Progress In Electromagnetics Research C, Vol. 31, 271–285, 2012

TBD ALGORITHM BASED ON IMPROVED RANDOMIZED HOUGH TRANSFORM FOR DIM TARGET DETECTION

L. Fan[1, 2, *], X. Zhang[1], and L. Wei[1]

[1]School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

[2]School of Physics and Electronic Engineering, Leshan Normal University, Leshan 614004, China

Abstract—The track-before-detect (TBD) methodologies jointly process more consecutive scans and show superior detection performance for the low signal-to-noise ratio (SNR) targets over the conventional methods. A TBD algorithm based on improved Randomized Hough Transform for dim target detection is proposed in this paper. This algorithm uses the sequence numbers of scans to make sure that the point pairs are selected from different scans, avoiding the unreasonable situation that the point pairs may be selected from the same scan in the traditional Randomized Hough Transform (RHT). Second, it introduces a new voting method. Based on the minimum Euclidean distance criterion, this voting method finds the optimal parameter cell to vote, making the voting result better than the traditional RHT. In addition, we not only increase score of the optimal parameter cell but also update the corresponding parameter, thus suppressing the deviation between the recovered track and the target's track. Simulation results demonstrate the proposed algorithm can detect the dim target more rapidly and accurately than traditional RHT, especially under the background of low SNR.

1. INTRODUCTION

Early detection and trajectory estimation of moving targets from remote surveillance radars is a very challenging problem [1–11]. The detection and tracking strategy should be power efficient to deal with the low signal-to-noise ratio (SNR) targets, whereas the

Received 23 May 2012, Accepted 4 August 2012, Scheduled 6 August 2012

* Corresponding author: Ling Fan (lingf@uestc.edu.cn).

complexity should not hamper the process of early decisions. This specification is well met by the track-before-detect (TBD) approaches. TBD procedures allow simultaneous detection and tracking and show superior detection performance over the conventional methods. Many TBD algorithms have been proposed, such as the Hough Transform (HT)-based TBD [12–14], the Dynamic Programming (DP)-based TBD [15–18] and the Particle Filter (PF)-based TBD [19, 20].

The HT is a feature detector, often used in image processing [21, 22]. In view of the long computation time and large memory requirements of the HT, Kultanen et al. [23] developed the Randomized Hough Transform (RHT) to complete the task of extracting global features such as line segments from binary images. The algorithm overcomes most problems associated with the standard HT, including speed and memory consumption. However, the traditional RHT has some disadvantages when used to tackle the TBD problem for radar system. First, the traditional RHT-based TBD (RHT TBD) extracts the target's track in a data plane which is overlaid by multi-scan radar echo data. Because the scans' sequence numbers are lost, RHT TBD may select point pairs (samples or possible tracks) from the same-scan radar echo data and initiate false target's tracks. Second, during the process of voting, the RHT TBD doesn't traverse the Dynamic-Link List (DLL) to find the best parameter cell to vote, leading to non-optimal results. Third, the RHT TBD only increases score of the parameter cell, but does not update the corresponding parameter. Thus, there will be a big deviation between the recovered track and the target's track when the parameter in DLL has a big deviation with all the parameters which vote for it.

Considering the above problems of the traditional RHT TBD, a TBD algorithm based on improved Randomized Hough Transform for dim target detection is proposed in this paper. This algorithm offers three improvements on the traditional RHT TBD: (1) Using the scans' sequence numbers to make sure that the point pairs are selected randomly from different scans of radar echo data. Thus, it can assure that the detected track is formed by the radar echo data which come from different scanning moments. In this way, the ratio of valid samples (samples which come from target's points) increases greatly and the number of false target's tracks reduces to a great extent. (2) This algorithm also introduces a new voting method: first, we traverse the DLL to find out a collection, where the voting parameter falls into the given tolerance of each element of the collection. Then based on the minimum Euclidean distance criterion, we get the optimal parameter cell from the collection to vote. Thus the voting result is much better than the traditional RHT TBD. (3) At the same time as voting, the parameter in the optimal parameter cell is also updated to suppress the deviation between the detected track and the target's track. In addition, during the process of voting, each polling point pair's positions are recorded in DLL to restore the specific location of the target, which is beneficial for further tracking.

This paper includes five sections. Section 2 introduces the target and measurement model. Section 3 explains the traditional RHT and the proposed algorithm is presented in Section 4. Section 5 contains simulation examples demonstrating the concept. Section 6 contains a conclusion.

2. TARGET AND MEASUREMENT MODEL

2.1. Target Motion Model

The problem is depicted as a dim target with constant velocity moving in x-y plane. The target motion model is expressed as follows:

$$S_{k+1} = FS_k + V_k \tag{1}$$

S_k is the target state in the kth scan. F is the state-transition matrix. V_k is zero mean Gaussian noise which describes the target's slow maneuver in the kth scan. Where

$$S_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^T \tag{2}$$

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where x_k ($\dot{x}_k$) and y_k ($\dot{y}_k$) denote target's position (velocity) in x and y directions respectively in the kth scan. T is the time interval between the two adjacent scans.

2.2. Measurement Model

The measurements are the reflected power on x-y plane,

$$z_{i,j}^k = \begin{cases} n_{i,j}^k & \text{no target in cell } (i,j) \text{ at } k\text{th scan} \\ A_k + n_{i,j}^k & \text{have target in cell } (i,j) \text{ at } k\text{th scan} \end{cases} \tag{4}$$

where $1 \leq i \leq M$, $1 \leq j \leq N$, $1 \leq k \leq K$. M and N are the total number of the resolution cells in x axis and y axis respectively. K is the total number of the scans. A_k is the amplitude of target's echo in the kth scan, $n_{i,j}^k$ is an exponential distribution noise with parameter is σ^2.

3. RANDOMIZED HOUGH TRANSFORM

Since this paper considers a trajectory of a moving target is a straight line, here, we introduce RHT for a straight line, only. More detailed explanation of RHT see, e.g., [23, 24]. Let D denote the set of all points in an original binary image. Let (x, y) is the coordinates in the original image and (ρ, θ) is the two parameters of the line. We randomly sample two points $d_i = (x_i, y_i)$, $d_j = (x_j, y_j)$, $d_i \neq d_j$ out of the set D. By solving the following joint equations,

$$\begin{cases} x_i \cos\theta + y_i \sin\theta = \rho \\ x_j \cos\theta + y_j \sin\theta = \rho \end{cases} \tag{5}$$

we can get a parameter pair (ρ, θ). The Dynamic-Link List (DLL) is created in which each cell has both a parameter pair and an integer value called the score. When a pair (ρ, θ) is obtained, we search and check whether there is a cell in DLL with the same parameter pair. If there is such a cell, then we vote for this parameter cell, i.e., increase its score by one. If none is found, then we create a new cell with parameters equal to (ρ, θ) and score equal to one and insert it into DLL as a new cell. After a certain number of random sampling, it is not difficult to see that the corresponding accumulator cells are incremented, if the image space contains a line. As a result, finding out a cell in DLL which has maximum score and comparing it with a threshold, we can detect a line.

4. IMPROVED RANDOMIZED HOUGH TRANSFORM

In this section, three shortcomings of the RHT are introduced firstly, and the corresponding improved strategies are proposed respectively.

4.1. Select Point Pairs from Different Scans

The traditional RHT is a batch processing algorithm, i.e., the measurements of multi-scan are superposed in a 2-dimensional data plane (x-y plane), and lose their sequence numbers. In consequence, the point pairs may be selected from the same scan in selecting process for the traditional RHT. However, there is at most a single measurement received from a target in each scan. Therefore, it is invalid if the point pairs from the same scan. In order to overcome this invalid selecting, the measurements' sequence numbers are retained and the data space is expanded to three-dimensional $(x - y - k)$, as illustrated in Figure 1. The red dot indicates the targets' measurements and the red straight line connecting the dots indicates the targets'

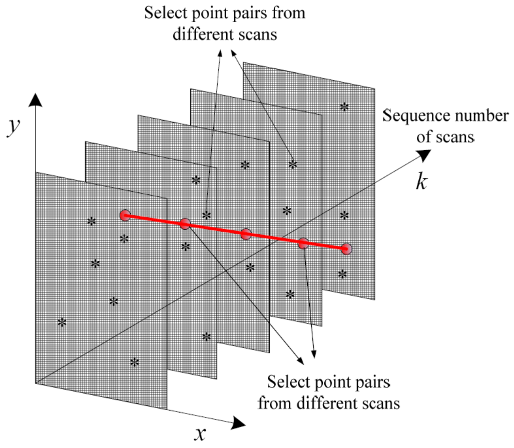

Figure 1. Selecting point pairs in three-dimensional data space.

trajectory, and the asterisk indicates the measurements caused by noise, in Figure 1. We can ensure that the point pairs are always selected from different scans in such three-dimensional data space.

The steps of selecting point pairs and calculating the corresponding parameters are proposed as follows:

Step one, select a point pair $\left(z_{i_1,j_1}^{k_1}, z_{i_2,j_2}^{k_2}\right)$ from any two different scans randomly, where, $k_1 \neq k_2$.

Step two, calculate the corresponding velocity of $z_{i_1,j_1}^{k_1}$ and $z_{i_2,j_2}^{k_2}$ by the following equation:

$$v = R/(|k_1 - k_2| \cdot T) \tag{6}$$

where $R = \left|z_{i_1,j_1}^{k_1} - z_{i_2,j_2}^{k_2}\right|$ is the distance between $z_{i_1,j_1}^{k_1}$ and $z_{i_2,j_2}^{k_2}$. T is the time interval between the two adjacent scans.

Step three, according to the priori velocity information $(V_{\min}, V_{\max})$ of the target, we can further eliminate the invalid samples by the following equations:

$$V \leq V_{\min} \quad \text{or} \quad V \geq V_{\max} \tag{7}$$

$$V_{\min} \leq V \leq V_{\max} \tag{8}$$

If v satisfies (7), $z_{i_1,j_1}^{k_1}$ and $z_{i_2,j_2}^{k_2}$ are invalid and discarded. Then go back to step one, re-elect another point pair.

If v satisfies (8), calculate the corresponding parameter (ρ, θ) of

$z_{i_1,j_1}^{k_1}$ and $z_{i_2,j_2}^{k_2}$ by the following equations:

$$\theta = a\tan(-(i_1 - i_2)/(j_1 - j_2)) \tag{9}$$

$$\rho = i_1 \cos(\theta) + j_1 \sin(\theta) \tag{10}$$

where ρ is the length of the normal vector from the origin to the line, and θ is the angle measured counterclockwise from $+x$ axis to the normal.

So far, we have selected a valid sample and calculated the corresponding parameter (ρ, θ). Then we should decide whether to make (ρ, θ) vote for the parameter cell of DLL or just put it in the end of DLL. Following, a new voting method is introduced in Section 3.2.

4.2. A New Voting Method

Firstly, we review the problem of the traditional RHT during the voting process. Let $P_c = \{\rho_m, \theta_m\}$ denotes the set of all existed parameter cells in DLL. If $P_c = \emptyset$, put (ρ, θ) in the end of DLL, set its corresponding poll to 1 and record the positions of the voting point pair in DLL. If $P_c \neq \emptyset$, decide (ρ, θ) to vote for the existed parameter cell in DLL or just put in the end of DLL according to the following criterion:

$$|\rho_m - \rho| \leq \Delta\rho \quad \text{and} \quad |\theta_m - \theta| \leq \Delta\theta \tag{11}$$

where $\Delta\rho$ and $\Delta\theta$ are the given tolerances of polar distance and polar angle respectively. If (ρ, θ) falls into the given tolerance of multi parameter cells, (ρ, θ) will vote for the multi parameter cells. Figure 2 illustrates this situation. (ρ_i, θ_i), (ρ_j, θ_j) and (ρ_k, θ_k) are the existed parameter cells in DLL and (ρ, θ) is a new parameter which would be decided to vote for the existed parameter cell in DLL or just put in the end of DLL. The ellipse centered in the each existed parameter cell indicates the range of given tolerance and the diagonally shaded area indicates the public area of three parameter cells (ρ_i, θ_i), (ρ_j, θ_j) and (ρ_k, θ_k). Due to (ρ, θ) locating at this public area, it will vote for three parameter cells. This voting result is not optimal and will lead to some false tracks, obviously.

To avoid the defect of the traditional RHT in the voting process, the minimum Euclidean distance criterion is used to optimize the voting process. If multi parameter cells (ρ_m, θ_m), $m > 1$ in DLL satisfy the criterion (11) simultaneously, i.e., (ρ, θ) locates at the diagonally shaded area in Figure 2, we will find the parameter cell which is minimum distance with (ρ, θ) to vote. The distance between (ρ, θ) and (ρ_m, θ_m) is

$$dis\,(\rho_m, \theta_m) = \sqrt{(\rho_m - \rho)^2 + (\theta_m - \theta)^2} \tag{12}$$

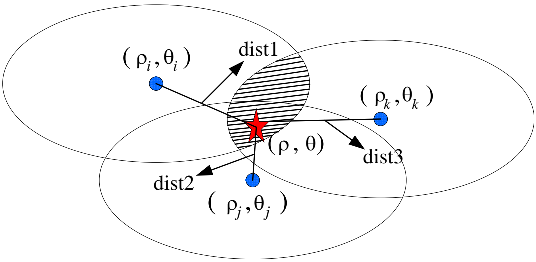

Figure 2. Illustrating the voting process.

According to the minimum Euclidean distance criterion, the goal is to find the parameter cell (ρ_m, θ_m) that minimize dis, i.e.,

$$\left(\hat{\rho}, \hat{\theta}\right) = \min_{\rho_m, \theta_m} dis(\rho_m, \theta_m) \tag{13}$$

Then, vote for $(\hat{\rho}, \hat{\theta})$ and record the positions of the point pair.

4.3. Update Parameter in DLL's

If a cell in DLL satisfies (11), this cell is voted and its score is increased by one, but the original point pairs are not recorded, in the traditional RHT. However, the voted cell and (ρ, θ) may be have a large deviation even if there meet (11). In addition, the original point pairs must be recorded to recover the targets' trajectory. Therefore, an iterative mean method is used to update the parameter cell as follows.

$$(\rho_m', \theta_m') = ((\rho_m, \theta_m) + (\rho, \theta))/2 \tag{14}$$

In this way, the parameter cell (ρ_m, θ_m) is adjusted its error according to the parameters which vote for, continuously, and the deviation will be suppressed.

5. EXPERIMENT RESULTS AND ANALYSIS

5.1. Simulation Results

In the radar system, the measurement is usually finished in polar coordinate. However, the data processing is commonly done in Cartesian coordinate. To evaluate the performance of the proposed algorithm, a point target with constant velocity and complex white Gaussian noises are simulated in the $x - y - k$ data space.

The signal-to-noise ratio (SNR) for each scan of radar echo data is defined as follows:

$$(SNR)_k = 10 \log(A_k^2/\sigma^2) \tag{15}$$

where the unknown target amplitude A_k at each scan is randomly drawn from an Rayleigh distribution with an average return amplitude $\bar{A}$.

According to the definition of (14), different scans may have different SNR. To evaluate the performance of the proposed method more rationally, we give a definition to the average SNR for multi-scan radar echo data, as follows:

$$SNR = 10\log\left(\bar{A}^2/\sigma^2\right) \tag{16}$$

System parameters and algorithm parameters are shown in Table 1 and Table 2 respectively.

Table 1. System parameters.

Parameters	Value
Scan Interval T	3 s
Unit number of data space (x, y, k)	$(100, 100, 10)$
Resolution $(\Delta x, \Delta y)$	(0.3 km, 0.3 km)
Target's Initial State $(x_0, \dot{x}_0, y_0, \dot{y}_0)$	(30 km, −0.9 Mach, 27 km, −0.9 Mach)

Table 2. Algorithm parameters.

Parameters	Value
False-alarm Probability P_{fa}	0.01
Given Resolution $(\Delta\rho, \Delta\theta)$	(0.3 km, 2°)
Priori Information $(V_{\min}, V_{\max})$	(0.2 Mach, 2 Mach)
Maximum Sampling Number S_{sample}	$n \times C_K^2$

Where P_{fa} is the false-alarm probability of the first threshold. The priori velocity information $V_{\min}$ and $V_{\max}$ are the allowed minimum and maximum velocity, respectively, and they are set to a larger range to consider the possible velocity of the target. The maximum sampling number is $S_{sample} = n \times C_K^2$, where n is the total number of the data which surpass the first threshold in the data space. K is the total number of scans. C_K^2 is the number of combinations of K things taken 2 at a time.

Simulate data according to the above system parameters under two different backgrounds. One is $SNR = 8\,\text{dB}$. The other is $SNR = 10\,\text{dB}$. Then use the traditional RHT and the proposed algorithm to process the simulated data. After the completion of sampling and voting, we find the global maximum poll in the DLL, giving the corresponding parameter $(\rho'\theta')$. Then restore the target's track according to the parameter $(\rho'\theta')$.

When $SNR = 8\,\mathrm{dB}$, the results of these two algorithms are shown in Figure 3 and Figure 4 respectively. When $SNR = 10\,\mathrm{dB}$, the results are shown in Figure 5 and Figure 6 respectively. To demonstrate the shortcomings of the traditional RHT and the excellence of the proposed algorithm vividly, we show the simulated data, the target's real locations, the detected track in the upper half of each image and show the target's real locations, the detected target's track, the polling point pairs' locations in the lower half of each image.

In Figures 3–6, circles express the target's real locations and solid lines are the detected tracks. In the upper halves of these four figures, dots express noises. While in the lower halves, dots are the locations of the point pairs which vote for the detected tracks.

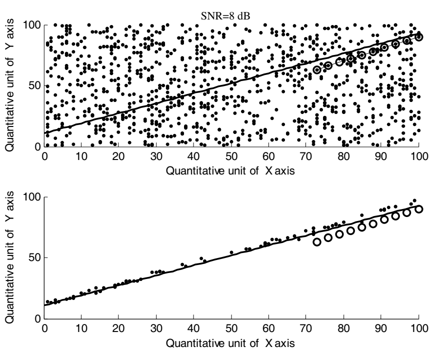

Figure 3. Result of the traditional RHT.

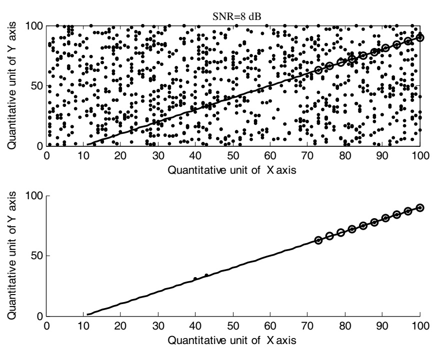

Figure 4. Result of the propoaed algorithm.

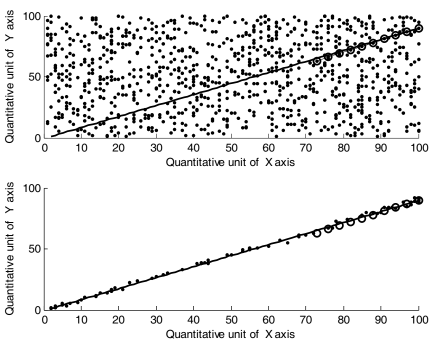

Figure 5. Result of the traditional RHT.

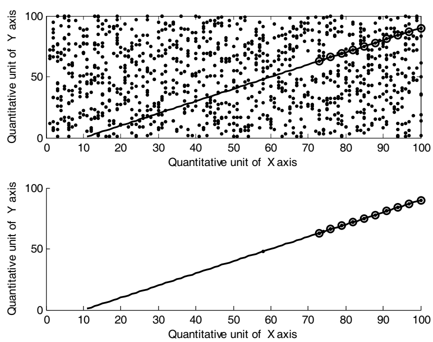

Figure 6. Result of the proposed algorithm.

As Figure 3 shows, under the background of $SNR = 8\,\mathrm{dB}$, there is a big deviation between the detected track and the target's track, that is, the traditional RHT does not detect the target's track accurately. Under the background of $SNR = 10\,\mathrm{dB}$, the traditional RHT basically recovers the target's track correctly. However, the track recovered by the traditional RHT contains a lot of false-alarm points. So it can not give the specific locations of the target and this does not advance to further detection and tracking. And the total processing time is 20.25 seconds under the CPU Intel P4 2.4G environment.

As Figure 4 and Figure 6 show, under the background of $SNR = 8\,\mathrm{dB}$ and $SNR = 10\,\mathrm{dB}$, the proposed algorithm in this paper can restore the target's track accurately due to its improvements upon the traditional RHT. In addition, the target's specific locations can be also recovered accurately, thus it can provide a great help for further detection and tracking. And the total processing time is just 1.313 seconds under the CPU Intel P4 2.4G environment.

We also implement the Monte Carlo simulation to quantitatively describe the performance of the proposed algorithm. The total number of scans is 11. The probability of detection (p_d), the algorithm errors and the processing time are given in Figure 7 and Figure 8, where the algorithm errors between the detected track and the target's track are given in the form of mean errors of the track's parameters (rho and theta, where rho is the length of the normal from the origin to the track and theta is the angle measured counterclockwise from $+x$ axis to the normal) respectively under different SNR, as shown in Figure 8.

As Figure 8 shows, the mean errors of rho and theta are small and decrease with the improvement of SNR. The mean time of track extraction is not more than 1.4 s. This illustrates that the proposed algorithm can detect the target's track relatively accurately and its performance will be better with the increase of the SNR.

5.2. Experiment Results

In order to further validate the effectiveness of the proposed method in this paper, the measured data which has been operated by the constant false alarm rate (CFAR) is processed by the traditional RHT and the proposed algorithm.

Here, to improve the real-time property of the algorithms, we use sliding window to process the 40 scans measured data. The experiment parameters are shown in Table 3.

The maximum sampling number is $S_{sample} = n \times C_K^2$, where n is the total number of non-zero data in the data space after CFAR processing. K is the length of the sliding window. Here, we set $K = 3$.

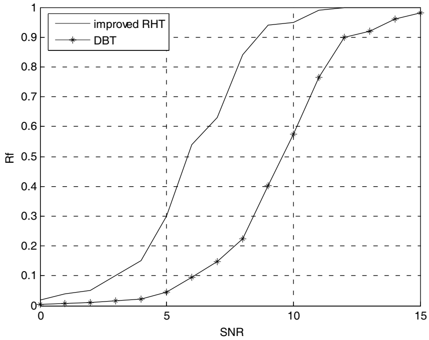

Figure 7. Comparison between the optimal DBT and the proposed algorithm.

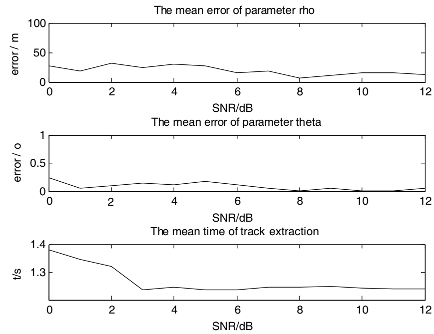

Figure 8. Result of quantitative analysis.

Table 3. Experiment parameters.

Parameters	Value
Priori Information ($V_{\min}, V_{\max}$)	(0.2 Mach, 2 Mach)
Given Resolution ($\Delta\rho, \Delta\theta$)	($0.3\,\text{km}$, 2°)
Maximum Sampling Number S_{sample}	$n \times C_K^2$

Figure 9 is the data plane which is overlaid by all of the 40 scans measured data. In the range of $x < 200\,\text{km}$, there are just some sporadic clutters and these clutters are basically in different equidistant ring to the radar. So we deem that these clutters don't threaten to the radar's surveillance area. In addition, considering the need for distance early warning and to reduce the operand, we just intercept an interesting region to process, as shown in Figure 10.

Using the traditional RHT and the proposed algorithm to process the measured data in each sliding window, after the completion of sampling and voting, we find the global maximum poll in the DLL and then restore the point pairs which vote for the maximum poll in the resultant plane. With sliding the windows and processing the data constantly, we will get the integrated target's track at last.

Figure 11 and Figure 12 are the results using the traditional RHT and the proposed algorithm in this paper to process the interception region data. Here, the positions of polling point pairs are also recovered in the result figures.

In Figure 11 and Figure 12, the solid line is the target's track

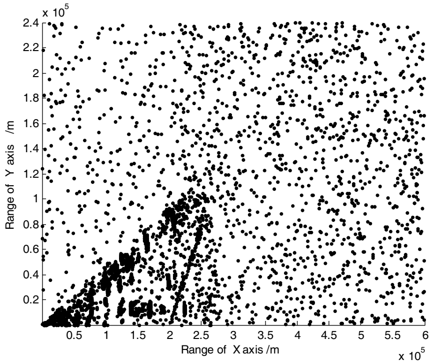

Figure 9. Image of 40 scans measured data.

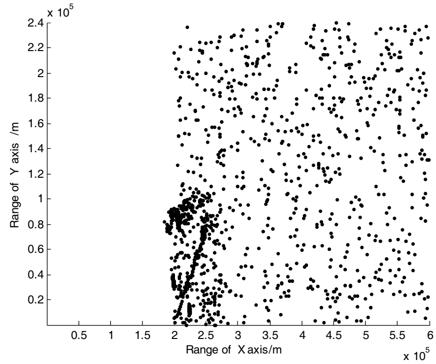

Figure 10. Image of the intercepted region.

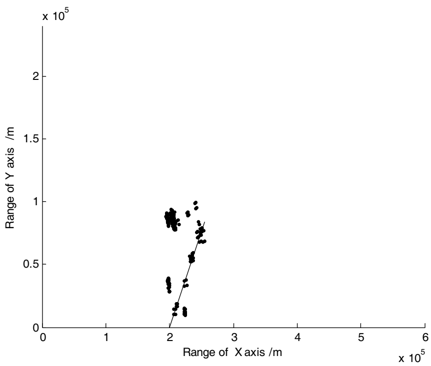

Figure 11. Result of the traditional RHT.

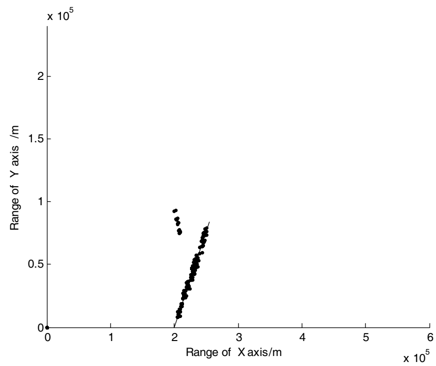

Figure 12. Result of the proposed algorithm.

and it is obtained by the target's movement parameters based on the minimum variance criteria. Dots are the recovered target's positions. As Figure 11 shows, using traditional RHT to process the mentioned measured data, there are lots of false alarms and the targets' track is not detected almost. The total processing time of 40 scans data is 102.57 seconds under the CPU Intel P4 2.4G environment.

As Figure 12 shows, using the proposed algorithm to process the mentioned measured data, the target's track is detected accurately and almost has no false alarms. And the total processing time of 40 scans data is 7.609 seconds under the CPU Intel P4 2.4G environment. The mean error of parameter rho is 64.85 m and the mean error of parameter theta is 0.292 degree.

6. CONCLUSION

A TBD algorithm based on improved Randomized Hough Transform for dim target detection is proposed in this paper. This algorithm offers three improvements on the traditional RHT. (1) It uses the scans' sequence numbers to make sure that the point pairs are selected randomly from different scans, avoiding the unreasonable situation of detecting the target's track in a single-scan radar echo data; (2) It introduces a new voting method that outputs better voting result than traditional RHT; (3) At the same time as voting, the parameter of the optimal parameter cell is also updated to suppress the deviation between the detected track and the target's track. As the simulation results show, compared with the traditional RHT, the proposed algorithm is more insensitive to noises and can detect the target's track accurately. In addition, it has a better real-time performance than traditional RHT.

REFERENCES

1. Xue, W. and X.-W. Sun, "Multiple targets detection method based on binary Hough Transform and adaptive time-frequency filtering," *Progress In Electromagnetics Research*, Vol. 74, 309–317, 2007.
2. Hatam, M., A. Sheikhi, and M. A. Masnadi-Shirazi, "Target detection in pulse-train MIMO radars applying ica algorithms," *Progress In Electromagnetics Research*, Vol. 122, 413–435, 2012.
3. Habib, M. A., M. Barkat, B. Aissa, and T. A. Denidni, "Ca-Cfar detection performance of radar targets embedded in 'noncentered Chi-2 gamma' clutter," *Progress In Electromagnetics Research*, Vol. 88, 135–148, 2008.
4. Alyt, O. A. M., A. S. Omar, and A. Z. Elsherbeni, "Detection and localization of RF radar pulses in noise environments using wavelet packet transform and higher order statistics," *Progress In Electromagnetics Research*, Vol. 58, 301–317, 2006.
5. Haridim, M., H. Matzner, Y. Ben-Ezra, and J. Gavan, "Cooperative targets detection and tracking range maximization using multimode ladar/radar and transponders," *Progress In Electromagnetics Research*, Vol. 44, 217–229, 2004.
6. Singh, A. K., P. Kumar, T. Chakravarty, G. Singh, and S. Bhooshan, "A novel digital beamformer with low angle resolution for vehicle tracking radar," *Progress In Electromagnetics Research*, Vol. 66, 226–237, 2006.

7. Liu, H.-Q. and H.-C. So, "Target tracking with line-of-sight identification in sensor networks under unknown measurement noises," *Progress In Electromagnetics Research*, Vol. 97, 373–389, 2009.
8. Bi, S. and X. Y. Ren, "Maneuvering target Doppler-bearing tracking with signal time delay using interacting multiple model algorithms," *Progress In Electromagnetics Research*, Vol. 87, 15–41, 2008.
9. Shi, Z.-G., S.-H. Hong, and K. S. Chen, "Tracking airborne targets hidden in blind doppler using current statistical model particle filter," *Progress In Electromagnetics Research*, Vol. 82, 227–240, 2008.
10. Hong, S., L. Wang, Z.-G. Shi, and K. S. Chen, "Simplified particle Phd filter for multiple-target tracking: Algorithm and architecture," *Progress In Electromagnetics Research*, Vol. 120, 481–498, 2011.
11. Kural, F., F. Arikan, O. Arikan, and M. Efe, "Performance evaluation of track association and maintenance for a Mfpar with doppler velocity measurements," *Progress In Electromagnetics Research*, Vol. 108, 249–275, 2010.
12. Carlson, B. D., E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough Transform, Part I: System concept," *IEEE Trans. Aerosp. Electron. Sys.*, Vol. 30, No. 1, 102–108, 1994.
13. Carlson, B. D., E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough Transform, Part II: Detection statistics," *IEEE Trans. Aerosp. Electron. Sys.*, Vol. 30, No. 1, 109–115, 1994.
14. Carlson, B. D., E. D. Evans, and S. L. Wilson, "Search radar detection and track with the Hough Transform, Part III: Detection performance with binary integration," *IEEE Trans. Aerosp. Electron. Sys.*, Vol. 30, No. 1, 116–125, 1994.
15. Wallace, W. R., "The use of track-before-detect in pulse-Doppler radar," *Proc. Int. Conf. Radar*, 315–319, 2002.
16. Buzzi, S., M. Lops, and L. Venturino, "Track-before-detect procedures for early detection of moving target from airborne radars," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 41, No. 3, 937–954, 2005.
17. Buzzi, S., M. Lops, and M. Ferri, "Track-before-detect procedures in a Multi-target environment," *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 44, No. 3, 1135–1148, 2008.

18. Deng, X., Y. Pi, M. Morelande, and B. Moran, "Track-before-detect procedures for low pulse repetition frequency surveillance radars," *IET Proc. Radar Sonar Navig.*, Vol. 5, No. 1, 65–73, 2011.
19. Boers, Y. and H. Driessen, "Multitarget particle filter track before detect application," *IEE Proc. Radar Sonar Navig.*, Vol. 151, No. 6, 351–357, 2004.
20. Fan, L., X. Zhang, and J. Shi, "Track-before-detect procedures for detection of extended object," *EURASIP Journal on Advances in Signal Processing*, Vol. 2011, No. 35, 2011.
21. Hough, P. V. C., "Methods and means for recognizing complex patterns," U.S. Patent 3069654, 1962.
22. Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to detect lines and curves in pictures," *Communication of the ACM*, Vol. 15, No. 1, 11–15, 1972.
23. Kultanen, P., L. Xu, and E. Oja, "Randomized Hough Transform (RHT)," *10th International Conference on Pattern Recognition*, Vol. 1, 631–635, 1990.
24. Xu, L. and E. Oja, "Randomized Hough Transform (RHT): Basic mechanisms, algorithms, and computational complexities," *CVGIP Image Understanding*, Vol. 57, No. 2, 131–154, 1993.