Target Detection for VHF SAR Ground Surveillance

Wenxing Ye, Christopher Paulson, Dapeng Wu

Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida, 32611
Correspondence author: Prof. Dapeng Wu, E-mail: wu@ece.ufl.edu,
http://www.wu.ece.ufl.edu

Abstract

In this paper, we develop a target detection algorithm based on a supervised learning technique that maximizes the margin between two classes, i.e., the target class and the non-target class. Specifically, our target detection algorithm consists of 1) image differencing, 2) maximum-margin classifier, and 3) diversity combining. The image differencing is to enhance and highlight the targets so that the targets are more distinguishable from the background. The maximum-margin classifier is based on a recently developed feature weighting technique called Iterative RELIEF; the objective of the maximum-margin classifier is to achieve robustness against uncertainties and clutter. The diversity combining utilizes multiple images to further improve the performance of detection, and hence it is a type of multi-pass change detection. We evaluate the performance of our proposed detection algorithm, using the CARABAS-II synthetic aperture radar (SAR) image data and the experimental results demonstrate superior performance of our algorithm, compared to the benchmark algorithm.

Key words: Index Terms-VHF SAR, CARABAS-II, target detection, high dimensional classification, I-RELIEF

1. Introduction

Detecting targets concealed in foliage or camouflage in a large area is a challenge problem and is critical for ground surveillance. It has many applications such as detecting the deployment of enemies hidden in the forest, locating the position of an accident in forest rescue activities and marking the foliage-covered terrain changes.

There are mainly two difficulties for this problem, namely, 1) how to remotely acquire information that contains important features of foliage-concealed targets, and 2) how to distinguish targets from background and clutter. Synthetic aperture radar (SAR) operated at low Very High Frequency (VHF)-band is a good solution to the first problem [1][2]. At the low VHF-band around 20MHz - 90MHz, radar wave is more likely to make a return on objects exceeding certain dimension. Since this dimension is usually much larger than the leaves and branches under which targets are concealed, radar signal can penetrate the forest canopy and get reflected by the targets under it. These backscatters describe the scene covered and are used to form SAR images in which large objects show themselves as bright areas. In this way, VHF-band SAR technology transforms the foliage penetration problem into a traditional image based target detection problem. The second problem is also known as automatic target detection (ATD) problem. Major techniques for ATD include adaptive boosting [3], extended fractal feature [4], genetic programming [5], multiscale autoregressive (MAR), multiscale autoregressive moving average (MARMA) models, singular value decomposition (SVD) methods [6] and constant false alarm rate (CFAR) processing [7]. CFAR processing is widely used to give a globally applicable threshold for a constant probability of false alarms through estimating and removing the local background statistics.

According to Lundberg et al. [8], the main technical challenge in designing an ATD algorithm lies in how to keep the false alarm rate at a low level while yet achieving high detection rate. Suppressing false alarms is especially important in the case of detecting concealed targets because the foliage can add substantial amount of noise to the image. The denser the forest is, the more noisy the image looks. In certain cases, it is almost impossible to distinguish targets from noise and background clutter such as huge trunks and rocks, given a single image. To mitigate this problem, multiple images can be used to suppress noise and background clutter.

For ground surveillance, it is reasonable to assume that targets are in the areas of change between two images taken at different times while background clutter is unchanged between the two images. Hence, the target detection problem can be addressed by change detection techniques, if multiple images are given. The objective of change detection is to find areas of change between an image under test and a reference image; here, an image under test may contain targets, and a reference image is an image of the same geographical region as the image under test; a reference image is captured at a different time and may not contain targets or contain targets that are in totally different locations, compared to the image under test. The changed areas or differences between two images may contain targets of interest, and the differencing can greatly suppress the background noise and clutter. Hence, change detection can help increase the probability of detection and reduce the false alarm rate [9][10][11][12].

In this paper, we propose a target detection scheme that leverages change detection and a max-margin classifier based on nonlinear I-RELIEF feature weighting technique. In our target detection scheme, change detection helps separate targets from static background; the max-margin classifier makes our algorithm more robust to noise and unexpected clutter; we also use diversity combining to boost the performance of the algorithm further. Specifically, the first step is image differencing between an image under test and a reference image. Here, we assume that all the images are geometrically registered so that the same pixel in two images corresponds to the same geographical location; and all the images are radiometrically adjusted so that the lighting variation between two images is removed. Then, feature extraction, feature weighting and distance-ratio-based classification are applied to the difference image. Refer to Section 4 for more details. Diversity combining means combining signals from multiple sources into a single improved signal. In our case, differencing images between a specific image under test and multiple independent reference images are considered as diversity sources. After the same classification process on each source, multiple decisions are obtained independently. Majority voting among all the decisions gives a final decision, which is more reliable. Of course, the number of sources needs to be odd as required by majority voting. Fig. 1 shows the flow chart of our proposed algorithm.

We evaluate our proposed algorithm using a public released data set [13] acquired with the airborne CARABAS-II system which produces SAR images at low VHF-band around 20MHz – 90MHz during a flight campaign held in northern Sweden. It includes 24 images with 4 different targets deployments

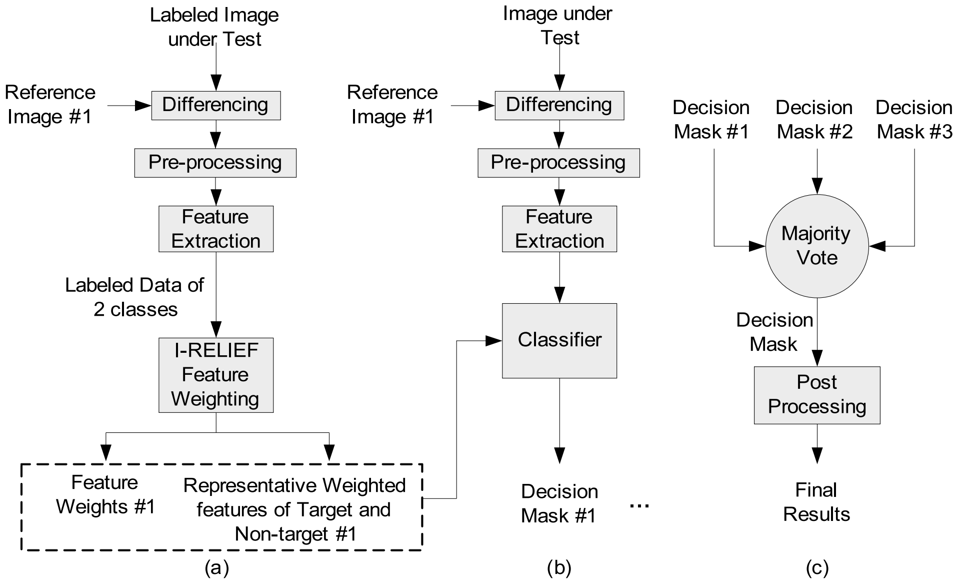


Figure 1: Flow chart: (a) learning; (b) testing; (c) diversity combining

and 6 different flight passes for each deployment [8]. The image corresponding to mission 3 pass 5 was used for learning and the rest 23 images were used for testing. With different threshold values for distance ratio in the classifier, a performance curve of correct detection rate versus number of false alarms is obtained. Compared with the benchmark algorithm associated with the data set [8], our proposed scheme achieves much lower false alarm rate while yet achieving the same target-detection probability.

The remainder of the paper is organized as below. In Section 2, we describe the image data set used for target detection. In Section 3, a baseline algorithm associated with the data set is presented. Section 4 presents our proposed scheme, which consists of four parts, namely, feature extraction, feature weighting, classification, and post processing. Section 5 shows our experimental results and compare our proposed scheme with the baseline algorithm. Finally, Section 6 concludes the paper.

2. Data Description

The image data set used for the evaluation of our proposed target-detection scheme is a subset of data collected during a flight campaign held in Sweden in

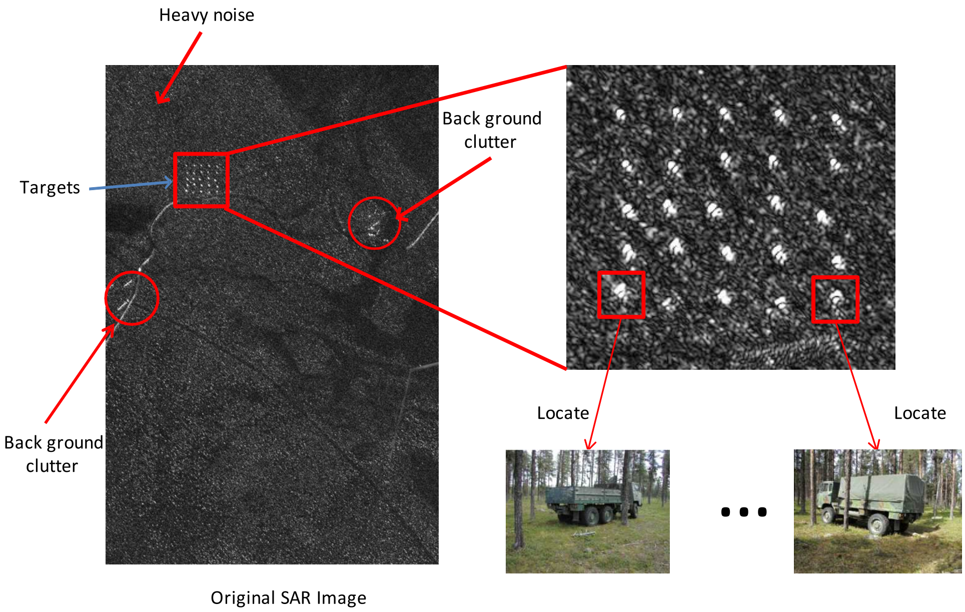


Figure 2: A sample image from the data set showing the targets, noise and background clutters.

the early summer of 2002 [13]. The images were taken by CARABAS-II, the second generation ultra-wide-band SAR mounted on a Sabreliner airplane. The system was operated in the frequency range of 20MHz - 90MHz with corresponding wave lengths between 3.3 meters and 15 meters. This dimension is much larger than that of leaves and branches and close to dimensions of vehicles which are the imagined targets to be pursued.

In this data set, there are several disturbing factors such as heading difference between images, different target orientations, different target sizes and radio frequency interference. Figure 2 shows an image example from the data set. The campaign was run at a spot in northern Sweden which mainly include a river and two blocks of forests. A rectangular area of 3km by 2km is chosen as the focus of all the data. And the GPS parameters of the corners are given with the data set. The recorded SAR images are all in 3000 by 2000 pixel size. Each pixel in the image corresponds to an area of 1m by 1m on the ground. 25 vehicles of three types and four deployments are used as targets hiding in the forests. There are ten TGB11 ($4.4 \times 1.9 \times 2.2$ m), eight TGB30 ($6.8 \times 2.5 \times 3.0$ m) and seven TGB40 ($7.8 \times 2.5 \times 3.0$ m).

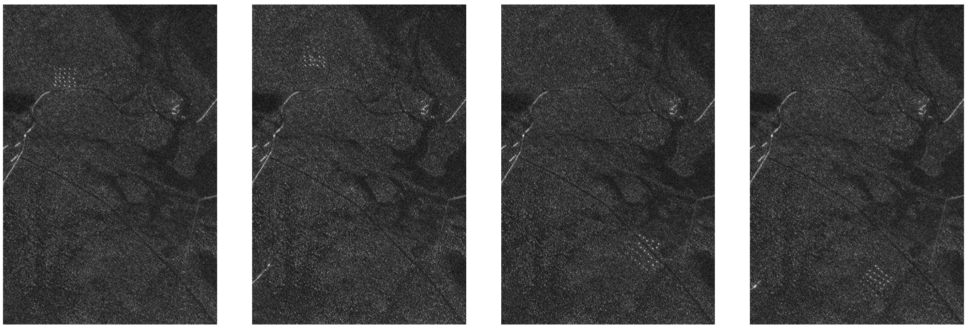


Figure 3: Four sample images from the data set with deployments Sigismund, Karl, Fredrik and Adolf-Fredrik from left to right respectively.

TGB11, TGB30 and TGB40 represent different kinds of vehicles. Sigismund, Karl, Fredrik and Adolf-Fredrik are the codes of the four deployments shown in Figure 3. The vehicles have totally different positions and orientations between different deployments. To guarantee the variety of the data set, for every single vehicle deployment, six different flight headings were adopted. So totally 24 images were acquired. There is a TV transmitter which is the source of radio frequency interference located south-east of the focused area. This affects the SAR imaging precesses and makes various contribution for different flight heading angles. Along with the data set, the actual position of each vehicles are given in the form of GPS parameters for the purpose of evaluation. Please refer to [8] for more information of the data set.

3. Baseline Algorithm for Target Detection

Along with the data set, a baseline algorithm and experimental results are given in [8]. Details about the algorithm are discussed in [14]. This algorithm is based on change analysis, statistic hypothesis test and CFAR normalization techniques. First of all, two pixel values are extracted from image under test I_t and reference image I_r at the corresponding position respectively. They are combined into a vector as:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{1}$$

It is assumed that each pixel value in the image is a random variable and values of different pixels are statistically independent. z_1, z_2 are two real-valued random variables and $\mathbf{z}$ is a random vector. Then, target or change, clutter and noise signal are defined in the similar way as:

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \; \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \; \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \tag{2}$$

where indices 1 and 2 indicate the two images. Hypothesis test is used to determine whether a change is present. The two hypotheses are defined as:

$$\begin{array}{ll} H_0 : \mathbf{z} = \mathbf{c} + \mathbf{n} = \mathbf{q} & \text{(no change)} \\ H_1 : \mathbf{z} = \mathbf{s} + \mathbf{c} + \mathbf{n} = \mathbf{s} + \mathbf{q} & \text{(change)} \end{array} \tag{3}$$

A test statistic of likelihood ratio is computed to test the two hypotheses. The test statistic is defined by:

$$\Delta(\mathbf{z}) = \frac{P(\mathbf{z}|H_1)}{P(\mathbf{z}|H_0)} \tag{4}$$

The random vectors of noise **n** and clutter **c** are assumed to have bivariate zero-mean circular Gaussian distribution. So the sum of the two **q** also follows bivariate zero-mean circular Gaussian distribution. The two probability density functions (PDF) of the likelihoods in (4) are:

$$P(\mathbf{z}|H_0) = \frac{1}{\pi^2|\mathbf{C}|}\mathbf{exp}(-\mathbf{z}^T\mathbf{C}^{-1}\mathbf{z}) \tag{5}$$

and

$$P(\mathbf{z}|H_1) = \frac{1}{\pi^2|\mathbf{C}|}\mathbf{exp}(-(\mathbf{z}-\mathbf{s})^T\mathbf{C}^{-1}(\mathbf{z}-\mathbf{s})) \tag{6}$$

where **C** is a 2 × 2 covariance matrix defined by:

$$\begin{aligned} \mathbf{C} &= E\{(\mathbf{z} - E\{\mathbf{z}\})(\mathbf{z} - E\{\mathbf{z}\})^T\} \\ &= \begin{pmatrix} \sigma_1^2 & \rho_{21}\sigma_2\sigma_1 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \end{aligned} \tag{7}$$

where $\sigma_k^2 = E\{q_k^2\}$ and $\rho_{kl} = E\{q_k q_l\}/(\sigma_k\sigma_l)$. The value of covariance matrix **C** is estimated using pixel values within smaller local image blocks. Then the statistic test is

$$\begin{array}{ll} |\mathbf{s}^T\mathbf{C}^{-1}\mathbf{z}| > \lambda & \rightarrow \text{decide } H_1 \\ |\mathbf{s}^T\mathbf{C}^{-1}\mathbf{z}| \leq \lambda & \rightarrow \text{decide } H_0 \end{array} \tag{8}$$

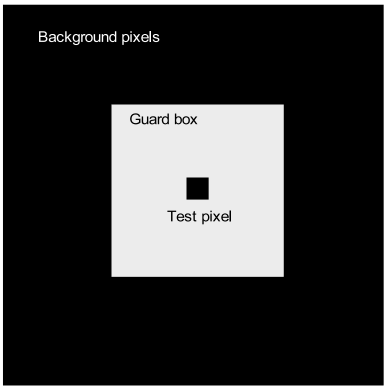


Figure 4: The area used for computing the background statistics surrounding a pixel under test

It is assumed that the target is not present in one image but present in the other image, that means:

$$\mathbf{s} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ or } \mathbf{s} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{9}$$

With the change detection algorithm described above, we can convert two SAR images into one binary image with '0' indicating no change and '1' indicating change. Next, we describe how to determine the value of threshold λ. Because the statistic features vary at different locations in an image, the threshold λ should also change for different locations. This change is difficult to predict for different images. To normalize the value of threshold λ, a CFAR detection filter is used. It is used to estimate the local background statistics in the change image and remove the variance between different locations. The shape of the CFAR filter is shown in Fig. 4. The background statistics, mean value and standard deviation are estimated based on the pixel values within the outer box but outside the inner box. The sizes of outer box and inner box are 31×31 and 19×19 respectively. Then the background statistics are compared with the pixel under test. The mean value is subtracted from the center pixel, then divided by the standard deviation. The resulting value is compared with a global threshold λ' to make a decision. After thresholding, morphological operations, i.e., one erosion operation and two dilation operations are applied to the binary change mask to remove false alarms.

4. Our Proposed Scheme for Target Detection

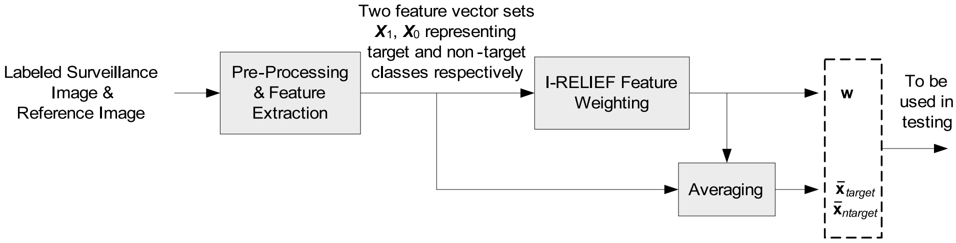

Labeled Surveillance Image & Reference Image
Pre-Processing & Feature Extraction
Two feature vector sets X_1, X_0 representing target and non-target classes respectively
I-RELIEF Feature Weighting
Averaging
w
$\bar{x}_{target}$
$\bar{x}_{ntarget}$
To be used in testing

Figure 5: Flow chart of the learning stage

The flow chart of our proposed target-detection scheme is shown in Fig. 1. The input of our algorithm is SAR images and the output of our algorithm is locations of possible targets, if there is any. In the situation considered in this paper, as in most situations, a target occupies an area of more than one pixel in an image. But the location of a target is usually specified by the coordinates of a single point, e.g., the centroid, the top-left corner point, or the very top point of the target. So the ATD task is divided into two subtasks: 1) label each pixel as target or non-target; 2) group connected target pixels into targets and extract their coordinates. Usually, the first part is much more challenging than the second part. If the image is correctly labeled, the second step becomes simple. The first subtask is actually classification. A carefully designed classifier is the core of this subtask. In our scheme, we use a distance-ratio-based classifier and train the classifier through supervised learning. The learning process is shown in Figure 5 and the testing procedure is shown in Figure 6. The major steps of our classification algorithm are feature extraction, feature weighting, and classification/target-detection, which are presented in the following sections, respectively.

4.1. Feature Extraction

Our feature extraction method is based on change analysis. The first step is to acquire change information. Simple differencing between image under test I_t and reference image I_r is adopted here. A difference image I_d is obtained by

$$I_d = \begin{cases} I_t - I_r & I_t - I_r > 0 \\ 0 & I_t - I_r \leq 0 \end{cases} \tag{10}$$

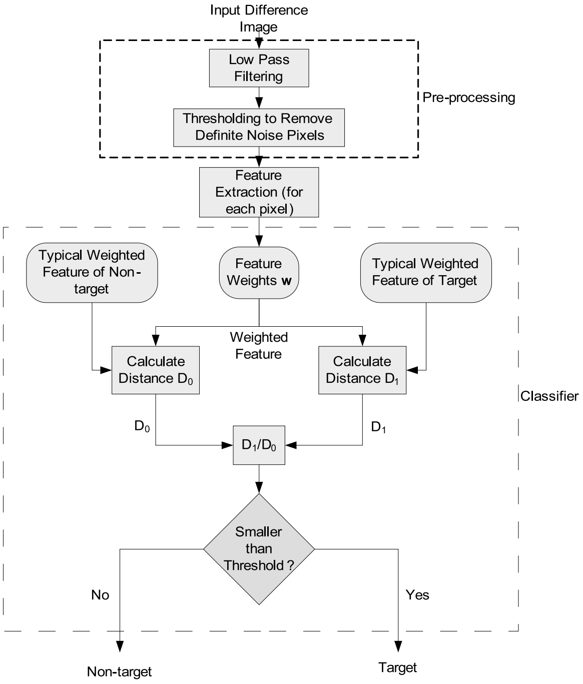


Figure 6: Flow chart of our classification algorithm

In the difference images, target areas should always be brighter than background areas. All the images in this data set are all geometrically registered. In order to reduce the influence of noise on classification, the difference image is processed by a low pass filter. It is equivalent to be convoluted by a core matrix with averaging effect. A convolution core of 5×5 pixels, which is of the same size as that used in the benchmark algorithm [8] is adopted. Also, a prescreening step is performed. It forces those pixels with intensities lower than a given threshold to be 0. This threshold should be enough to preserve all the target pixels. It is intended to remove those pixels that are obviously noise or background. These denoising processes can contribute to boosting the performance of the classifier and increasing the converging speed at the training stage. All the input data is SAR images of 3000 by 2000 pixels. Because there are multiple targets in each image, it is not reasonable to take the whole image as a sample and label it as target or non-target. Instead, for each pixel, a local feature set is extracted and conveyed to the classifier. Then, each pixel is classified as target pixel or non-target pixel. In the baseline algorithm, only the intensity of current pixel is conveyed to the classifier as features. It does not consider the local environment of that pixel and is easy to be fooled by some high-intensity noise. To avoid this problem, for every pixel in the SAR image, all its neighbors within a $2n + 1 \times 2n + 1$ window are considered helpful in representing the behavior of that pixel and used to extend the feature extracted from the current pixel. Their pixel values, Fourier transform coefficients or other transform results can all be used as features. These features include lots of local information in addition to one pixel value and are widely used in image analysis [15]. Here, for a given pixel i, original pixel values of its neighbors are extracted and reordered to form a feature vector $\mathbf{x}_i$. Then, by sliding a window within the image, a set of features X is obtained. Fig. 7 shows the process of feature extraction.

4.2. Feature Weighting

This is actually the supervised learning stage, shown in Figure 5 of the whole algorithm. In this stage, the I-RELIEF feature weighting algorithm searches for a set of feature weights which maximizes the discrimination between the target and non-target groups. Then the corresponding weight vector $\mathbf{w}$, centroid of target group $\bar{\mathbf{x}}_{target}$ and centroid of non-target group $\bar{\mathbf{x}}_{ntarget}$ in weighted space are transfered to the following testing stage.

As mentioned in the previous section, all the pixel values within the neighborhood of pixel under test are extracted to form a feature vector $\mathbf{x}$. These

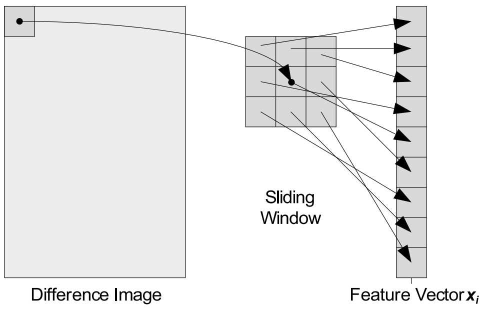


Figure 7: Feature extraction process

features do not have the same contribution to discriminating the target and non-target classes. For example, the feature pixels closer to the pixel under test, i.e., near the center of the sliding window, deserve more attention than those farther away from the pixel under test, i.e., near the border. Otherwise, in a non-target case, the noise within the border area may trigger the classifier to decide that a target is present. In the training stage, some features which are very noisy may prevent the training algorithm from converging or lead to the classifier's over-fitting to the training data. In order to figure out how much attention should be paid to each feature, feature weighting, which assigns a real-valued number to each feature is adopted. The real-valued number here is called the weight of the corresponding feature. For a given feature vector $\mathbf{x}$ and a global weight vector $\mathbf{w}$, a weighted feature vector $\mathbf{x}'$ is obtained by multiplying each feature $x^{(i)}$ with its corresponding weight $w^{(i)}$. Define

$$\mathbf{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(I)} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w^{(1)} \\ w^{(2)} \\ \vdots \\ w^{(I)} \end{bmatrix}, \quad \mathbf{x}' = \begin{bmatrix} x'^{(1)} \\ x'^{(2)} \\ \vdots \\ x'^{(I)} \end{bmatrix}, \tag{11}$$

where $x'^{(i)} = x^{(i)} \times w^{(i)}$, $i = 1, 2, \ldots, I$, where I is the data dimensionality.

We use I-RELIEF or Iterative-RELIEF, an improved interpretation of RELIEF for feature weighting. The key idea of I-RELIEF is to solve a convex optimization problem with a margin-based objective function. For RELIEF, the margin is defined based on a 1-NN, i.e. one nearest neighbor classifier. It only considers one nearest neighbor in the same class and one nearest neighbor in the other class. For I-RELIEF, the margin is averaged between all the sample pairs weighted by the possibility of being an outlier. The feature weights are iteratively estimated according to their ability to discriminate between neighboring patterns. Most algorithms for feature weighting and feature selection, which is the specific case of feature weighting with weights taking values of 0 or 1, rely on heuristic searching, because it is hard to define an objective function that can be optimized with low computational complexity. So they do not guarantee to give optimal solutions. RELIEF [16] addresses this problem by optimizing an objective function with low computational complexity. Let $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_1^N \in \mathbb{R}^I \times \{\pm 1\}$ denotes a training data set, where I is the data dimensionality, $\mathbf{x}_n$ is a sample feature, y_n is the label of the feature, and N is the total number of samples in the training data set. Then, in RELIEF, the feature weighting problem is converted into the following optimization problem:

$$\begin{array}{ll} \max\limits_{\mathbf{w}} & \sum_{n=1}^{N}(\sum_{i=1}^{I} w^{(i)}|x_n^{(i)} - NM^{(i)}(x_n)| - \sum_{i=1}^{I} w^{(i)}|x_n^{(i)} - NH^{(i)}(x_n)|) \\ \text{s.t.} & \|\mathbf{w}\|_2^2 = 1, \mathbf{w} \geq 0 \end{array} \qquad (12)$$

where NM means nearest miss which is the nearest neighbor of x from different class and NH means nearest hit which is the nearest neighbor of x from the same class. This is generated from a natural idea of scaling each feature such that the averaged margin in a weighted feature space is maximized. The constraint $\|\mathbf{w}\|_2^2 = 1$ prevents the maximization from increasing to infinity. The constraint $\mathbf{w} \geq 0$ ensures the weight vector is a distance metric. I-RELIEF [17] is an improved version of RELIEF. It solves two major problems in RELIEF: 1) the nearest neighbors are defined in the original feature space, which may not be the nearest in the weighted feature space; 2) the margin calculation can be influenced by outliers greatly. Based on the assumption that nearest neighbors and identity of a pattern are hidden random variables, I-RELIEF iteratively estimates feature weights following the principle of EM algorithm until convergence. This algorithm is proved to be convergent [17].

To be more specific, for a data sample $\mathbf{x}_n$, two sets $\mathcal{M}_n$ and $\mathcal{H}_n$ are defined as the sets of nearest misses and nearest hits of $\mathbf{x}_n$ respectively, set $\mathcal{S}_n = \{(s_{n1}, s_{n2})\}$ is used to save indices of nearest hit and miss, where $s_{n1} \in \mathcal{M}_n$ and $s_{n2} \in \mathcal{H}_n$. $\mathbf{o} = [o_1, o_2, \ldots, o_N]$ is a binary vector indentifying outliers. $o_n = 0$ if $\mathbf{x}_n$ is an outlier, $o_n = 1$ otherwise. $P_m(i|\mathbf{x}_n, \mathbf{w})$, $P_h(i|\mathbf{x}_n, \mathbf{w})$ and $P_o(o_n = 0|\mathcal{D}, \mathbf{w})$ are defined as the probabilities of the i-th data point being the nearest miss, hit of $\mathbf{x}_n$ and that $\mathbf{x}_n$ being an outlier respectively as following:

$$P_m(i|\mathbf{x}_n, \mathbf{w}) = \frac{f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})}{\sum_{j\in\mathcal{M}_n} f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})} \tag{13}$$

$$P_h(i|\mathbf{x}_n, \mathbf{w}) = \frac{f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})}{\sum_{j\in\mathcal{H}_n} f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})} \tag{14}$$

$$P_o(o_n = 0|\mathcal{D}, \mathbf{w}) = \frac{\sum_{i\in\mathcal{M}_n} f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})}{\sum_{j\in\mathcal{D}\backslash\mathbf{x}_n} f(\|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}})} \tag{15}$$

where $f(\cdot)$ is a kernel function. For example, $f(d) = exp(-d/\sigma)$ where σ is the user defined kernel width.

According to [17], the detailed algorithm can be described as below. For briefness, some notations are defined as $\alpha_{i,n} = P_m(i|\mathbf{x}_n, \mathbf{w}^{(t)})$, $\beta_{i,n} = P_h(i|\mathbf{x}_n, \mathbf{w}^{(t)})$, $\gamma_n = 1 - P_o(o_n = 0|\mathcal{D}, \mathbf{w}^{(t)})$, $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_2 = 1, \mathbf{w} \geq 0\}$, $\mathbf{m}_{n,i} = |\mathbf{x}_n - \mathbf{x}_i|$ if $i \in \mathcal{M}_n$ and $\mathbf{h}_{n,i} = |\mathbf{x}_n - \mathbf{x}_i|$ if $i \in \mathcal{H}_n$.

Step 1: After t-th iteration,the Q function is calculated as:

$$\begin{aligned}
&Q(\mathbf{w}|\mathbf{w}^{(t)}) = E_{\{S,o\}}[C(\mathbf{w})],\\
&= \sum_{n=1}^{N} \gamma_n \Big(\sum_{i\in\mathcal{M}_n} \alpha_{i,n} \|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}} - \sum_{i\in\mathcal{H}_n} \beta_{i,n} \|\mathbf{x}_n - \mathbf{x}_i\|_{\mathbf{w}} \Big),\\
&= \sum_{n=1}^{N} \gamma_n \Big(\sum_j w_j \underbrace{\sum_{i\in\mathcal{M}_n} \alpha_{i,n} m_{n,i}^j}_{\bar{m}_n^j} - \sum_j w_j \underbrace{\sum_{i\in\mathcal{H}_n} \beta_{i,n} h_{n,i}^j}_{\bar{h}_n^j} \Big),\\
&= \mathbf{w}^T \sum_{n=1}^{N} \gamma_n (\bar{\mathbf{m}}_n - \bar{\mathbf{h}}_n) = \mathbf{w}^T \boldsymbol{\nu}.
\end{aligned} \tag{16}$$

Step 2: The re-estimation of $\mathbf{w}$ in the $(t+1)$-th iteration is $\mathbf{w}^{(t+1)} = arg \max_{\mathbf{w}\in\mathcal{W}} Q(\mathbf{w}|\mathbf{w}^{(t)}) = \boldsymbol{\nu}^{+}| \, \|\boldsymbol{\nu}^{+}\|_2$. The above two steps iterate alternatively until convergence, i.e. $\left\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\right\| < \theta$.

4.3. Classification

From the training stage, an optimal feature weight vector $\mathbf{w}^*$ is generated. In the new weighted feature space, the distance between two different classes is maximized and the distance within the same class is minimized. In the weighted space, various distance-based classifier can be designed. In this paper, we choose a distance ratio to design our classifier. Next, we define this distance ratio.

In the training phase, we can obtain the centroid of the target sample set $\bar{\mathbf{x}}_{target} = \frac{1}{N_t}\sum_{i\in CT}\mathbf{x}'_i$ and the centroid of non-target sample set $\bar{\mathbf{x}}_{ntarget} = \frac{1}{N_n}\sum_{i\in CN}\mathbf{x}'_i$, where CT is the set of indices of target samples, CN is the set of indices of non-target samples, N_t is the number of target samples, and N_n is the number of non-target samples. Then, for a feature sample (under test) $\mathbf{x}$, the distance ratio is defined by

$$DR(\mathbf{x}) = \frac{|\mathbf{x} - \bar{\mathbf{x}}_{ntarget}|}{|\mathbf{x} - \bar{\mathbf{x}}_{target}|} \tag{17}$$

Given an input SAR image under test, a distance ratio is calculated for each pixel. Then, the distance ratio is compared to a threshold and a decision is made for each pixel, based on the following criterion:

$$\begin{aligned} DR(\mathbf{x}) > \lambda &\;\;\rightarrow\;\; \text{decide change/target, labeled as `1'} \\ DR(\mathbf{x}) \leq \lambda &\;\;\rightarrow\;\; \text{decide no change/non-target, labeled as `0'} \end{aligned} \tag{18}$$

The above process produces a binary-valued image with ‘1’ denoting target and ‘0’ denoting non-target. This image is also called change mask. For the same input SAR image under test, different reference images may result in different change masks. With multiple change masks, we can apply a majority voting rule to each pixel and obtain a final change mask, as shown in Figure 8. Note that for a different reference image, we need to re-do the training of our classifier since a different reference image represents a different training sample set.

4.4. Post Processing

The output produced by our classifier is a binary valued image. But the objective of target detection is to obtain the locations of targets. To achieve this, we group connected pixels whose value is “1”, and declare such a connected region as a target. The coordinates of the centroid of a connected region represents the location of the target associated with the region.

Another purpose of post processing is to remove false alarms. Since we have prior knowledge about the size of the targets, we can remove the connected regions whose sizes are smaller than an expected value. Our experimental results in the next section show the effectiveness of such post processing.

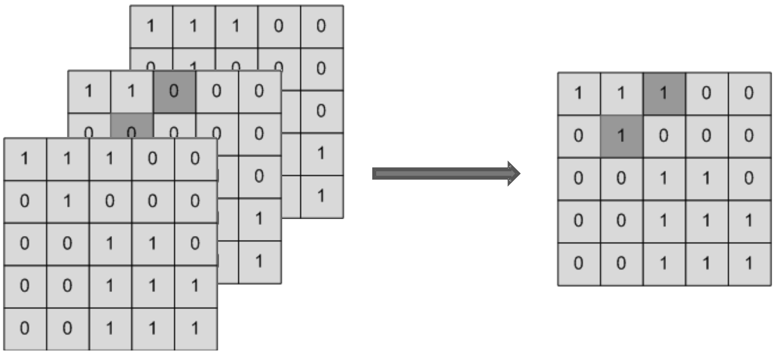


Figure 8: Majority voting of change masks.

5. Experimental Results

In the experiments, for any given input image, the reference images are always chosen to be those images taken under the same flight angle but different deployments. The requirement of same flight angle ensures that the imaging conditions are the same. According to the data set, the four deployments Sigismund, Karl, Fredrik and Adolf-Fredrik are denoted as mission 2, 3, 4 and 5 respectively. So, for each image, there can be no more than three different reference images from other deployments.

Processing step	Parameter	Value
Preprocessing	Averaging kernel size	5 × 5 pixels
	Denoising threshold	0.25
Feature extraction	Sliding window size	19 × 19 pixels
I-RELIEF feature weighting	Maximum number of iterations	500
	Distance metric	'Euclidean'
	Kernel function	$f(d) = \exp(-d/\sigma)$
	Kernel width σ	25
Classification	Threshold λ on DR	1/3
Post processing	Minimum number of connected pixels as a target	35 pixels
Evaluation	Distance threshold	10 pixels

Table 1: Parameters used in the experiments

Table 1 shows the parameters used in our experiments. Our rationale of choosing 19 × 19 for the sliding window size is the following. For best performance, the sliding window should be large enough to cover a whole target. On the other hand, a larger sliding window means higher computational complexity. We tested sliding window sizes of 7 × 7, 19 × 19, 31 × 31 and 41 × 41 pixels. The later three achieve the same performance while the first one performs worse. Hence, we choose 19 × 19 for the sliding window size. The setup of the parameters for I-RELIEF feature weighting is simply a maximum iteration constrain and a definition of kernel specified by its distance metric, kernel function and kernel width. The Euclidean distance metric and kernel function of $f(d) = exp(-d/\sigma)$ defines one of the most common kernel structure. The kernel width σ is a key parameter controlling the trade off between the variance and bias of the estimation results. Larger σ leads to smaller variance and larger bias and vice versa. Since smaller number of samples leads to larger variance of the estimation and we do not have many training samples, we choose a relative large kernel width of 25 to trade some of the bias for smaller variance of our estimation. If more training samples are given, this values should be reduced to increase the accuracy of the scheme. The choice of threshold λ depends on how much emphasis we have on the correct detection rate and the false alarm rate, because larger value of λ leads to smaller false alarm rate as well as smaller detection rate. We set $\lambda = 1/3$ to adjust our detection rate to the same level as that given by the benchmark algorithm, so that we can compare the false alarm rate with the benchmark algorithm at the condition of same detection rate. The value of minimum number of connected pixels as a target is estimated according to the size of the real target we are detecting, e.g vehicles or tanks, height of the flight when the data was collected, resolution and diffusion effect of the radar used. All these information should be given by the scenario where our scheme is applied. The distance threshold is set to 10 pixels, the same as that used in the benchmark algorithm, to ensure the fairness of the comparison.

When we reshape all the feature and weight vectors into form of matrix with the same size of the sliding window, we can see in Figure 9 that the shape of the major part of the target is automatically extracted as the weight matrix. So, as long as the sliding window is large enough to hold the whole target, the weight matrix could be trained to match the target itself and the performance of the scheme will not decay with the increasing size of the window. This explains the phenomenon stated above that larger window size does not improve the performance. Of course, the performance will be influenced if the sliding window is so large that more than one target are within its range.

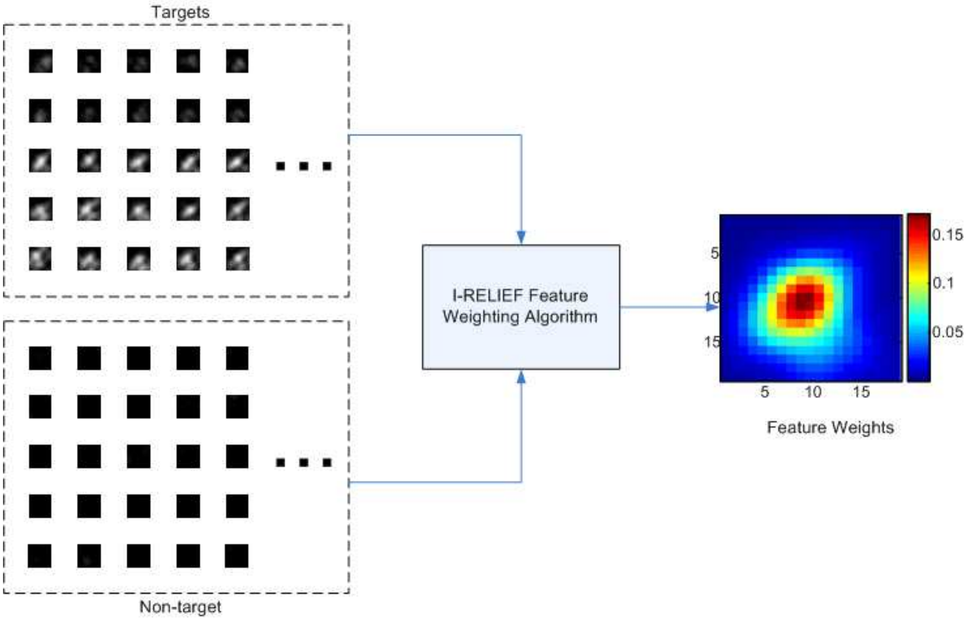
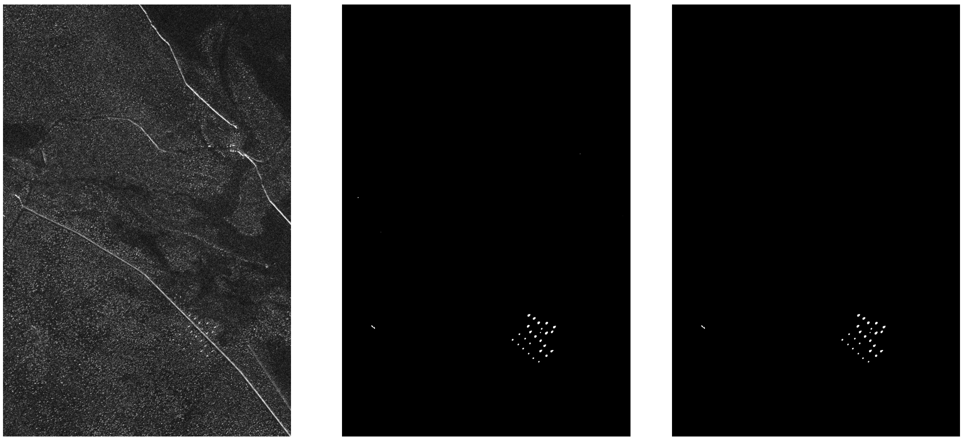


Figure 9: Feature and weight vectors in matrix form.

In the training stage, the image of mission 3 flight 5 is used as training samples. The coordinates of the 25 target vehicles are included in the data set. For each target, a sliding window is manually shifted within a 5×5 neighborhood of the given coordinates of the target, and one feature vector belongs to the target class is extracted from the sliding window at each position. A feature vector set labeled as target including totally 625 samples is extracted from the 25 target locations. Another feature vector set labeled as non-target is extracted from 625 background positions which are manually chosen to avoid the target region. These two label sample sets are used as the input of the I-RELIEF feature weighting algorithm to give an output of weight vector $\mathbf{w}^*$. Values of $\bar{\mathbf{x}}_{target}$ and $\bar{\mathbf{x}}_{ntarget}$ are also estimated from the training sample sets. According to the above reference look-up matrices, three different references are chosen, and training and testing are performed independently for each of the three references. Finally, a majority vote is applied to the three change masks obtained from the training and testing w.r.t. the three references, and the algorithm outputs the coordinates of all the detected targets.

For performance evaluation, the output coordinates are compared to the ground-truth target positions. If the location of a target detected by the ATD algorithm, is within the disk of 10-pixel (i.e., 10-meters) radius, centered at the ground truth position, then we declare that the detection is correct. If more than one target is found within this disk, one will be counted as a correct detection while others will be counted as false alarms.

Table 2 shows the testing results of our proposed scheme with majority vote from 3 references compared with the benchmark algorithm. Figure 10 shows an example of the input and output images of our proposed scheme with 3 references, where Figure 10a is the raw input image from mission 4 pass 2 in the data set, Figure 10b is the output binary mask before post processing and Figure 10c is the output binary mask after post processing. If you zoom in on the masks, you should be able to find that several small dot areas were removed by the post processing. Table 3 shows the testing results of our proposed scheme with a single reference image compared with the benchmark algorithm. In the testing stage, the image of mission 3 flight 5 is not used for testing because it serves as training samples. The experimental results show that compared to the benchmark algorithm [8], our proposed scheme produces much fewer false alarms, i.e., 11 for ours vs. 86 for the



(a) Input data, mission 4 pass 2.

(b) Result before post processing.

(c) Result after post processing.

Figure 10: An example of the input and outputs, with and without post processing of our proposed scheme with 3 references.

Image under test		Correct detections		False alarms	
Mission	Pass	Benchmark Algorithm	This Scheme	Benchmark Algorithm	This Scheme
2	1	25	25	2	0
3	1	22	24	1	3
4	1	25	25	2	0
5	1	23	25	4	2
2	2	25	25	2	0
3	2	25	25	4	1
4	2	25	25	3	2
5	2	25	21	4	0
2	3	25	25	3	1
3	3	23	23	4	0
4	3	25	25	0	2
5	3	24	25	2	0
2	4	24	25	3	0
3	4	25	25	2	0
4	4	25	25	4	0
5	4	25	22	4	0
2	5	25	25	3	0
3	**5**	**17**	**19**	**10**	**13**
4	5	25	25	2	0
5	5	23	25	29	0
2	6	25	25	1	0
3	6	25	24	3	0
4	6	25	25	1	0
5	6	23	25	3	0
Total		579	583	96	24
Miss.3 pass5 Removed		562	564	86	11

Table 2: Comparing of results from benchmark algorithm and that from proposed scheme with 3 different reference images.

Image under test		Correct detections		False alarms	
Mission	Pass	Benchmark Algorithm	This Scheme	Benchmark Algorithm	This Scheme
2	1	25	25	2	0
3	1	22	23	1	3
4	1	25	25	2	0
5	1	23	25	4	3
2	2	25	25	2	0
3	2	25	25	4	1
4	2	25	25	3	1
5	2	25	21	4	0
2	3	25	25	3	1
3	3	23	23	4	0
4	3	25	25	0	1
5	3	24	25	2	0
2	4	24	25	3	0
3	4	25	25	2	0
4	4	25	25	4	0
5	4	25	23	4	1
2	5	25	25	3	0
3	**5**	**17**	**18**	**10**	**18**
4	5	25	25	2	0
5	5	23	23	29	10
2	6	25	25	1	0
3	6	25	24	3	0
4	6	25	25	1	0
5	6	23	25	3	0
Total		579	580	96	39
Miss.3 pass5 Removed		562	562	86	21

Table 3: Comparing of results from benchmark algorithm and that from proposed scheme with exact the same single reference image.

benchmark, while yet achieving better target detection performance, i.e., 564 for ours vs. 562 for the benchmark.

The proposed scheme uses more than one reference images which means more information than what the benchmark algorithm needs. Can the scheme still give better performance without the help of extra information? The comparing of results from benchmark algorithm and that from the proposed scheme with exact the same single reference image is also shown in Table 3. The reference image is picked up according to matrix ref_m which is built based on the correspondence in the benchmark algorithm in [8]. It shows that without extra reference images, the proposed algorithm still provides much better performance than the benchmark algorithm does.

Remark 1. *Our scheme achieves better performance than that of the benchmark algorithm because of the following three reasons. First of all, instead of assuming the intensity values from testing and reference images as independent random variables as assumed in [8] and [18], we do direct image differencing to suppress the background clutters and noise. Since all the images in the data set are registered, image differencing could achieve better suppression performance without assuming the spatial independence of pixel values. The spatial correlation is automatically alleviated. Secondly, we use the maximum-margin classifier to boost the performance with given samples by doing the classification in a transformed feature space with larger distinction between the two categories. In the end, we use diversity combining by repeating the detection procedure with different reference images and combining the results. This means we explore more information contained in the data set which is helpful for our detection task.*

6. Conclusion and Future Work

In this paper, we looked into a target detection problem for wide area ground surveillance through SAR. To solve this problem, we proposed a new scheme based on local feature extraction and supervised learning through feature weighting. Instead of making assumptions of statistic models like conventional benchmark algorithm did, this scheme extracts prior information through learning, which made it more flexible to be fitted in different scenarios. On the other hand, this scheme uses local features instead of single pixels to make decisions, so it should be more robust to single-pixel noise. Besides, in the training stage, the shape factor of the targets was automatically included into the weight vector, which means the probability of giving false alarms on different shaped potential detections was reduced. Compared with the benchmark algorithm stated in [8], this scheme produced much fewer false alarms, 11 compared to 86 at the same correct detection level, 564 compared to 562.

There is still room to improve our scheme. For example, we can use wavelet coefficients or scale invariant features (e.g., obtained by scale-invariant feature transform (SIFT)) instead of image intensities as the input and reconsider our feature weighting algorithm in a kernel space. As a result, we will get a generalized feature weighting algorithm on sophisticated features. To make better usage of the inter-class and inner-class margins, we can try out some more advanced classifiers such as support vector machine. It is also possible to improve the preprocessing step with the latest denoising techniques, e.g., non-local means algorithm. Furthermore, we will try to find more data sets on which our scheme could be evaluated.

Disclaimers

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

Acknowledgement

This material is based on research sponsored by AFRL under agreement number FA8650-06-1-1027. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] B. Binder, M. Toups, S. Ayasli, E. Adams, SAR foliage penetration phenomenology of tropical rain forest and northern US forest, Proceedings of IEEE International Radar Conference, 1995, Alexandria, VA (8-11 May 1995) 158–163.

[2] L. Bessette, S. Ayasli, Ultra-wideband P-3 and CARABAS II foliage attenuation and backscatter analysis, Proceedings of 2001 IEEE Radar Conference, Atlanta, GA (1-3 May 2001) 357–362.

[3] Y. Sun, Z. Liu, S. Todorovic, J. Li, Adaptive boosting for synthetic aperture radar automatic target recognition, IEEE Transactions on Aerospace and Electronic Systems 43 (1) (January 2007) 112–125.

[4] L. Kaplan, Improved SAR target detection via extended fractal features, IEEE Transactions on Aerospace and Electronic Systems 37 (2) (April 2001) 436–451.

[5] D. Howard, S. Roberts, R. Brankin, Target detection in SAR imagery by genetic programming, Advances in Engineering Software 30 (5) (May 1999) 303–311.

[6] T. Cooke, N. Redding, J. Schroeder, J. Zhang, Comparison of selected features for target detection in synthetic aperture radar imagery, Conference record of the 33rd Asilomar Conference on Signals, Systems, and Computers 2 (1999) 859–863.

[7] S. Kuttikkad, R. Chellappa, Non-Gaussian CFAR techniques for target detection in high resolution SAR images, Proceedings of IEEE International Conference on Image Processing 1 (1994) 910–914.

[8] M. Lundberg, L. Ulander, W. Pierson, A. Gustavsson, A challenge problem for detection of targets in foliage, Proceedings of SPIE.

[9] L. Ulander, P. Frolind, A. Gustavsson, H. Hellsten, B. Larsson, Detection of concealed ground targets in CARABAS SAR images using change detection, Proceedings of SPIE - International Society Optical Engineering 3721 (1999) 243–252.

[10] L. Ulander, B. Flood, P. Follo, P. Frolind, A. Gustavsson, T. Jonsson, B. Larsson, M. Lundberg, W. Pierson, G. Stenstrom, Flight Campaign Vidsel 2002, CARABAS-II change detection analysis (FOI-R-1001-SE).

[11] L. Ulander, W. Pierson, M. Lundberg, A. Gustavsson, Performance of VHF-band SAR change detection for wide-area surveillance of concealed ground targets, Proceedings of SPIE - International Society Optical Engineering 5427 (2004) 259–270.

[12] L. Ulander, W. Pierson, M. Lundberg, P. Follo, P. Frolind, A. Gustavsson, CARABAS-II SAR change detection performance on ground targets concealed in foliage, Proceedings of EUSAR 2004, 5th European

Conference on Synthetic Aperture Radar, Ulm, GE (25-27 May 2004) 297–300.

[13] https://www.sdms.afrl.af.mil/datasets/vhf_change_detection/index.php.

[14] L. Ulander, M. Lundberg, W. Pierson, A. Gustavsson, Change detection for low-frequency SAR ground surveillance, IEE Proceedings of Radar Sonar Navigation 152 (6) (December 2005) 413–420.

[15] W. Hong, Hybrid models for representation of imagery data, Ph.D. thesis, University of Illinois at Urbana-Champaign (2006).

[16] K. Kira, L. Rendell, A practical approach to feature selection, Proceedings of the 9th International Workshop on Machine Learning (1992) 249–256.

[17] Y. Sun, J. Li, Iterative RELIEF for feature weighting, Proceedings of the 23rd ACM International Conference on Machine learning 148 (2006) 913–920.

[18] Z. Wei, G. Jian, W. Jie, Change detection of concealed targets using repeat-pass SAR images, Asian and Pacific Conference on Synthetic Aperture Radar (2007) 275–278.