97, respectively. These results indicate that TGHT can obtain excellent detection results for different object detection tasks in different scenes.

2) *Comparison With State-of-the-Art Deep Learning Methods:* To further verify the performance of TGHT, three state-of-the-art deep-learning-based object detection methods, i.e., HSF-Net [35], Fast R-CNN [36], and Faster R-CNN [37], were selected to evaluate TGHT in terms of PR curves using the publicly available NWPU-2 dataset. HSF-Net was built recently to achieve multiscale ship detection in RSIs, and Fast R-CNN and Faster R-CNN are treated as representative deep-learning-based object detection methods that have obtained excellent object detection results in natural images and RSIs.

Considering the detected objects in the NWPU-2 dataset are ships, the region growing based sea-land segmentation method [25] was introduced as a preprocessing step of the proposed TGHT method. To obtain the PR curve, the parameter V_{th} of TGHT was tuned to generate different detection results. Through a comparison of PR curves (see Fig. 13) between TGHT and three state-of-the-art deep-learning-based methods obtained from [35] and the average precision (AP) (see Table XIII), we see that TGHT outperformed Fast R-CNN and Faster R-CNN and obtained comparable detection results to HSF-Net (i.e., HSF-Net obtained a higher precision, and the proposed TGHT obtained a higher recall). The proposed TGHT method obtained the best AP among all the methods. Note that the deep-learning-based methods use a large number of training samples (i.e., 322 ships). In comparison, the proposed TGHT only uses a single sample to generate the TR-R-table to detect objects. Thus, the proposed

TABLE XIII
AP (%) OF DIFFERENT METHODS

	Fast R-CNN	Faster R-CNN	HSF-Net	TGHT
AP	71.05%	72.15%	80.52%	87.55%

TGHT is effective for object detection in RSIs, especially for cases where training samples are not sufficient.

VI. CONCLUSION

To improve performance of existing contour extraction technologies in terms of detection results and storage load, the TGHT is proposed to detect effectively objects with different orientations and sizes in RSIs. The main contributions of TGHT can be concluded as three aspects.

- 1) Compared to existing contour extraction methods utilizing a 4D-HCS or complex invariant features to detect objects with unknown orientations and sizes in RSI, TGHT uses a novel tensor space voting mechanism to accumulate votes by avoiding using either a large Hough counting space (4D-HCS) or complex invariant features of other GHT extensions.
- 2) Compared to existing FAs removal methods analyzing the characteristic of FAs qualitatively, TGHT provides analytical expression to reveal the cause of FAs quantitatively, and then removes these FAs by using effective strategies.
- 3) Compared to existing contraction methods that do not obtain the accurate votes for potential object caused by fuzzy voting and coordinate quantification strategies, the TGHT utilizes the tensor space contour representation and voting mechanism to obtain accurate votes for potential objects, which ensures the effective object detection results.

Furthermore, a remarkable fact is that the tensors used in TGHT consume small storage load by using the multiorder-BT-based searching method. As a result, the TGHT significantly reduces the storage requirement compared with GHT by using small time consumption. The limitation of TGHT is the deformed and incomplete object contour that will easily lead the missing alarms. The future work is to investigate tensor-decomposition-based robust contour representation to improve the detection results in RSIs with complex scenes.

REFERENCES

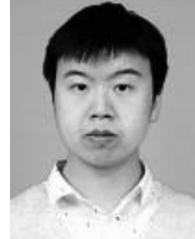
- [1] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
- [2] Y. Hu, X. Li, N. Zhou, L. Yang, L. Peng, and S. Xiao, "A sample update-based convolutional neural network framework for object detection in large-area remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 947–951, Jun. 2019.
- [3] T. Li, Z. Liu, R. Xie, and L. Ran, "An improved superpixel-level CFAR detection method for ship targets in high-resolution SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 184–194, Jan. 2018.
- [4] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.
- [5] W. Li, Y. Zhang, N. Liu, Q. Du, and R. Tao, "Structure-aware collaborative representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7246–7261, Sep. 2019.

- [6] D. Zhang, J. Han, G. Cheng, Z. Liu, S. Bu, and L. Guo, "Weakly supervised learning for target detection in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 701–705, Apr. 2015.
- [7] 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.
- [8] C. Li, R. Cong, J. Hou, S. Zhang, Y. Qian, and S. Kwong, "Nested network with two-stream pyramid for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9156–9166, Nov. 2019.
- [9] R. Cong, J. Lei, H. Fu, M. Cheng, W. Lin, and Q. Huang, "Review of visual saliency detection with comprehensive information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 2941–2959, Oct. 2019.
- [10] G. Liu, Y. Zhang, X. Zheng, X. Sun, K. Fu, and H. Wang, "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.
- [11] A. Manno-Kovacs, "Direction selective contour detection for salient objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 375–389, Feb. 2019.
- [12] Q. Zhang, L. Zhang, W. Shi, and Y. Liu, "Airport extraction via complementary saliency analysis and saliency-oriented active contour model," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 7, pp. 1085–1089, Jul. 2018.
- [13] N. Liu, Z. Cui, Z. Cao, Y. Pi, and S. Dang, "Airport detection in large-scale SAR images via line segment grouping and saliency analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 434–438, Mar. 2018.
- [14] A. O. Ok, "A new approach for the extraction of aboveground circular structures from near-nadir VHR satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3125–3140, Jun. 2014.
- [15] I. Zingman, D. Saupe, O. A. B. Penatti, and K. Lambers, "Detection of fragmented rectangular enclosures in very high resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4580–4593, Aug. 2016.
- [16] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.
- [17] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2nd ed., Amsterdam, The Netherlands: Elsevier, 2008.
- [18] Y. Lin, H. He, Z. Yin, and F. Chen, "Rotation-invariant object detection in remote sensing images based on radial-gradient angle," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 746–750, Apr. 2015.
- [19] J. A. R. Artolazabal, J. Illingworth, and A. S. Aguado, "LIGHT: Local invariant generalized Hough transform," in *Proc. 18th Int. Conf. Pattern Recognit.*, Hong Kong, China, 2006, pp. 304–307.
- [20] H. Yang, S. Zheng, J. Lu, and Z. Yin, "Polygon-invariant generalized Hough transform for high-speed vision-based positioning," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1367–1384, Jul. 2016.
- [21] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.
- [22] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.
- [23] H. He, Y. Lin, F. Chen, H. Tai, and Z. Yin, "Inshore ship detection in remote sensing images via weighted pose voting," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3091–3107, Jun. 2017.
- [24] J. Xu, X. Sun, D. Zhang, and K. Fu, "Automatic detection of inshore ships in high-resolution remote sensing images using robust invariant generalized Hough transform," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2070–2074, Dec. 2014.
- [25] H. Chen, T. Gao, W. Chen, Y. Zhang, and J. Zhao, "Contour refinement and EG-GHT-based inshore ship detection in optical remote sensing image," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8458–8478, Nov. 2019.
- [26] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [27] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1485–1490, Sep. 2005.
- [28] F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming*. New York, NY, USA: McGraw-Hill, 1990.
- [29] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.
- [30] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit.*, Hong Kong, China, 2006, pp. 850–855.
- [31] I. Al-Furajh, S. Aluru, S. Goil, and S. Ranka, "Parallel construction of multidimensional binary search trees," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 2, pp. 136–148, Feb. 2000.
- [32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [33] W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang, "Object detection in high-resolution remote sensing images using rotation invariant parts based model," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 74–78, Jan. 2014.
- [34] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vision*, Santiago, Chile, 2015, pp. 1395–1403.
- [35] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.
- [36] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, Santiago, Chile, 2015, pp. 1440–1448.
- [37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 1, 2017.



Hao Chen (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2001, 2003, and 2008, respectively.

Since 2004, he has been with the School of Electronics and Information Engineering, Harbin Institute of Technology. He is currently a Professor. His main research interests include remote sensing image processing and image compression.



Tong Gao received the B.S. degree from Hubei University, Wuhan, China, in 2015, and the M.S. degree from Inner Mongolia University, Hohhot, China, in 2017. He is currently working toward the Ph.D. degree in information and communication engineering with the Harbin Institute of Technology, Harbin, China.

His research interests include multisource information fusion and object recognition.

Guodong Qian received the B.S. and M.S. degrees from the National University of Defense Technology, Changsha, China, in 2006 and 2014, respectively.

He is currently with the Institute of Remote Sensing Information of Beijing, Beijing, China. His research interests include image processing and image analysis.



Wen Chen (Student Member, IEEE) received the B.S. degree from the Harbin Institute of Technology at Weihai, Weihai, China, in 2016, and the M.S. degree from the Harbin Institute of Technology, Harbin, China, in 2019. He is currently working toward the Ph.D. degree in information and communication engineering with the School of Electronics and Information Engineering, Harbin Institute of Technology.

His research interests include image processing and object detection.



Ye Zhang (Member, IEEE) received the B.S. degree in communication engineering, and the M.S. and Ph.D. degrees in communication and electronic system from the Harbin Institute of Technology, Harbin, China, in 1982, 1985, and 1996, respectively.

He is currently a Professor and Doctoral Supervisor with the Department of Information and Communication Engineering, Harbin Institute of Technology. His research interests include remote sensing image analysis, image/video compression and transmission, and multisource information collaboration processing.