RESEARCH

Open Access

Dynamic programming network for point target detection



Jingneng Fu^{1,2,3,4*} and Hongyan Wei^{1,2,3}

*Correspondence: fujingneng@ioe.ac.cn

 ¹ Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China
 ² Key Laboratory of Science and Technology on Space Optoelectronic Precision Measurement, Chinese Academy of Sciences, Chengdu 610209, China
 ³ University of Chinago Academy

³ University of Chinese Academy of Sciences, Beijing 100049, China

⁴ Youth Innovation Promotion Association, Chinese Academy of Sciences, Beijing 100029, China

Abstract

To improve the efficiency of the dim point target detection based on dynamic programming (DP), this paper proposes a multi-frame target detection method based on a DP ring network (DPRN). In the proposed method, first, the target trajectory is approximated using the piecewise linear function. The velocity space partition DP (VSP-DP) is used to accumulate the merit functions of a target on each piecewise linear trajectory segment to avoid the merit function diffusion in different velocity spaces. In addition, the velocity space matching DP (VSM-DP) is employed to realize the state transition of a target between adjacent piecewise linear trajectory segments. Then, the VSP-DP and VSM-DP are used to construct a DP network (DPN). Second, to suppress the merit function diffusion further, the sequential and reverse DPNs are connected in a head-to-tail manner to form a DPRN, and the merit function of the DPRN is obtained by averaging the merit functions of the sequential and reverse DPNs. Finally, the target trajectory is obtained by tracking the extreme points of the merit functions of the DPRN. The simulation and analysis results show that the proposed DPRN combines the advantages of high detection probability of the high-order DP and high execution efficiency of the first-order DP. The proposed DPRN is suitable for radars and infrared searching and tracking systems.

Keywords: Point target detection, Dynamic programming, Dynamic programming ring, Dynamic programming network, Merit function diffusion suppression

1 Introduction

Dim point target detection and tracking have been one of the difficult problems in the field of radars and infrared searching and tracking systems. According to the detection and tracking order, the point target detection methods can be roughly divided into detection-before-track (DBT) methods [1–3] and track-before-detection (TBD) methods [4–6]. The amplitude of a dim point target can be lower than that of its surround-ing background, and the single-frame DBT can easily lose the dim target. Therefore, the TBD, which processes a number of frames before making a decision on the target's existence, is necessary. However, compared with the DBT, the TBD requires larger data storage and a wider search scope.

Dynamic programming (DP) divides a complex problem into a series of subproblems and searches for a possible optimal solution for each subproblem. Since Barniv et al. [7]



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativecommons.org/licenses/by/4.0/.

first used DP to develop the TBD, which can significantly reduce the requirements of data storage and search scope, the DP-TBD methods have been the mainstream dim point target detection approach. However, due to unavoidable incorrect state estimation of the dim point target in each transition stage, the DP-TBD methods face merit function diffusion [8–11]; namely, the distribution of the merit function at the target position is similar to the shape of a comet, which affects point target detection and tracking.

To address the aforementioned problems, many improved algorithms for merit function have been proposed, including algorithms based on amplitude constraints [12], system memory coefficients [13], multi-level thresholds [14, 15], velocity space partition (VSP) [16, 17], velocity space matching (VSM) [18–20], high-order DP [21–25], and DP ring (DPR) structure [25]. The introduction of amplitude constraints [12], system memory coefficients [13], and multi-level thresholds [14, 15] has a good effect in suppressing the merit function diffusion of strong targets, with a signal-to-noise ratio (SNR) of more than two, but can degrade the detection performance of dim targets. In the VSP-DP [16, 17], a merit function is assigned one velocity space, and the merit function is updated only in the corresponding velocity space; finally, the target detection is performed independently in each velocity space. Therefore, the VSP-DP can be regarded as a DP version of a three-dimensional (3D) Hough transform [5, 26-28]. The VSP-DP prevents the merit function of a target from diffusing to the other velocity spaces. However, for target maneuvers in multiple velocity spaces, the accumulation of target energy cannot be completed in each velocity space. The VSM-DP [18–20] matches the current velocity space according to the target state in the previous stage and then performs the DP accumulation in the matched velocity space. The VSM-DP can realize effective energy accumulation for a target transferring between different velocity spaces, but it aggravates merit function diffusion compared with the VSP-DP. The high-order DP $\begin{bmatrix} 21-25 \end{bmatrix}$ is equivalent to combining the 3D Hough transform and the VSM-DP, which can significantly suppress merit function diffusion and improve the probability of target detection. However, the high computational complexity of the 3D Hough transform reduces its detection efficiency, thus limiting the high-order DP application to point target detection. The main aim of the DP is to reduce the computational complexity of exhaustive algorithms, such as the Hough transform. The DP version of the 3D Hough transform is equivalent to the VSP-DP. Therefore, the key to reducing the complexity of the high-order DP is to design a DP that combines the VSP-DP and the VSM-DP. It should be noted that traditional DP can be transformed into the DPR [25] by introducing a ring data structure, after which the DPR can effectively suppress merit function diffusion, thus significantly improving the detection performance.

Aiming at suppressing the merit function diffusion and reducing algorithm complexity, this paper develops a multi-frame target detection algorithm based on a DP ring network (DPRN) according to the recent development trends in this field. First, the target trajectory is approximated using the piecewise linear function. The VSP-DP is used to accumulate the merit functions of the target in each piecewise linear trajectory segment to avoid the merit function diffusion in different velocity spaces. In addition, the VSM-DP is employed to realize the state transition of a target between adjacent piecewise linear trajectory segments. In this way, the structure combining the VSP-DP and VSM-DP forms a DP network (DPN). Second, to suppress the merit function diffusion further, the sequential and reverse DPNs are connected in a headto-tail manner to obtain a DPRN, and the merit function of the DPRN is obtained by averaging the merit functions of the sequential and reverse DPNs. Finally, the target trajectory is obtained by tracking the extreme points of the merit functions of the DPRN. The simulation and analysis results show that compared with the traditional DP and DPR, the proposed DPRN can significantly improve the detection probability of point targets, achieving high execution efficiency.

The remainder of this paper is organized as follows. Section 2 introduces the DPRN-based point target detection algorithm. Section 3 describes the simulation experiment, analyzes the effectiveness of the proposed algorithm, and compares the proposed algorithm with several typical DPR-based point target detection algorithms. Section 4 concludes the paper.

2 DPRN-based point target detection

2.1 Point target observation model

Radars or infrared searching and tracking systems obtain a two-dimensional (2D) image for a full scan, and the image plane combines the time axis to form a 3D observation space. When long-distance targets are moving in the surveillance region of a searching and tracking system, an image sequence of observation data corresponding to the targets is obtained.

At time $t, 1 \le t \le N$, observation data with coordinates p on an image plane Ω are denoted by $X_p^{(t)}$ and expressed as follows [18]:

$$X_{\boldsymbol{p}}^{(t)} = \begin{cases} A^{(t)} + n_{\boldsymbol{p}}^{(t)}, \text{ target at coordinates of } \boldsymbol{p}, \\ n_{\boldsymbol{p}}^{(t)}, & \text{no target at coordinates of } \boldsymbol{p}, \end{cases}$$
(1)

where $n_p^{(t)}$ represents an additive noise that obeys the zero-mean Gaussian distribution, and $n_p^{(t)} \sim N(0, \sigma_n^2)$; $A^{(t)}$ denotes the target amplitude, which is assumed to be a positive constant, i.e., $A^{(t)} = A > 0$. Thus, SNR can be defined as A/σ_n . In this study, the point target is assumed to be a single pixel for simplicity. More details about processing complex background and extended target can be found in [29].

According to the DP-based point target detection, this study uses the merit function and velocity as energy and motion features of a point target, respectively. Thus, the state of all pixels in image plane Ω at time t can be defined by $S_{\Omega}^{(t)} = \left\{ \left(I_p^{(t)}, v_p^{(t)} \right) | p \in \Omega \right\}$, where $I_p^{(t)}$ and $v_p^{(t)}$ represent the DP merit function and velocity of a pixel at coordinates p at time t, respectively.

In addition, the trajectory of a point target is regarded as a continuous curve in the 3D observation space, and the geometric features of the point target trajectory are equivalent to the dynamic features of a point target. However, when the observation time is short enough, the trajectory of the point target can be approximated as a straight line in a 3D space. In view of that, this study considers the piecewise linear target trajectory in a 3D space as dynamic feature of a point target, as shown in Fig. 1. Thus, the merit function of a target on each piecewise linear trajectory segment can be defined in a small velocity subspace to suppress the merit function diffusion.



Fig. 1 Target trajectory approximated by the piecewise linear function



Fig. 2 Schematic diagram of the DPN

2.2 DPRN

For the convenience of discussion, N consecutive image frames are grouped into image sequences with a length of L. Image sequences with a length of L denote the basic units of image processing, as shown in Fig. 2. The VSP-DP is used to assign merit functions with different velocity spaces in each piecewise linear trajectory segment to prevent the merit functions from diffusing to the other velocity spaces. Meanwhile, the VSM-DP is used to realize the state transition of a target between adjacent piecewise linear trajectory segments.

Under the constraint that the speed of a point target should not exceed $\nu_{\max} \in Z^+$ pixel/frame, and according to the speed estimation accuracy of the first-order DP (i.e., 1 pixel/frame), the velocity space is partitioned as follows:

$$[-\nu_{\max}, \nu_{\max}]^2 = \mathcal{U}_{-\nu_{\max} \le \nu_X, \nu_Y < \nu_{\max}}[\nu_X, \nu_X + 1] \times [\nu_Y, \nu_Y + 1],$$

$$\nu_{X, \nu_Y \in Z^+}$$
(2)

where v_X and v_Y represent the projection components of the velocity in the *X* and *Y* directions of the image coordinate system, respectively.

Equation 2 shows that the velocity space can be divided into $K = 4v_{max}^2$ subspaces, each of which corresponds to four valid transition states $\{(v_X, v_Y), (v_X + 1, v_Y), (v_X, v_Y + 1), (v_X + 1, v_Y + 1)\}$, as shown in Fig. 3. Furthermore, the center of the velocity subspace is expressed as $(\overline{v}_X, \overline{v}_Y) = (v_X + 0.5, v_Y + 0.5)$, and the velocity subspaces are labeled according to Eq. 2 as follows:

$$k = 2\nu_{\max}(\nu_{\max} + \overline{\nu}_X - 0.5) + (\nu_{\max} + \overline{\nu}_Y - 0.5) + 1, \ 1 \le k \le K;$$
(3)

the state transition set corresponding to the velocity subspace label $k(1 \le k \le K)$ is denoted as Ω_k .

Current Location 🖾 Valid State Transitions in Next Frame



Fig. 3 Under the constraint that the speed of a point target should not exceed $v_{max} = 2$ pixel/frame, the velocity space can be partitioned into K = 16 subspaces, which correspond to the four valid transition states

When time *t* satisfies the condition of $t\% L \neq 1$, the VSP-DP operation is performed. In the velocity subspace labeled *k*, state $(I_{p,k}^{(t)}, v_{p,k}^{(t)})$ of a pixel with the coordinates *p* is defined as follows:

$$\begin{cases} I_{\boldsymbol{p},\boldsymbol{k}}^{(t)} = X_{\boldsymbol{p}}^{(t)} + \max_{\boldsymbol{q} \in \boldsymbol{p} - \boldsymbol{\Omega}_{\boldsymbol{k}}} \left\{ I_{\boldsymbol{q},\boldsymbol{k}}^{(t-1)} \right\}, \\ \boldsymbol{\nu}_{\boldsymbol{p},\boldsymbol{k}}^{(t)} = \boldsymbol{p} - \operatorname{argmax}_{\boldsymbol{q} \in \boldsymbol{p} - \boldsymbol{\Omega}_{\boldsymbol{k}}} \left\{ I_{\boldsymbol{q},\boldsymbol{k}}^{(t-1)} \right\}, \end{cases}$$

$$\tag{4}$$

where % represents the modulo operation.

When time *t* satisfies the condition of t%L = 0 or is equal to one, the VSM-DP operation, including the state merging and state splitting, is performed.

State merging is performed at the state node t%L = 0, and the post-merge state $(I_p^{(t)}, v_p^{(t)})$ of a pixel with the coordinate p is defined by:

$$\begin{cases} I_{p}^{(t)} = I_{p,k_{p}^{(t)}}^{(t)}, \\ \mathbf{v}_{p}^{(t)} = \mathbf{v}_{p,k_{p}^{(t)}}^{(t)}, \end{cases}$$
(5)

where the velocity subspace label correspond to the post-merge state and is given by:

$$k_{\boldsymbol{p}}^{(t)} = \operatorname{argmax}_{k} \left\{ I_{\boldsymbol{p},k}^{(t)} \right\}.$$
(6)

When t = 1, the initialization step is executed, the state of the velocity subspace labeled $k(1 \le k \le K)$ is defined by:

$$\begin{cases} I_{\mathbf{\Omega},k}^{(t)} = X_{\mathbf{\Omega}}^{(t)}, \\ \mathbf{v}_{\mathbf{\Omega},k}^{(t)} = 0. \end{cases}$$
(7)

When time *t* satisfies the conditions of t % L = 1 and $t \neq 1$, the state splitting is performed. In the velocity subspace labeled *k*, the state $(I_{p,k}^{(t)}, v_{p,k}^{(t)})$ of a pixel with coordinates *p* is defined as follows:

$$\begin{cases} I_{p,k}^{(t)} = X_p^{(t)} + \max_{q \in p - \Omega_k} \left\{ w_{p,q}^{(t)} I_q^{(t-1)} \right\}, \\ v_{p,k}^{(t)} = p - \operatorname{argmax}_{q \in p - \Omega_k} \left\{ w_{p,q}^{(t)} I_q^{(t-1)} \right\}, \end{cases}$$
(8)

where the speed matching flag is given by:

$$w_{\boldsymbol{p},\boldsymbol{q}}^{(t)} = \begin{cases} 1, & \left\| \boldsymbol{p} - \boldsymbol{q} - \boldsymbol{v}_{\boldsymbol{q}}^{(t-1)} \right\|_{\infty} \le 1, \\ 0, & \text{otherwise,} \end{cases}$$
(9)

where $\|\cdot\|_{\infty}$ represents the Chebyshev norm.

Then, the structure of the above-mentioned two-level DP forms a DP network (DPN), as shown in Fig. 2. The role of the basic unit length *L* of the DPN is similar to the order of a high-order DP. When L = 1, the DPN degenerates into the VSM-DP; but when L = N, the DPN degenerates into the VSP-DP.

The detection probability can be significantly improved by merely adding a ring data structure to the DP-TBD. Finally, the sequential and reverse observation data are concatenated in a head-to-tail manner, and the DPN is made to run on the ring structure; thus, the DPN becomes a DPRN, as shown in Fig. 4. In the velocity subspace labeled k, the states of the sequential and reverse DPNs at time $t \operatorname{are} \left(I_{\Omega,k}^{(+t)}, \mathbf{v}_{\Omega,k}^{(+t)}\right)$ and $\left(I_{\Omega,k}^{(-t)}, \mathbf{v}_{\Omega,k}^{(-t)}\right)$, respectively; thus, the state $\left(I_{\Omega,k}^{(*t)}, \mathbf{v}_{\Omega,k}^{(*t)}\right)$ of the DPRN can be expressed by:

$$\begin{cases} I_{\mathbf{\Omega},k}^{(*t)} = \frac{1}{2} \left(I_{\mathbf{\Omega},k^+}^{(+t)} + I_{\mathbf{\Omega},k^-}^{(-t)} \right), \\ \nu_{\mathbf{\Omega},k}^{(*t)} = \frac{\mathbf{v}_{\mathbf{\Omega},k^+}^{(+t)} I_{\mathbf{\Omega},k^-}^{(+t)} - \mathbf{v}_{\mathbf{\Omega},k^-}^{(-t)} I_{\mathbf{\Omega},k^-}^{(-t)}}{I_{\mathbf{\Omega},k^+}^{(+t)} + I_{\mathbf{\Omega},k^-}^{(-t)}}, \end{cases}$$
(10)



Fig. 4 Schematic diagram of the DPRN

where the DPRN has the same time-reversal symmetry as the DPR, so the velocity subspace label of the sequential DPN is $k^+ = k$, as given by Eq. 3, and the corresponding velocity subspace label of the reverse DPN is defined by:

$$k^{-} = 2\nu_{\max}(\nu_{\max} - \bar{\nu}_X - 0.5) + (\nu_{\max} - \bar{\nu}_Y - 0.5) + 1.$$
(11)

2.3 Multi-target detection

When there are multiple targets in the field of view, their merit functions interfere, bringing additional challenges to multi-target detection. To overcome these challenges, this paper extracts the target coordinates multiple times and uses the extracted coordinates to construct target trajectories one by one. The execution procedure of a multi-target detection algorithm based on the DPRN is given in Algorithm 1. The execution procedure of the multi-target detection algorithms based on the DPRN is similar to that of the DPRN-based algorithms, except that the DPR-based algorithms do not require VSP and thus do not execute Step 3.

Algorithm 1 DPRN-Based Multi-Target Detection.

Input: Set the sequence length *N*; the basic unit length of the DPRN *L*; the upper limit of the target motion speed ν_{max} ; the number of iterations is equal to the target number N_{tar} ; the iteration counter is set to l = 1;

Output: The multi-target trajectories $\left\{ \boldsymbol{p}_{i}^{(t)} | 1 \leq i \leq N_{\text{tar}}, 1 \leq t \leq N \right\}$.

Step 1: Energy accumulation.

Run the DPRN to obtain the *l* th iteration merit functions, which are denoted by $\{I_{\mathbf{Q},k}^{(*t,l)}|1 \le t \le N, 1 \le k \le K\}$.

Step 2: Merit function maximum value coordinate extraction.

At time *t*, the coordinates corresponding to the maximum values of the merit function image $I_{\Omega,k}^{(*t,l)}$ of the velocity subspace labeled by *k* are given by:

$$\boldsymbol{p}_{k}^{(*t,l)} = \operatorname{argmax}_{\boldsymbol{p} \in \Omega} \left\{ I_{\boldsymbol{p},k}^{(*t,l)} \right\}, 1 < t \le N, \ 1 \le k \le K;$$
(12)

Step 3: Velocity subspace determination.

If it holds that,

$$k^{(t_{\text{Nod}},l)} = \operatorname{argmax}_{k} \left\{ \sum_{t_{\text{Nod}} - L < t \le t_{\text{Nod}}} I_{p_{k}^{(*t,l)},k}^{(*t,l)} \right\};$$
(13)

then, it is considered that the target state is in the velocity subspace labeled by $k^{(t_{\text{Nod}},l)}$ in the time period of $t_{\text{Nod}} - L < t \leq t_{\text{Nod}}$, and the coordinates corresponding to the maximum values of the merit function image in the velocity subspace are extracted as follows:

$$\boldsymbol{p}^{(*t,l)} = \boldsymbol{p}_{k^{(t_{\text{Nod}},l)}}^{(*t,l)}, t_{\text{Nod}} - L < t \le t_{\text{Nod}},$$
(14)

where $t_{\text{Nod}}(t_{\text{Nod}}\%L = 0)$ is the time at the state node.

Step 4: Trajectory detection.

The trajectory $\left\{ \boldsymbol{p}_{traj}^{(t,l)} | 1 \le t \le N \right\}$ can be generated by performing the moving pipeline filtering [30] on the coordinate set $\left\{ \boldsymbol{p}^{(*t,l)} | 1 \le t \le N \right\}$.

The iteration counter is increased by one (i.e., l = l + 1); if $l \le N_{\text{tar}}$, the algorithm goes to Step 5; otherwise, it goes to Step 6.

Step 5: Updating observation data.

To prevent the same trajectory is detected multiple times, once the trajectory $\left\{ \boldsymbol{p}_{traj}^{(t,l)} | 1 \leq t \leq N \right\}$ is detected, the corresponding pixels of the observation data in the moving pipeline are replaced by median operation; go to Step 1.

Step 6: Trajectory regularization.

If any two trajectories intersect, the trajectory regularization is performed according to the trajectory segment fitting error [25].

Finally, the multi-target trajectories $\left\{ \boldsymbol{p}_{i}^{(t)} | 1 \leq i \leq N_{\text{tar}}, 1 \leq t \leq N \right\}$ are output.

3 Simulations and analysis

The DPRN-based point target detection algorithm was simulated and verified using MATLAB software running on a computer with a processor i5-2400CPU at 3.10 GHz and 3.40-GB memory. The simulation experiment consisted of two parts: single-target detection and multi-target detection.

Considering that the traditional DPs and the DPRs have a large difference in detection performance, and the DPRN is also a DPR, the DPRN was compared with several DPRs developed from traditional DPs in the experiment. The comparison algorithms included the CFO-DPR evolved from the classic first-order DP algorithm (CFO-DP) [18], the CSO-DPR developed from the classic second-order DP (CSO-DP) with backtracking [24], and the second-order DPR with merit function filtering

(MFF-DPR) [25]. The comparison between the classic DPs and the corresponding DPRs, is given in the experimental part in [25].

3.1 Single-target detection test

The images used in this simulation had a size of 128×128 pixels, a maximum sequence length of 100, and a background noise of $n_p^{(t)} \sim N(0, 1)$. Three types of trajectories were tested in the single-point target detection experiment, as shown in Fig. 5. In Fig. 5, the straight-line trajectory 1 is marked in red; its initial coordinates were (5, 10), and the motion velocity was (1.2, 1.1) pixel/frame; the straight-line trajectory 2 is marked in blue; its initial coordinates were (30, 20), and the motion velocity was (0.6, 0.8) pixel/ frame; the arc trajectory is marked in green; its motion speed of 1 pixel/frame, arc center coordinates of (64, 64), and an arc radius of 20 pixels.

The SNRs of the point targets on each trajectory were in the range of 1.5–3.0. For each group of test data, 1000 rounds of simulations were conducted, and the average detection probability was calculated by:

$$P_d = \frac{\text{Number of correctly detected targets}}{\text{Total number of real targets}}.$$
(15)

According to the point target detection algorithm described in Sect. 2.3, only one round of the DPR signal accumulation was needed, and only a single-pixel coordinate needed to be extracted from a single image frame. The single-point target detection process was as follows.

Assume that the calculated coordinates of a point target at time *t* were $p_c^{(t)}$ (see Eq. 14), and the theoretical coordinates of the point target were $p_r^{(t)}$; then, if $\left\| p_c^{(t)} - p_r^{(t)} \right\|_{\infty} \le 1$, it was deemed that the point target was detected at time *t*.

First, the single-target detection performance of the DPRN was tested. In the three test cases shown in Fig. 5, the number of image frames was N = 100. In test case 1, the maximum target motion speed was $v_{max} = 2$ pixel/frame; in test cases 2 and 3, the maximum target motion speed was $v_{max} = 1$ pixel/frame. To observe the influence of length L of a basic DPRN unit on the detection performance, different L values were set in the tests, including values of 5, 10, 20, 50, and 100.

For a target moving along a straight line at a constant speed, as shown in Fig. 6a and b, the longer the length L of a basic DPRN unit was, the higher the target detection



Fig. 5 Three types of trajectories in the single-target detection experiment: a test case 1; b test case 2; c test case 3



Fig. 6 Single-target detection probability results of the DPRN algorithm for the image sequence length of N = 100: **a** test case 1; **b** test case 2; **c** test case 3

probability was. However, when $L \ge 50$, the increasing rate of the detection probability slowed down. The DPRN could detect not only targets moving along a straight line at a constant speed but also maneuvering targets. The maximum length of the basic unit was determined by the maneuvering characteristics of a target, as shown in Fig. 6c. When the basic unit length L was too large (e.g.,L = 100, Fig. 6c), the transfer of the target between different velocity spaces was limited, making it impossible to detect the complete target trajectory. Therefore, to achieve a balance between the detection probability and the adaptability in the state transition, the basic unit length was set to L = 20 in this study.

Next, the single-target detection performance of the DPRN was compared with those of the CFO-DPR, CSO-DPR, and MFF-DPR. In the three test cases, different numbers of data frames N were used, N = 20, 50, 100 (Fig. 7).

With the increase in the SNR value or the number of data frames, the detection probabilities of the four DPR algorithms gradually increased. As shown in Fig. 8a and b, the smaller the value of v_{max} was, the smaller the number of state transitions and the influence of noise interference were, and the higher the detection probability was. The number of single-step state searches defined the upper limit of the DPR detection performance, and the number of single-step state searches was approximately proportional to the square of the maximum target motion speed v_{max} . When $v_{max} = 2$ pixel/frame, the single-target detection probabilities of the four DPR algorithms decreased gradually at the SNR of 1.5. When the SNR was 1.5, the detection probability P_d of the DPRN could not reach 80%, as shown in Fig. 8a. Therefore, for detecting a dim point target from a long distance, it was necessary to eliminate the influence of the detection platform motion or increase the imaging frame rate.

Generally, the DPRN algorithm achieved excellent performance in merit function diffusion suppression as shown in Fig. 7, and the detection probability performances of the four DPR algorithms ranked in descending order were: DPRN > MFF-DPR > CSO-DPR > CFO-DPR. The target detection result obtained by the DPRN using only 20 image frames was better than that of the MFF-DPR using 100 image frames. When SNR = 1.5, the detection probability of the DPRN was 10% higher than that of the MFF-DPR; at SNR \geq 1.8, the detection probability of the DPRN is



Fig. 7 The distributions of different merit functions of the test case 2. **a**, **d**, **g**, **j** Distributions of merit functions with sequential accumulation at time t = 50; **b**, **e**, **h**, **k** distributions of merit functions with reverse accumulation at time t = 50; **c**, **f**, **i**, **l** distributions of averaged merit functions with sequential accumulation and reverse accumulation at time t = 50; **a**-**c** the calculation results of the CFO-DPR; **d**-**f** the calculation results of the CFO-DPR; **g**-**i** the calculation results of the MFF-DPR; **j**-**l** the calculation results of the DPRN in K = 4 velocity subspaces; the basic unit length of the DPRN is L = 20. The sequence length is N = 100, the point target is SNR = 1.5

higher than 95%. This could be due to the DPRN characteristics. Namely, although the DPRN was similar to the high-order DPR, unlike the high-order DPR, the DPRN did not have to handle the high computational complexity, so it could use a large L value (i.e., $L \ge 10$). In addition, when the SNR value was in the range of 1.5–3.0, the



Fig. 8 Single-target detection probability results of different DPR algorithms: a test case 1; b test case 2; c test case 3

 Table 1
 Single-frame processing times of the DPRs in the single-target detection (unit: ms)

Target speed limit <i>v</i> _{max} (pixel/frame)	CFO-DPR	CSO-DPR	MFF-DPR	DPRN (with/without parallel computing)
1	29	117	155	31/47
2	63	343	449	84/175

difference between the detection probability values obtained at N = 50 and N = 100 was less than 1% in the single-target detection test of the DPRN algorithm. Therefore, increasing the number of image frames blindly cannot improve the detection performance. Thus, it is necessary to reasonably choose the number of data frames according to the target SNR to reduce unnecessary calculation while ensuring high detection probability.

Finally, the operation efficiency of the four detection algorithms in single-target detection was evaluated using the metric of single-frame processing time, as shown in Table 1. The principal factor affecting the single-frame processing time was the number of single-step state searches. As the first-order algorithms, the CFO-DPR and DPRN algorithms could process each image frame with fewer single-step state searches compared to the CSO-DPR and MFF-DPR algorithms, which were the second-order algorithms. Therefore, the first-order algorithms have higher operation efficiency than the second-order algorithms. Different from the CFO-DPR, in the DPRN, there was an overlap between the velocity subspaces, and the size of the stored data that needed to be processed by the DPRN was K times that of the CFO-DPR; thus, the DPRN required more computing resources than the CFO-DPR. However, these negative factors were counteracted by the adoption of a parallel data structure by the VSP-DP in DPRN. When tested on a computer with an i5-2400 four-core CPU, the calculation time of the DPRN algorithm was reduced by 30% compared with that of the DPRN algorithm without parallel computing toolbox of MATLAB. In particular, when $v_{max} = 2$ pixel/frame and DPRN contained K = 16

velocity subspaces, the operation efficiency was increased by 50%. After the adoption of parallel processing, the calculation time of the DPRN was very close to that of the CFO-DPR.

3.2 Multi-target detection test

In the multi-point target detection test, the size of the simulation images was 128×128 pixels, the sequence length was 100, and the background noise was $n_p^{(t)} \sim N(0, 1)$. In this test, three types of test cases were used, as shown in Fig. 9. In Fig. 9, the straight-line trajectory 1 is marked in red; its initial coordinates were (24, 34) and its motion velocity was (0.8, 0.6) pixel/frame; the straight-line trajectory 2 is marked in blue; it had the initial coordinates of (66, 73) and a motion velocity of (-0.6, -0.7) pixel/frame; the arc trajectory is marked in green; it had a motion speed of 1 pixel/frame, arc center coordinates of (64, 46), and an arc radius of 20 pixels. The two straight-line trajectories intersect at a point with the coordinates of (48, 52), while the straight-line trajectory 1 and arc trajectory intersect at a point with the coordinates of (67, 66).

In each test case, the SNR of a point target was in the range of 1.5–3.0, and 1000 simulations were performed for each data group.

After the target trajectories were generated using the DPR point target detection algorithms described in Sect. 2.3, the multi-target detection performance of each algorithm was evaluated using the detection probability given by Eq. 15 and the false alarm rate that was calculated by:

$$P_f = \frac{\text{Number of non} - \text{target points in all trajectories}}{\text{Number of pixels in the image sequence}}.$$
 (16)

First, commonalities of the multi-target detection algorithms to be tested were analyzed. Since only one coordinate point was extracted from each image frame in each round, and only one trajectory was retained in each round, the correlation complexity of the generated trajectory was greatly reduced, and the multi-target detection algorithm complexity was $O(N_{tar})$. Because the relative amplitudes of the merit functions of the targets could vary, and the intersection of multiple target trajectories could cause the merit functions to interfere with each other, it was challenging to extract a complete target trajectory after one execution round of the DP algorithm.



Fig. 9 Three types of trajectories in the multi-target detection experiment: a test case 1; b test case 2; c test case 3

However, when the original data samples corresponding to the trajectory segments were replaced through the process of median filtering, the trajectory segments did not be extracted repeatedly. Theoretically, a complete target trajectory could be constructed after multiple rounds of coordinate extraction. However, through trajectory regularization, the chance of erroneous target identification caused by trajectory intersection could be reduced.

Next, the performances of the four DPR-based multi-target detection algorithms were compared. The multi-target detection results obtained by the four DPR-based algorithms are shown in Fig. 10. Similar to the single-target detection results presented in Fig. 8, the performances of the four algorithms ranked in descending order were as follows: DPRN > MFF-DPR > CSO-DPR > CFO-DPR.

Compared with the single-target detection results in Fig. 8b and c, the performances of the multi-target algorithms in the multi-target detection test decreased to different extents. This could be due to two reasons. First, the single-target detection algorithm only needed to check whether the extracted target positions were correct, so there was no need for trajectory correlation. In contrast, the multi-target detection algorithm needed to correlate the trajectories with the targets, so only parts of a trajectory could be identified when the target trajectory was discontinuous. Second, the mutual influence between multiple target trajectories could also affect the detection performance. Since the introduction of the VSP into the DPRN reduced the mutual interference between target trajectories, the detection probability of the DPRN did not decrease significantly with the number of targets, which is more favorable compared to the other DPR-based algorithms.

With the increase in the SNR value, the false alarm probability results of the four multi-target detection algorithms showed a downward trend, as presented in Fig. 11. At the same SNR value, the false alarm probability of the DPRN was the lowest among all algorithms. Comparing the results in Fig. 11c with those in Fig. 11a and b, it could be concluded that the false alarm probability exhibited a salient upward trend with the number of target trajectories. This was because the mutual interference between target trajectories became more serious as the number of targets increased.



Fig. 10 Multi-target detection probability results of the DPR algorithms: a test case 1; b test case 2; c test case 3



Fig. 11 False alarm rate values of the DPR-based algorithms obtained in multi-target detection: **a** test case 1; **b** test case 2; **c** test case 3

Table 2 Single-frame processing times of different DPRs in multi-target detection (unit: ms)

	CFO-DPR	CSO-DPR	MFF-DPR	DPRN (with/without parallel computing)
Test case 1	65	236	315	64/97
Test case 2	65	237	313	65/98
Test case 3	95	353	485	97/146

Finally, the single-frame processing times of different algorithms in the single-target detection were compared, as shown in Table 1. In the multi-target detection, much time was spent on the DPR merit function updating, as shown in Table 2. The time required for target extraction, trajectory correlation, observation data updating, and trajectory regularization was about 10% of the merit function updating time. In addition, the two high-order DPRs, namely the CSO-DPR and MFF-DPR, process images slowly. In particular, the single-frame image processing time of the MFF-DPR algorithm in test case 3 was nearly 0.5 s, which indicated that this algorithm is not suitable for real-time processing.

When tested on a computer with the i5-2400 four-core CPU, the calculation time of parallel DPRN was at least 30% less than that of the DPRN without parallel computing. After the adoption of parallel processing, the calculation time of the DPRN was very close to that of the CFO-DPR. However, as an image batch processing algorithm, the parallel computing time of the DPRN could not still meet the real-time processing requirement of modern radars and infrared searching and tracking systems. Thus, how to reduce the complexity of DPRN algorithms could further be considered in future research.

4 Conclusions

This paper proposes a multi-frame target detection algorithm based on a DPRN. First, the target trajectory is approximated using the piecewise linear function. Then, the structure combining the VSP-DP and VSM-DP forms a DPN, which has the characteristics of high detection probability of the higher-order DP and efficient execution efficiency of the first-order DP. In addition, to suppress the merit function diffusion, the sequential and reverse DPNs are connected in a head-to-tail manner to form a DPRN, and the merit function of the DPRN is obtained by averaging the merit functions of the sequential and reverse DPNs. Finally, the target trajectory is obtained by tracking the extreme points of the merit functions of the DPRN. The simulation and analysis results show that the proposed DPRN is suitable for radars and infrared point target detection systems.

However, the DPRN shares certain deficiencies with the other DPR algorithms, such as an image batch processing algorithm; also, the DPRN needs to store a large amount of data, and its algorithm complexity is high, so it cannot meet the real-time processing requirement of radars and infrared searching and tracking systems. Developing an improved version of the DPRN with image sequential processing capability could be a focus of future research. The maneuvering characteristics of the target define the upper limit of the basic unit length, thus defining the detection capability of the DPRN. Therefore, it would be of great importance to determine the limit of target detection according to the maneuvering characteristics of a target.

Abbreviations

DP	Dynamic programming
DPR	Dynamic programming ring
DPN	Dynamic programming network
DPRN	Dynamic programming ring network
CSO-DP	Classic second-order dynamic programming ring
CFO-DPR	Classic first-order dynamic programming ring
MFF-DPR	Merit function filtering dynamic programming ring
3D	Three-dimensional
DBT	Detection-before-track
TBD	Track-before-detection
VSP	Velocity space partition
VSM	Velocity space matching
SNR	Signal-to-noise ratio

Acknowledgements

The authors would like to thank their colleagues working with them in the Institute of Optics and Electronics at the Chinese Academy of Sciences. The authors also would like to thank the anonymous reviewers for their constructive comments and suggestions.

Author contributions

The authors' contributions are as follows. JF proposed the idea, programmed the method, and revised the manuscript. HW wrote the first version of the manuscript and performed an in-depth discussion of the related literature. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No. 2022381).

Availability of data and materials

Data sharing is not applicable to this study.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 9 February 2023 Accepted: 20 June 2023 Published online: 26 June 2023

References

- K. Xie, K. Fu, T. Zhou et al., Small target detection based on accumulated center-surround difference measure. Infrared Phys. Technol. 67 (2014), 229–236 (2014)
- J. Han, Y. Ma, B. Zhou et al., A robust infrared small target detection algorithm based on human visual system. IEEE Geosci. Remote Sens. Lett. 11(12), 2168–2172 (2014)
- 3. J. Fu, H. Zhang, H. Wei et al., Small bounding-box filter for small target detection. Opt. Eng. 60(3), 033107 (2021)
- 4. M. Ward, Target velocity identification using 3D matched filter with Nelder-Mead optimization, in *IEEE Aerospace Conference*, (2011) pp. 1–7
- 5. J. Fu, H. Wei, H. Zhang et al., Three-dimensional pipeline hough transform for small target detection. Opt. Eng. **60**(2), 023102 (2021)
- 6. B. Vo, A random finite set conjugate prior and application to multi-target tracking, in *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, (2011) pp. 431–436
- Y. Barniv, Dynamic programming solution for detecting dim moving targets. IEEE Trans. Aerosp. Electron. Syst. 21(1), 144–156 (1985)
- H. Jiang, W. Yi, L. Kong et al., Tracking targets in G0 clutter via dynamic programming based track-before-detect, in IEEE Radar Conference, (2015) pp. 10–15
- J. Arnold, S.W. Shaw, H. Pasternack, Efficient target tracking using dynamic programming. IEEE Trans. Aerosp. Electron. Syst. 29(1), 44–56 (1993)
- S.M. Tonissen, R.J. Evans, Target tracking using dynamic programming: algorithm and performance, in IEEE Conference on Decision and Control, (1995) pp. 2741–2746
- S.M. Tonissen, R.J. Evans, Peformance of dynamic programming techniques for track-before-detect. IEEE Trans. Aerosp. Electron. Syst. 32(4), 1440–1451 (1996)
- 12. Y. Guo, Z. Zeng, S. Zhao et al., An amplitude association dynamic programming TBD algorithm with multistatic radar, in *Chinese Control Conference*, (2016) pp. 5076–5079
- 13. R. Succary, H. Kalmanovitch, Y. Shurnik et al., Point target detection. Infrared Technol. Appl. 3(1), 671–675 (2003)
- L. Cai, C. Cao, Y. Wang et al., A secure threshold of dynamic programming techniques for track-before-detect, in *IET International Radar Conference*, (2013) pp. 14–16
- 15. E. Grossi, M. Lops, L. Venturino, Track-before-detect for multiframe detection with censored observations. IEEE Trans. Aerosp. Electron. Syst. **50**(3), 2032–2046 (2014)
- S. Chen, S. Xiao, H. Lu, Dim targets detection based on multi-regions dynamic programming and track matching. Infrared Laser Eng. 36(5), 738–741 (2007)
- Q. Guo, Z. Li, W. Song et al., Parallel computing based dynamic programming algorithm of track-before-detect. Symmetry 11(29), 11010029 (2019)
- L.A. Johnston, V. Krishnamurthy, Performance analysis of a dynamic programming track-before-detect algorithm. IEEE Trans. Aerosp. Electron. Syst. 38(1), 228–242 (2002)
- D. Orlando, G. Ricci, Y. Bar-Shalom, Track-before-detect algorithms for targets with kinematic constraints. IEEE Trans. Aerosp. Electron. Syst. 47(3), 1837–1849 (2011)
- 20. H. Xing, J. Suo, X. Liu, A dynamic programming track-before-detect algorithm with adaptive state transition set, in *International Conference in Communications, Signal Processing, and Systems*, (2020) pp. 638–646
- 21. D.K. Zheng, S.Y. Wang, J. Yang et al., A multi-frame association dynamic programming track-before-detect algorithm based on second order Markov target state model. J. Electron. Inf. Technol. **34**(4), 885–890 (2012)
- S. Wang, Y. Zhang, Improved dynamic programming algorithm for low SNR moving target detection. Syst. Eng. Electron. 38(10), 2244–2251 (2016)
- L. Sun, J. Wang, An improved track-before-detect algorithm for radar weak target detection. Radar Sci. Technol. 5(4), 292–295 (2007)
- H.U. Lin, S.Y. Wang, Y. Wan, Improvement on track-before-detect algorithm based on dynamic programming. J. Air Force Radar Acad. 24(2), 79–82 (2010)
- J. Fu, H. Zhang, W. Luo et al., Dynamic programming ring for point target detection. Appl. Sci. 12(1151), 12031151 (2022)
- 26. J. Hu, T. Zhang, Hough transform relative to a four-dimensional parameter space for the detection of constant velocity target. Opt. Eng. **49**(12), 1127–1134 (2010)
- 27. A. Moqiseh, M.M. Nayebi, 3-D hough transform for surveillance radar target detection, in *IEEE Radar Conference*, (2008) pp. 1–5
- B. Yan, N. Xu, W. Zhao et al., A three-dimensional hough transform-based track-before-detect technique for detecting extended targets in strong clutter backgrounds. Sensors 19(4), 30791600 (2019)
- O. Nichtern, S.R. Rotman, Parameter adjustment for a dynamic programming track-before-detect-based target detection algorithm. EURASIP J. Adv. Signal Process. 19(1), 1–19 (2008)
- K. Qian, S.H. Rong, K.H. Cheng, Anti-interference small target tracking from infrared dual waveband imagery. Infrared Phys. Technol. 118(3), 103882 (2021)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jingneng Fu is an associate professor in the Institute of Optics and Electronics (IOE) at the Chinese Academy of Sciences. He received his BS degree in optics from Chongqing University of Posts and Telecommunications in 2009 and his MS and PhD degrees in optics from the IOE in 2012 and 2022, respectively. His research interests include image analysis, computer vision, pattern recognition, and target tracking.

Hongyan Wei is an assistant research fellow in the Institute of Optics and Electronics (IOE) at the Chinese

Academy of Sciences. She received her BS degree in biomedical engineering from University of Electronic Science and Technology of China in 2007 and her MS degree in biomedical engineering from Tianjin University in 2009. Her research interests include image analysis, target tracking, and high-speed communication interface design.

Submit your manuscript to a SpringerOpen[™] journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com