sensors

MDPI

Article

Multitarget Tracking Algorithm Based on Adaptive Network Graph Segmentation in the Presence of Measurement Origin Uncertainty

Tianli Ma [1,2,*], Song Gao [1,2], Chaobo Chen [1] and Xiaoru Song [1]

[1] Autonomous Systems and Intelligent Control International Joint Research Center, Xi'An Technological University, Xi'an 710021, China; gaos@xatu.edu.cn (S.G.); chenchaobo@xatu.edu.cn (C.C.); songxiaoru@xatu.edu.cn (X.S.)

[2] School of Mechatronic Engineering, Xi'An Technological University, Xi'an 710021, China

[*] Correspondence: matianli111@xatu.edu.cn; Tel.: +86-158-2908-2533

Received: 11 September 2018; Accepted: 2 November 2018; Published: 6 November 2018

Abstract: To deal with the problem of multitarget tracking with measurement origin uncertainty, the paper presents a multitarget tracking algorithm based on Adaptive Network Graph Segmentation (ANGS). The multitarget tracking is firstly formulated as an Integer Programming problem for finding the maximum a posterior probability in a cost flow network. Then, a network structure is partitioned using an Adaptive Spectral Clustering algorithm based on the Nyström Method. In order to obtain the global optimal solution, the parallel A* search algorithm is used to process each sub-network. Moreover, the trajectory set is extracted by the Track Mosaic technique and Rauch–Tung–Striebel (RTS) smoother. Finally, the simulation results achieved for different clutter intensity indicate that the proposed algorithm has better tracking accuracy and robustness compared with the A* search algorithm, the successive shortest-path (SSP) algorithm and the shortest path faster (SPFA) algorithm.

Keywords: network flow theory; multitarget tracking; spectral clustering; A* search algorithm; RTS smoother; integer programming

1. Introduction

The purpose of multitarget tracking is to jointly estimate the number of targets and their state of motion from sensor data [1]. During the past decade, it has developed in a variety of directions, such as Air-Traffic Control [2], Marine Monitoring [3], Computer Vision [4], Autonomous Vehicle and Robot [5], etc. At present, multitarget tracking has achieved substantial advances [6]. However, the measurement origin uncertainty, for instance the unknown and time-varying number of targets, clutters, jamming signals and so forth seriously deteriorates the performance of the multitarget tracking system. Resolving the uncertainty of the measurement origin is a computationally expensive task which relied on the prior information about the target motion. To find the mapping from each measurement to its origin is often called measurement-to-track association or just data association [7,8].

Markov Chain Monte Carlo Data Association algorithm (MCMCDA) based on Bayesian Inference [9] and the Probability Hypothesis Density filter (PHD) based on finite set statistics (FISST) [10] have been proposed to cope with this problem of tracking multiple targets with measurement uncertainty. The MCMCDA algorithm can be viewed as a deferred-logic method since a track is decided based on the current and past measurements. It uses the Markov Chain Monte Carlo sampling instead of enumerating over all possible associations. In a PHD filter, it can estimate the target states by recursively computing the first-order moment of the multi-target state posterior probability distribution, without using the complex data association techniques. However, it was not designed to estimate the trajectories of targets. For this problem, Ba-Ngu Vo [11] proposed a newly labeled Random Finite Set (RFS) approach, known as the generalized labeled multi-Bernoulli (GLMB) filter, it can output trajectories and has a better performance in harsh environments.

Bar-Shalom [12] proposed an multidimensional assignment algorithm for solving the data association problem. In essence, the data association problem is converted to a combinatorial optimization problem under certain linear constraints where the total distance/benefit of assigning targets to measurements is minimized/maximized. There is a wide range of algorithms, such as Greedy algorithm [13], Genetic algorithm [14] and Lagrange relaxation theory [15], are used to find the sub-optimum solutions of the multidimensional assignment problem [16]. These approaches, while effective, need to solve the Non-deterministic Polynomial Complete (NPC) with a large amount of data. Goldberg [17] constructed an efficient min-cost flow framework. It has applied a scaling push relabel method to find the optimal solution. Under this framework, Zhang [18] formulated the multitarget data association problem as a maximum a posteriori (MAP) problem. It is mapped into the cost flow network and finds the global optimum solution by depending on the min-cost flow algorithm. An approach combining Dynamic Programming (DP) and Successive Shortest-Path algorithm (SSP) is presented by using Hidden Markov Model (HMM) in [19]. The multitarget tracking problem is formulated as an Integer Linear Programming (ILP) problem, and a greedy, successive shortest-path algorithm is used to reduce the runtime costs. For $k = 1$, this algorithm can obtain the global optimal solution. For $k > 1$, it only obtains the approximate solutions, where k is the unknown number of targets. The Shortest Path Faster algorithm (SPFA) is used to solve the Integer Programming problem of the min-cost flow network and quickly obtains the global optimal solution in [20]. The algorithm improves the robustness and tracking accuracy. In [21], an A* based tracking association algorithm is presented. The integer assumption is relaxed to the standard Linear Programming (LP) problem so that the global optimal solution can be obtained by the A* search algorithm.

The above-mentioned approaches are mainly focused on the object tracking based on video image. The available information of image targets are more than that of point targets. In addition, less clutter leads to simple network structure so that the aforementioned algorithms have a good tracking performance. For the problem of multiple point targets tracking in the presence of measurement origin uncertainty, the network may become more complicated that result in an enormous computational burden.

In this paper, a multitarget tracking algorithm based on adaptive network graph segmentation is proposed to address the problem of tracking multitarget with measurement uncertainty. Parallel search strategy is employed to solve the NP-complete problem. The network flow model of multitarget tracking is divided into different sub-graphs. The optimal trajectory is extracted by using the A* search algorithm. Our main contributions are: (1) a parallel network search framework is presented to cope with the multitarget tracking in the presence of measurement origin uncertainty; and (2) we proposed a adaptive spectral clustering algorithm based on the Nyström Method to obtain the network segmentation results for an unknown cluster number data set.

The rest of paper is organized as follows: the problem of multiple targets tracking is formulated as a cost flow network, and transform it into an Integer Programming problem in Section 2. In Section 3, the A* search algorithm is briefly reviewed. The original contribution of the paper is presented in Section 4, where we describe the adaptive spectral clustering algorithm based on the Nyström Method. Simulation results are provided in Section 5. Conclusions and possible extensions appear in Section 6.

2. Problem Formulation

The multitarget data association problem is regarded as a cost flow network. Let $\mathbf{Z} = \{\mathbf{z}_i\}$ be a set of measurements, $\mathbf{z}_i = \{\phi_i, \varphi_i, t_i\}$, where ϕ_i, φ_i is the position in x and y-axes, respectively. t_i is the time step of the measurement $\mathbf{z}_i$. $\mathbf{T}_k = \{\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, \ldots \mathbf{z}_{k_n}\}$ represents a trajectory. The set of trajectories is

$\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_L\}$, and the number of trajectories L is unknown. The key of the data association is to compute the maximum a posteriori (MAP) estimate of $\mathcal{T}$ given the measurement set $\mathbf{Z}$:

$$\begin{aligned} \mathcal{T}^* &= \arg\max_{\mathcal{T}} P(\mathcal{T}|\mathbf{Z}) \\ &= \arg\max_{\mathcal{T}} P(\mathbf{Z}|\mathcal{T})P(\mathcal{T}) \\ &= \arg\max_{\mathcal{T}} \prod_i P(\mathbf{z}_i|\mathcal{T})P(\mathcal{T}). \end{aligned} \tag{1}$$

Assume that tracks are independent from each other. The cost flow network framework of multitarget tracking is as follows:

$$P(\mathcal{T}) = \prod_{\mathbf{T}\in\mathcal{T}} P(\mathbf{T}), \tag{2}$$

$$P(\mathbf{T}) = P_s(\mathbf{z}_{k_0})P_l(\mathbf{z}_{k_1}|\mathbf{z}_{k_0})\ldots P_l(\mathbf{z}_{k_n}|\mathbf{z}_{k_{n-1}})P_t(\mathbf{z}_{k_n}), \tag{3}$$

where $P(\mathbf{T})$ is modeled as a Markov Chain. $P_s(\mathbf{z}_{k_0})$ is the initialization probability of a track starting at $\mathbf{z}_{k_0}$, $P_t(\mathbf{z}_{k_n})$ is the termination probability of a track ending at $\mathbf{z}_{k_n}$, and $P_l(\mathbf{z}_{k_j}|\mathbf{z}_{k_i})$ is the transition density from measurement $\mathbf{z}_{k_i}$ to measurement $\mathbf{z}_{k_j}$. $P(\mathbf{z}_i|\mathcal{T})$ denotes the likelihood function of measurement $\mathbf{z}_i$, which represents a measurement being a true target or a false alarm. In this paper, all measurements are regarded as targets; this means that $P(\mathbf{z}_i|\mathcal{T}) = 1$, and the posterior probability of trajectory set $\mathcal{T}$ is calculated as follows:

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} \prod_{\mathbf{T}\in\mathcal{T}} P(\mathbf{T}). \tag{4}$$

To take advantage of the concept of network flow in Network Optimization [22], the indictor variable $f_{i,j}$ is defined as the directed flow variable that from measurement $\mathbf{z}_i$ to measurement $\mathbf{z}_j$. $f_{s,i}$ and $f_{i,t}$ represent the starting flow variable and terminated flow variable, respectively. Depending on the flow conversation method [22], for all nodes $\mathbf{z}_i$, the sum of flows arriving at node $\mathbf{z}_i$ is equal to the sum of outgoing flows from node $\mathbf{z}_i$. For any track $\mathbf{T}_k$, it is satisfies the following equation:

$$f_{s,i} + \sum_j f_{j,i} = \sum_i f_{i,j} + f_{i,t}. \tag{5}$$

Moreover, the cost flow network must guarantee that only one node is represented by a target in a moment. Let the upper bound of the sum of outgoing flows from node $\mathbf{z}_i$ is 1. For any node, the constraint is

$$\forall \mathbf{z}_i, \mathbf{z}_j \quad \sum f_{i,j} \leq 1. \tag{6}$$

Taking into account that the target may appear or disappear from any location in the cost flow network, the source and sink node are introduced in [18], which is connected to each node, respectively. We transform the Network Optimization problem of Equation (4) into the IP problem, and the logarithm of the objective function can be rewritten as

$$\begin{aligned} \mathcal{T}^* &= \arg\min_{\mathcal{T}} \sum_{\mathbf{T}_k\in\mathcal{T}} -\log P(\mathbf{T}_k) \\ &= \arg\min_{\mathcal{T}} \sum_{\mathbf{T}_k\in\mathcal{T}} (c_{s,i}f_{s,i} + \sum_j c_{i,j}f_{i,j} + c_{j,t}f_{j,t}) \\ &= \arg\min_{\mathcal{T}} \sum_i c_{s,i}f_{s,i} + \sum_{i,j} c_{i,j}f_{i,j} + \sum_j c_{j,t}f_{j,t}, \end{aligned} \tag{7}$$

where $c_{s,i}$ is the cost of the flow from the source node to measurement $\mathbf{z}_i$, $c_{i,j}$ is the cost of the flow from measurement $\mathbf{z}_i$ to measurement $\mathbf{z}_j$, and $c_{j,t}$ is the cost of the flow from measurement $\mathbf{z}_j$ to the sink node. Figure 1 shows an example of the cost flow network. The IP problem of multitarget tracking with measurement uncertainty can be described as

$$\min \quad \sum_i c_{s,i} f_{s,i} + \sum_{i,j} c_{i,j} f_{i,j} + \sum_j c_{j,t} f_{j,t} \tag{8}$$

$$\begin{aligned} \text{s.t.} \quad & \forall \mathbf{z}_i, \mathbf{z}_j, \quad \sum f_{i,j} \leq 1 \\ & \forall \mathbf{z}_i, \mathbf{z}_j, \quad f_{i,j} \geq 0. \end{aligned} \tag{9}$$

The cost can be defined as follows:

$$\begin{cases} c_{s,i} = -\log P_s(\mathbf{z}_i), \\ c_{i,j} = -\log P_l(\mathbf{z}_j | \mathbf{z}_i), \\ c_{j,t} = -\log P_t(\mathbf{z}_j). \end{cases} \tag{10}$$

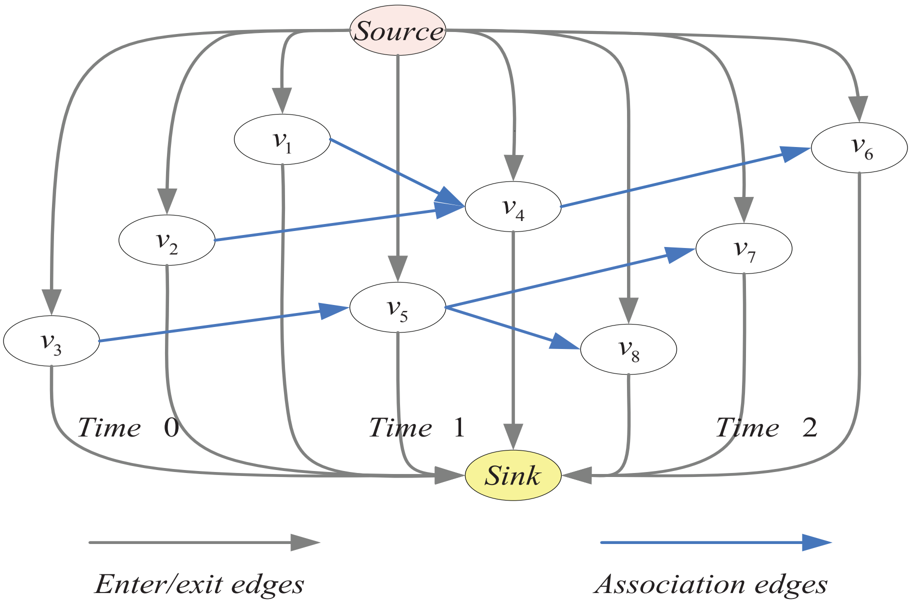

Figure 1. An example of cost flow model with three time steps.

For the IP problem of multitarget tracking, the traditional algorithms have a relatively higher computational cost due to the large number of network nodes. Hence, a parallel processing technology based on the A* search algorithm is presented to find the optimal solution.

3. Description of the A* Search Algorithm

The A* search algorithm is a heuristic graph search method which guides the search process by using the characteristic information of the problem. For the min-cost flow problem, the A* algorithm searches from the origin node, calculates and estimates the cost of each extended node, chooses the extended node that has a minimum cost and stops it when the algorithm reaches the destination node. Assuming that an evaluation function $f^*(x)$ is designed to estimate the minimum cost from origin node s_o through node x to destination node s_d. The estimated minimum cost is calculated as follows:

$$f^*(x) = g^*(x) + h^*(x), \tag{11}$$

where $g^*(x)$ is the cost from origin node s_o to the node x, $h^*(x)$ is the lower bound on the minimum cost from node x to destination node s_d, and $h(x)$ is the actual value from the node x to destination node

s_d, $h^*(x) \leq h(x)$. In order to ensure the optimality of the algorithm, Admissibility and Consistency conditions must be satisfied [23]. Admissibility condition: $f^*(x)$ never overestimates the true cost of a solution along the current path through. Consistency condition: A heuristic function $h(x)$ is consistent if, for every node x and every successor x' of x, the estimated cost of reaching the goal from x is no greater than that of the step cost of getting to x' form x plus the estimated cost of reaching the goal from x'

$$h^*(x) + e(x, x') \leq h^*(x'). \tag{12}$$

In the implementation of the A* search algorithm, two lists need to be built, named Open list and Close list. Open list is the set of nodes that have been calculated, and that are candidates for the selection of the next node. Close list is the set of nodes that have been selected, and that are not in Open list. In the initial stage, Open list contains a source node and Close list is empty. During the iterative process, the A* search algorithm calculates the evaluation function of each node in Open list chooses the node with the minimum cost and judges whether it is the termination node. If so, the algorithm is done. Otherwise, it extends all adjacent nodes and calculates the cost function of each node. If the solution exists, the A* search algorithm can guarantee obtaining the optimal solution [24].

The A* search algorithm can be expressed as follows: here, the Open list and Close list are denoted as O and C. E is the set of edges:

Step 1 Initialization:

Set $x_i = x_s$, $f^*(x_i) = 0$; $f^*(x_j) = \infty$, $g^*(x_j) = \infty$, $\forall x_j \neq x_i$; $O = \{x_i\}$, $C = O$.

Step 2 Node Selection:

Choose $x_i \in Argmin_{x_j \in O} f^*(x_j)$, $C = C \bigcup \{x_i\}$, $O = O \backslash \{x_i\}$.

Step 3 Stop Rule:

If $x_i = x_d$, then stop. otherwise, continue.

Step 4 Update $f^*(x_j)$ and $g^*(x_j)$:

For each $x_j \in E(x_i)$: If $g^*(x_i) + e(x_i, x_j) + h^*(x_j) < f^*(x_j)$, then $g^*(x_j) = g^*(x_i) + e(x_i, x_j)$; $f^*(x_j) = g^*(x_i) + e(x_i, x_j) + h^*(x_j)$.

If $x_j \notin O$, $O = O \bigcup \{x_j\}$. Go back to Step 2.

4. Multitarget Tracking Algorithm Based on Adaptive Network Graph Segmentation

Under the multitarget tracking environment, the A* search algorithm have two obvious defects that long running time and large storage space. To solve the problem, an adaptive spectral clustering algorithm is presented to segment the cost flow network, and the A* search algorithm is used to find the optimal track of segmented sub-graph. To take advantage of track mosaic technology, the optimal track of each sub-graph is combined, and the combined track is smoothed by the Rauch–Tung–Striebel smoother. The flow chart of the proposed algorithm is depicted in Figure 2.

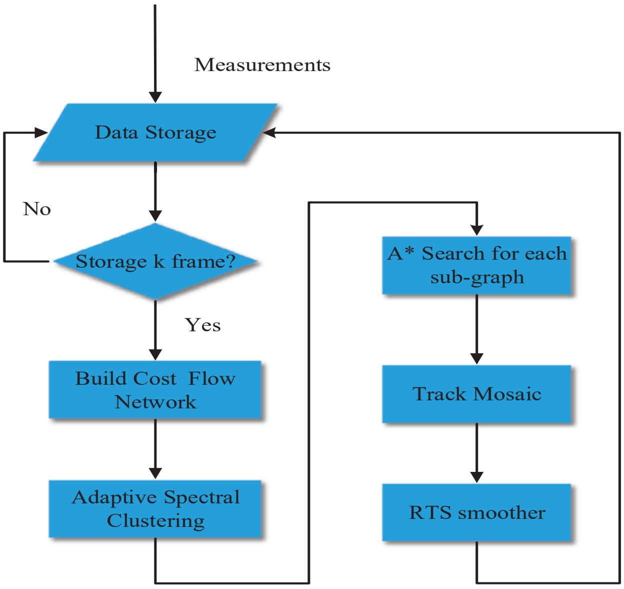

Figure 2. The flow chart of multitarget tracking algorithm based on adaptive network graph segmentation.

4.1. Adaptive Spectral Clustering

The segmentation problem of graph structure represented by the dissimilarity degree between nodes is the combinatorial optimization problem, which is NP-hard. The general solution is to consider the continuous relaxation form for this problem. Spectral clustering is an unsupervised learning method based on graph theory [25] for arbitrary image shapes. It uses the eigenvalue decomposition of graph matrix to build a spectral mapping space of the original data set, and the new space is partitioned by the K-means algorithm.

Let $G_1(\mathbf{V},\mathbf{E},\mathbf{W})$ be an undirected graph that transforms from the directed graph $G_0(\mathbf{V}_0,\mathbf{E}_0)$ with vertex set $\mathbf{V}=(v_1,v_2,\ldots,v_N)$ and edge set $\mathbf{E}=(e_{ij})_{i,j=1,2,\ldots,m}$. We assume that the edge-weighted adjacency matrix of undirected graph $\mathbf{W}=(w_{ij})_{i,j=1,2,\ldots,m}$, which is also called affinity matrix, is nonnegative, $w_{ij}=w_{ji}\geq 0$. If $w_{ij}=0$ that means the node v_i and v_j are not connected. Here, the weight of two nodes w_{ij} is the cost value c_{ij} in the directed graph model $G_0(\mathbf{V}_0,\mathbf{E}_0)$. The sets $\Psi_1....\Psi_k$ are the subsets of the graph. For sets $\Psi_1....\Psi_k$, $\Psi_1\bigcup\Psi_2..\bigcup\Psi_k=\mathbf{V}$ and $\Psi_i\bigcap\Psi_j=\varnothing, i\neq j$. Let $cut(\Psi_1,\Psi_2,\ldots,\Psi_k)$ be the sum of the cuts between sets $\Psi_1,\Psi_2,\ldots,\Psi_k$:

$$cut(\Psi_1,\Psi_2,\ldots,\Psi_k)=\sum_{i=1}^{k}cut(\Psi_i,\bar{\Psi}_i), \tag{13}$$

where $\bar{\Psi}_i$ denotes the complement $\mathbf{V}\backslash\Psi_i$. The purpose of the spectral clustering is to find the sets $\Psi_1,\Psi_2,\ldots,\Psi_k$ such that $MNcut$ is minimized. The objective function is given as follows:

$$MNcut=\sum_{i=1}^{n}\frac{cut(\Psi_i,\bar{\Psi}_i)}{\sum\limits_{u\in\Psi_i,v\in\bar{\Psi}_i}w_{u,v}}. \tag{14}$$

In [26], eigenvectors are clustered in the subspace that is generated by the first k eigenvectors of normalized Laplacian matrix. The degree matrix of the graph $\mathbf{D}$ and the normalized Laplacian matrix $\mathbf{L}_{sym}$ are defined as

$$\mathbf{D}=\begin{cases}\mathbf{D}_{ii}=\sum\limits_{j}w_{ij},\\ \mathbf{D}_{ij}=0 \quad i\neq j,\end{cases} \tag{15}$$

$$\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}. \tag{16}$$

Unfortunately, **W** grows as the square of the number of elements in the grouping problem, and it quickly becomes infeasible to fit **W** in memory. Hence, an adaptive spectral clustering based on the Nyström Method is proposed to reduce the complexity of the time and space. The Nyström Method is a technique for finding numerical approximations to eigenfunction problems.

Assuming that randomly sample n points from vertex set **V** and the number of the remaining points is $N - n$. Now, partition the affinity matrix **W** as

$$\mathbf{W} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}, \tag{17}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ represents the sub-block of weights among the random samples, $\mathbf{B} \in \mathbb{R}^{(N-n)\times n}$ contains the weights between the sample points and the rest of points, and $\mathbf{C} \in \mathbb{R}^{(N-n)\times(N-n)}$ denotes the weights matrix between all of the remaining points. Let $\bar{\mathbf{U}}$ represent the approximate eigenvectors of **W**:

$$\bar{\mathbf{U}} = \begin{bmatrix} \mathbf{U} \\ \mathbf{B}^T\mathbf{U}\Lambda^{-1} \end{bmatrix}. \tag{18}$$

The approximation of **W**, which we denote $\hat{\mathbf{W}}$, can be written as

$$\begin{aligned} \hat{\mathbf{W}} &= \bar{\mathbf{U}}\Lambda\bar{\mathbf{U}}^T \\ &= \begin{bmatrix} \mathbf{U} \\ \mathbf{B}^T\mathbf{U}\Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} \mathbf{U}^T & \Lambda^{-1}\mathbf{U}^T\mathbf{B} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \end{bmatrix}. \end{aligned} \tag{19}$$

From Equations (17) and (19), **C** is approximated by $\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$. Therefore, calculating the affinity matrix between remaining points is avoided. It is noteworthy that the columns of $\bar{\mathbf{U}}$ are not orthogonal. We need to orthogonalize $\bar{\mathbf{U}}$. If **A** is a positive definite matrix, $\mathbf{A}^{-1/2}$ represents the symmetric positive definite square root of **A**. Let $\mathbf{Q} = \mathbf{A} + \mathbf{A}^{-1/2}\mathbf{B}\mathbf{B}^T\mathbf{A}^{-1/2}$, and diagonalize **Q** as $\mathbf{Q} = \mathbf{U}_s\Lambda_s\mathbf{U}_s^T$. If the matrix $\mathbf{U}_v$ is defined as

$$\mathbf{U}_v = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^T \end{bmatrix} \mathbf{A}^{-1/2}\mathbf{U}_s\Lambda_s^{-1/2}, \tag{20}$$

then the affinity matrix W is diagonalized by $\mathbf{U}_v$ and Λ_s. Without calculating $\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$, a simple approach is proposed to calculated the row sums of $\hat{\mathbf{W}}$:

$$\hat{\mathbf{d}} = \begin{bmatrix} \mathbf{a}_r & \mathbf{b}_r \\ \mathbf{b}_c & \mathbf{B}^T\mathbf{A}^{-1}\mathbf{b}_r \end{bmatrix}, \tag{21}$$

where $\mathbf{a}_r, \mathbf{b}_r \in \mathbb{R}^{N-n}$ denote the rows sum of **A** and **B**. $\mathbf{b}_c \in \mathbb{R}^n$ denotes column sum of **B**. The normalized **A** and **B** can be obtained by $\hat{\mathbf{d}}$. The elements of the normalized **A** and **B** are given by

$$\mathbf{A}_{ij} = \frac{\mathbf{A}_{ij}}{\sqrt{\hat{\mathbf{d}}_i\hat{\mathbf{d}}_j}}, i, j = 1, \ldots, n, \tag{22}$$

$$\mathbf{B}_{ij} = \frac{\mathbf{B}_{ij}}{\sqrt{\hat{\mathbf{d}}_i\hat{\mathbf{d}}_{j+N}}}, i = 1, \ldots, n, j = 1, \ldots, N - n. \tag{23}$$

The number of clusters is generally determined by human experience and background knowledge. Next, the relationship between the spectrum of the weight matrix and the number of clusters can be obtained by analyzing the affinity matrix of graph.

If v_i and v_j belong to the same class, then $w_{ij} = 1$, otherwise $w_{ij} = 0$. According to perturbation analysis of spectral clustering [27], a permutation matrix $\mathbf{P}$ always exists to make elements of any node set $\mathbf{V}$ in the sequence belongs to a class after the $\mathbf{PV}$ transformation. The affinity matrix $\mathbf{W}$ is a block diagonal matrix that consists of n_k all-1 matrices. Thus, the elements of Laplacian matrix $\hat{\mathbf{L}}_{sym}$ is divided into k matrix blocks. For each matrix block $\hat{\mathbf{L}}_{n_i}$, $\lambda_k = n_i$, $k = 1, 2, \ldots n_i - 1$, where λ_k is the eigenvalues of matrix block $\hat{\mathbf{L}}_{n_i}$, and $\lambda_{n_i} = 0$. In light of Matrix Theory [28], the union of the eigenvalues of real symmetric matrix equals that of the block diagonal matrix that consists of these real symmetric matrices. Therefore, the eigenvalues of Laplacian matrix is the union of the eigenvalues of k matrices. The eigenvalues of Laplacian matrix consists of $(n-k)(n_i-1)$ nonzero eigenvalues and k zero eigenvalues. The number of clusters is the number of zero eigenvalues of Laplacian matrix $\hat{\mathbf{L}}_{sym}$.

4.2. k-Short Paths Algorithm

In the cost flow network of multitarget tracking, there may be multiple paths that are directed and paths are edge- and node- disjoint, which means that any two paths cannot share the same edge and node, and a path visits a node in the sub-graph at most once. Here, a path represents a possible track. We reformulated the multitarget tracking problem as an edge- and node- disjoint k-shortest paths problem on a directed acyclic graph. In order to obtain the k-shortest paths, the segmented graph is transformed into a undirected graph firstly, and then the A* search algorithm is adopted to find the single shortest path. If the maximum iteration count is not reached, remove nodes except source node and sink node on the single shortest path, and search for the next path until reaching the maximum iteration count.

4.3. Track Mosaic

To segment an undirected graph in the multitarget tracking environment using the adaptive spectral clustering algorithm, it may arise over segmentation. For example, a trajectory may be divided into several segments. In order to obtain the integral track, the mosaic technique is employed to deal with these segmented trajectories. Let $\mathbf{T}_i$ and $\mathbf{T}_j$ are two trajectories of different sub-graphs. x_{io} and x_{jo} are the initial position, and x_{id} and x_{jd} are the terminal position, respectively. The Euclidean distance d_D is used to decide whether to mosaic two trajectories. If $d_D < \tau$, $\mathbf{T}_i$ and $\mathbf{T}_j$ are mosaicked, where τ is the mosaic threshold.

4.4. Rauch–Tung–Striebel Smoother

A Rauch–Tung–Striebel (RTS) smoother [29] is used to smooth the extracted tracks. The RTS smoother consists of two parts. The first part is the Kalman filter, which calculates the state of target at each time and estimates the corresponding covariance matrix. The second part is backward recursion [30]. In this process, target state and the covariance matrix are taken as inputs to obtain the smoother output.

4.5. Time Complexity

The proposed algorithm is parallel processing in that each sub-graph uses the A* search algorithm to obtain the shortest path. The time complexity of the proposed algorithm is mainly composed of two parts, which are the time complexity of the adaptive spectral clustering algorithm and the worst time complexity of searching sub-graph. The time complexity of the adaptive spectral clustering algorithm is $O(n \times (N-n)) + O(n^3) + O(KNI)$. $O(n \times (N-n))$, $O(n^3)$ and $O(KNI)$ are the time complexity of calculating degree $\hat{\mathbf{d}}$, orthogonal eigenvectors $\mathbf{U}_v$ and K-means clustering algorithm, respectively. N is the number of nodes in the undirected graph, n is the sample points, K is the number of trajectories and I is the iteration number of K-means algorithm. The worst time complexity of searching sub-graph is $O(n_{\max}^2)$, and that $n_{\max}$ is less than N. The proposed algorithm is adopted to calculate the k-shortest path in $O(n \times (N-n)) + O(n^3) + O(KNI)$. The A* search algorithm, SPFA and SSP are recognized as three effective methods to solve the k-shortest path problem. The time complexity of these algorithms are $O(N^2)$, $O(NE)$ and $O(KN^2)$, respectively. E is the number of edge in the undirected graph. The complexity of these algorithms are primarily related to the value of K and N. For multitarget tracking systems, that is, N is large, and the time complexity of the proposed algorithm is far less than that of the algorithms mentioned above.

5. Experimental Results

In this section, the proposed algorithm was tested in different tracking scenarios. The optimal subpattern assignment (OSPA) [31] metric is used for performance assessment. The experiments have been performed on a computer with an Intel G840 2.8 GHz CPU and 4 GB of memory.

5.1. Clustering Quality Evaluation

To evaluate the clustering quality of the adaptive spectral clustering algorithm, the external quality measure (F-score) [32] and (MCR) [33] are used. F-score is used in spam detection for documentation as an overall assessment performance that combines the precision and recall ideas from information retrieval. F-score is defined as follows:

$$\text{F-score} = \sum_{i=1}^{K} (N_i \times \frac{2P_f(i,j)R_f(i,j)}{P_f(i,j) + R_f(i,j)})/N, \tag{24}$$

where $P_f(i,j) = N_{ij}/N_j$ represents the precision of the cluster j for the given class i. $R_f(i,j) = N_{ij}/N_i$ represents the recall of the cluster j for the given class i. N_i is the number of the members of class i. N_{ij} is the number of numbers of class i in cluster j. The MCR is given by

$$MCR = \frac{C_m}{C_t}, \tag{25}$$

where C_m is the number of misclassified targets. C_t is the total number of targets.

Figure 3 displays the comparison of classification results with different sampling rates. Since the sampling rate is 1%, clustering result is seriously affected. When the sampling rate exceeds 1%, it is clear that the classification results outperform that shown in Figure 3b. The clustering performance versus sampling rate are shown in Figure 4. It can be noted that F-score increases with increasing sample rate. Once the sample rate exceeds 30%, the data set can be accurately segmented. This is because the similarity matrix between the sampling points can be approximated by that of all points. In Figure 4b, the sampling rate is equal to 1%, MCR is 0.18. The MCR of other sampling rate is equal to 0. Based on the two evaluation criteria above, we choose the sampling rate as 10% in this paper.

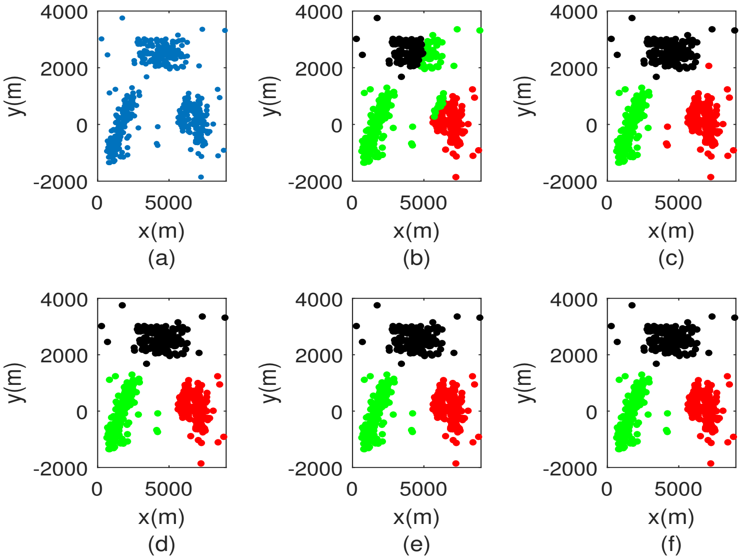

Figure 3. Comparison of classification results. (**a**) original data set; (**b**) sampling rate is 1%; (**c**) sampling rate is 10%; (**d**) sampling rate is 50%; (**e**) sampling rate is 70%; (**f**) sampling rate is 100%.

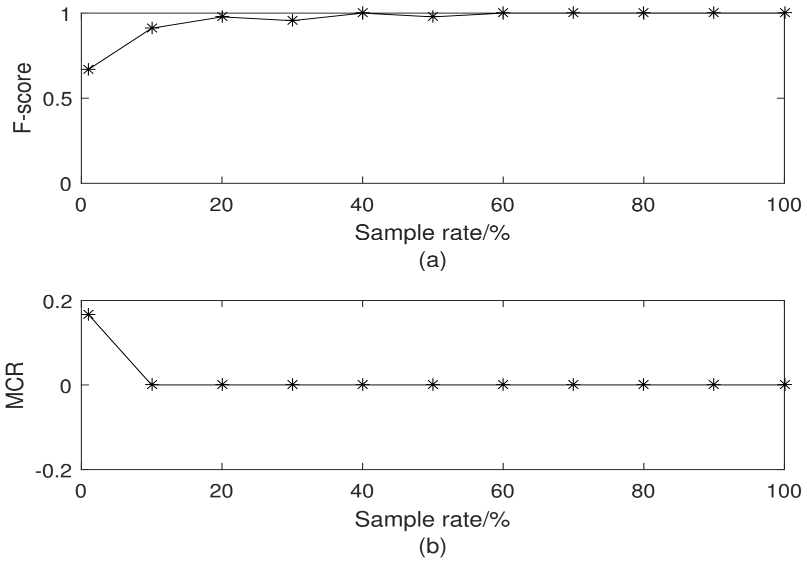

Figure 4. Clustering performance versus sampling rate. (**a**) F-score versus sampling rate; (**b**) MCR versus sampling rate.

5.2. Performance Analysis

In this subsection, we consider two scenarios for multitarget tracking. There are two types of dense clutter areas. Inside Type I dense clutter area, clutter points are uniformly distributed in the surveillance region. While Type II dense clutter area is elliptical and the position of its clutter points follows a 2D Gaussian distribution, whose mean is target position at k time and the standard deviations are determined by the major axes of the ellipse. The measurements are obtained from radar which located at [0, 0] m. The measurement model is described as

$$\mathbf{z}_t^i = \mathbf{H}\Phi_t^i + v_t^i, \tag{26}$$

where $\Phi_t^i = [\phi_t^i, \dot{\phi}_t^i, \varphi_t^i, \dot{\varphi}_t^i]^T$ is the state variable. The measurement noise v_t^i is zero-mean Gaussian random vector with covariance matrix $\mathbf{R} = diag([\delta^2, \delta^2])$, and $\delta = 10$ m in the scenarios 1 and 2.

The target motion can be modeled by combination of constant turn rate (CT) motion and constant velocity (CV) motion [34]. The dynamistic model of the target is described as follows:

$$\Phi_t^i = \mathbf{F}_t^i \Phi_{t-1}^i + w_t^i, \tag{27}$$

where $\mathbf{F}_t^i$ is the state transition matrix of the target i at the time t. Under the assumption of CV motion, it is defined as follows:

$$\mathbf{F}_t^i = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{28}$$

where T denotes the sampling time. In CT motion, it is defined as

$$\mathbf{F}_t^i = \begin{bmatrix} 1 & \dfrac{\sin(\omega_t^i T)}{\omega_t^i} & 0 & \dfrac{\cos(\omega_t^i T) - 1}{\omega_t^i} \\ 0 & \cos(\omega_t^i T) & 0 & -\sin(\omega_t^i T) \\ 0 & \dfrac{1 - \cos(\omega_t^i T)}{\omega_t^i} & 1 & \dfrac{\sin(\omega_t^i T)}{\omega_t^i} \\ 0 & \sin(\omega_t^i T) & 0 & \cos(\omega_t^i T) \end{bmatrix}, \tag{29}$$

where ω_t^i denotes the turn rate. The process noise w_t^i is zero-mean Gaussian random vector with covariance matrix

$$\mathbf{Q} = \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix}. \tag{30}$$

Scenario 1:

A multiple non-crossing tracking scenario is considered in the surveillance region [0, 9000] m × [−2000, 5000] m. There are four manoeuvring targets whose initial position are [8000, 2500] m, [500, 3000] m, [2500, 2000] m and [2500, −1500] m, respectively. Their initial velocities are [−80, −80] m/s, [50, −130] m/s, [0, 280] m/s and [150, 0] m/s, respectively. The initial and terminal time are [1, 1, 3, 1] s and [27, 30, 30, 28] s, respectively. The mosaic threshold $\tau = 10$.

In this case, the Type I clutter intensity is $\kappa_1(z_c) = \frac{m_{Ic}}{V_I} = \frac{5}{6.3 \times 10^7} = 0.79 \times 10^{-7}$, where m_{Ic} is the expected number of clutter measurements in this Type I dense clutter area, and V_I is the surveillance region. For the Type II dense clutter, the expected number of clutter measurements in this Type II dense clutter area is 10, and $V_{II} = 4.54 \times 10^5$ m^2 is the 'volume' of the Type II dense clutter surveillance region. Therefore, the Type II clutter intensity is $\kappa_2(z_c) = \frac{m_{IIc}}{V_{II}} = \frac{10}{\pi \times 600 \times 400} = 1.33 \times 10^{-5}$. We perform a total of 100 Monte-Carlo runs to obtain the average optimal subpattern assignment (OSPA) distance [35] and average of the estimated number of targets.

Figure 5 shows the average OSPA distances of the proposed algorithm, the A* search algorithm, SSP and SPFA. It is observed that the average OSPA distance of the proposed algorithm is approximately equal to that of the A* search algorithm, which are better than that of SPFA and SSP. Figure 6 shows an average of the estimated number of targets of the proposed algorithm, the A* search algorithm, SSP and SPFA. It can be seen that the proposed algorithm presents an considerable performance in the estimation of the target numbers. To make a comparison, set the Type II clutter intensity $\kappa_2(z_c) = 1.33 \times 10^{-6}$, 6.33×10^{-6}, 19.89×10^{-6}, 26.53×10^{-6}, 39.79×10^{-6}, respectively. The average OSPA distances of the proposed algorithm, the A* search algorithm, SSP and SPFA versus the Type II clutter intensity are shown in Figure 7.

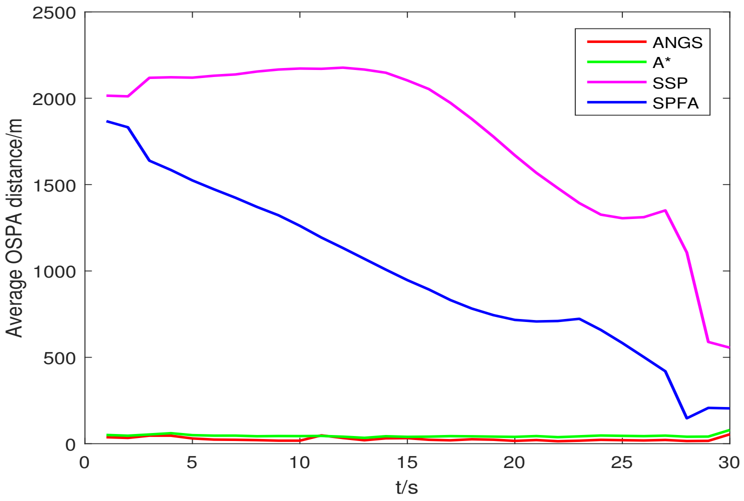

Figure 5. The average OSPA distance in scenario 1, with $c = 10$ and $p = 2$.

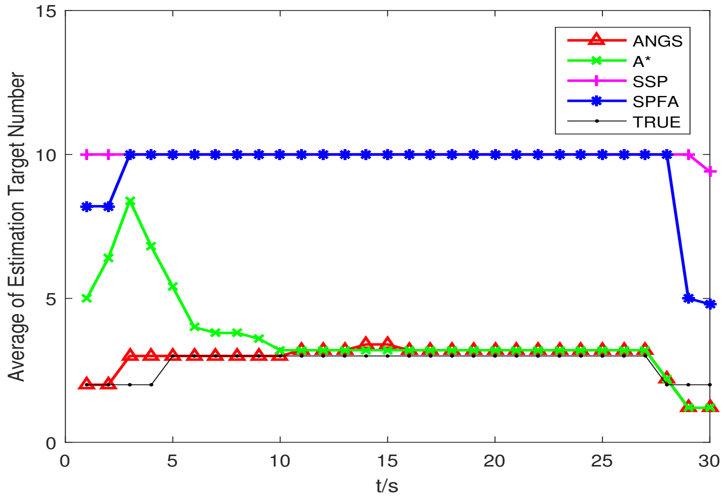

Figure 6. The average of the estimated numbers of targets in scenario 1.

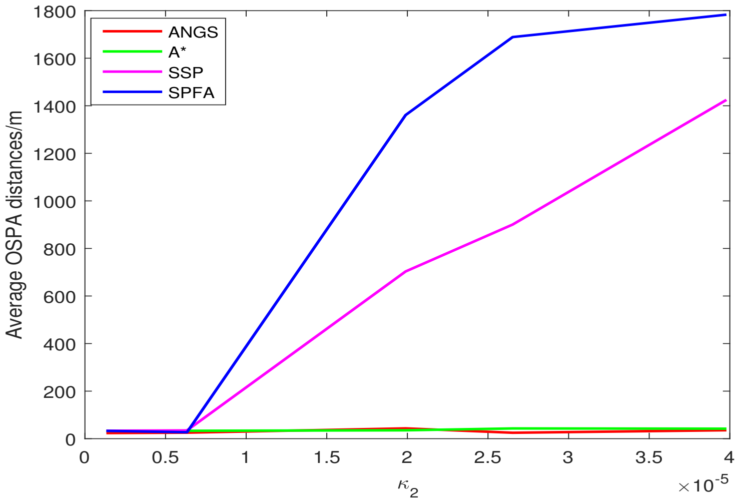

Figure 7. The average OSPA distances versus the Type II clutter intensity in scenario 1.

The average OSPA distances of the proposed algorithm have a relatively small difference than that of other algorithms at $\kappa_2(z_c) = 1.33 \times 10^{-6}$. The average OSPA distance of other algorithms increase rapidly with increasing the clutter intensity. The average OSPA distance of the proposed algorithm is basically no significant change, still at a lower value. When $\kappa_2(z_c) = 39.79 \times 10^{-6}$, it is obvious that the average OSPA distances of SSP and SPFA is about 40 and 50 times of that of the proposed algorithm, respectively.

Scenario 2:

An unknown and time-varying multiple crossing targets scenario is considered in the surveillance region [0, 10,000] m × [−4000, 5000] m. There are four manoeuvring targets whose initial positions are [7000, 4500] m, [3500, 4000] m, [1000, −500] m and [7000, −2500] m, respectively. Their initial velocities are [0, −150] m/s, [200, −20] m/s, [150, 300] m/s and [150, 300] m/s, respectively. The initial and terminal time are [1, 5, 1, 7] s and [27, 30, 30, 27] s, respectively. The mosaic threshold $\tau = 10$.

Trajectories intersect at [7478, 3618] m and [4443, −359] m. The average OSPA distances and the average of the estimated numbers of targets are shown in Figures 8 and 9. The average OSPA distances of the proposed algorithm, the A* search algorithm, SSP and SPFA are given in Figure 10. The Type I clutter intensity is $\kappa_1(z_c) = \frac{5}{9\times10^7} = 0.56 \times 10^{-7}$, and the Type II clutter intensity are $\kappa_2(z_c) = 1.33 \times 10^{-6}$, 6.33×10^{-6}, 19.89×10^{-6}, $gaihao 26.53 \times 10^{-6}, 39.79 \times 10^{-6}$, respectively.

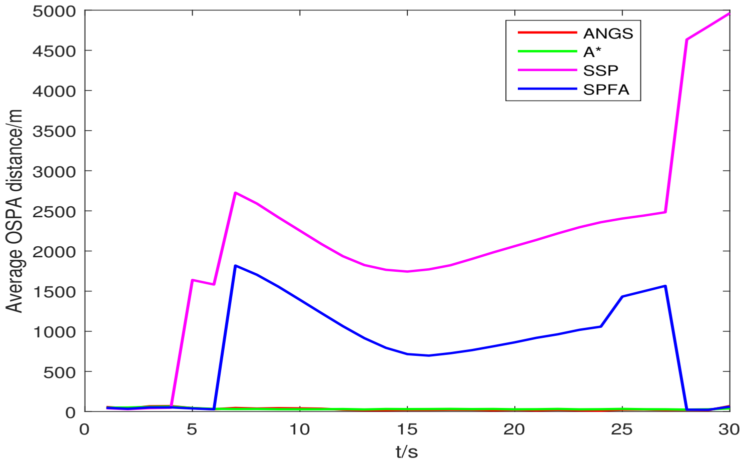

Figure 8. The average OSPA distance in scenario 2, with *c* = 10 and *p* = 2.

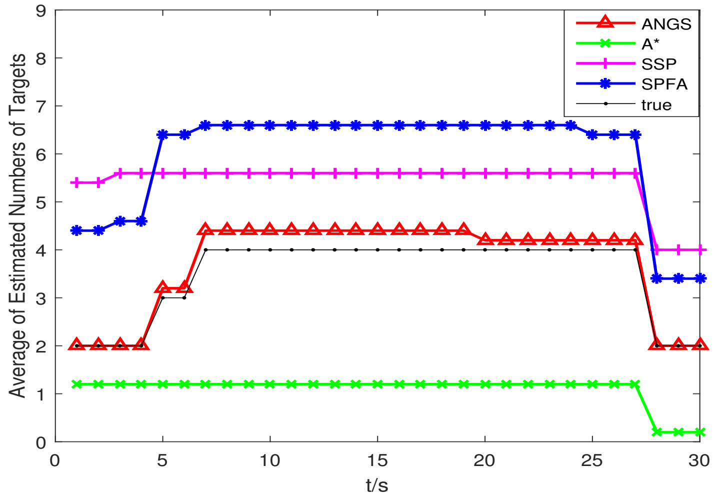

Figure 9. The average of the estimated numbers of targets in scenario 2.

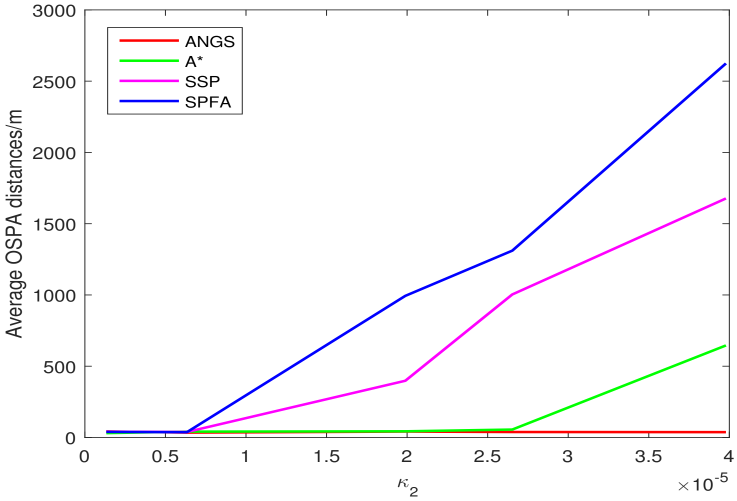

Figure 10. The average OSPA distances versus the Type II clutter intensity in scenario 2.

As seen from Figure 8, the SSP and SPFA have a larger average OSPA distances. In Figure 9, it is apparent that the average target number estimation of the proposed algorithm is exactly the same as the number of the true targets. In Figure 10, as expected, there is an overall increase of OSPA distances with with increasing the clutter intensity. The average OSPA distance of the proposed algorithm is at a lower value. When $\kappa_2(z_c) = 39.79 \times 10^{-6}$, the average OSPA distances of SSP and SPFA is about 45 and 70 times of that of the proposed algorithm, respectively.

5.3. Run Time

The average running time of different Type II clutter intensity is shown in Figure 11. It clear that the running time of the A* search algorithm, SSP, and SPFA increase exponentially. The running time of the proposed algorithm is growing slowly. When $\kappa_2(z_c) = 2.222 \times 10^{-7}$, the running time of the A* search algorithm is about 14 times that of the proposed algorithm.

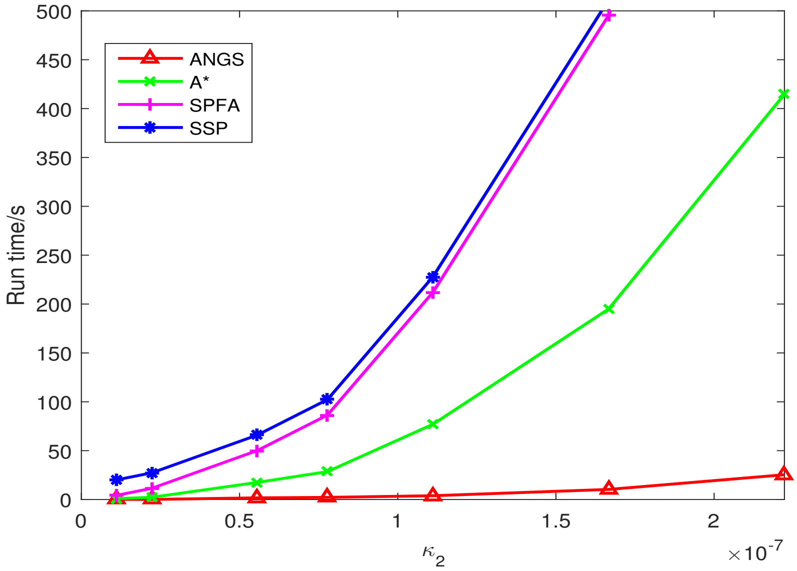

Figure 11. The comparison of average running time.

6. Conclusions

In this paper, we have presented a novel data association framework for multitarget tracking with measurement uncertainty that estimates unknown number and states of targets using the continuous multi-frame data. The multitarget tracking problem was formulated as network flow optimization problem for finding *k*-shortest paths, and an adaptive spectral clustering algorithm was used to segment the network structure. The optimal solution of each sub-graph can be obtained by the A* search algorithm. Experiment results indicate that the proposed algorithm is helpful in improving the accuracy of track extraction and can reduce the computational complexity. Future work will focus on tracking multitargets with low detection probability.

Author Contributions: T.M. and S.G. conceived and designed the algorithm; T.M. performed the experiments, analyzed the data, and drafted the paper; C.C. and X.S. revised the manuscript.

Funding: This research work was supported by the National Key Research and Development Program of China (Grant No. 2016YFE0111900).

Acknowledgments: We would like to thank the support received from Yuyan Cao and Xinmin Wang.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Challa, S.; Morelande, M.R.; Mušicki, D.; Evans, R.J. *Fundamentals of Object Tracking*; Cambridge University Press: Cambridge, UK, 2011.
2. Mallick, M.; Arulampalam, S.; Yan, Y.; Ru, J. Three-Dimensional Tracking of an Aircraft Using Two-Dimensional Radars. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 585–600. [CrossRef]
3. Rodriguez, D.; Aceros, C.; Valera, J.; Anaya, E. A Framework for Multiple Object Tracking in Underwater Acoustic MIMO Communication Channels. *J. Sens. Actuator Netw.* **2017**, *6*, 2. [CrossRef]
4. Hoak, A.; Medeiros, H.; Povinelli, R.J. Image-Based Multi-Target Tracking through Multi-Bernoulli Filtering with Interactive Likelihoods. *Sensors* **2017**, *17*, 501. [CrossRef] [PubMed]
5. Banfi, J.; Guzzi, J.; Amigoni, F.; Flushing, E.F.; Giusti, A.; Gambardella, L.; Caro, G.A.D. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Auton. Robots* **2018**, 1–16. [CrossRef]
6. Beard, M.; Vo, B.T.; Vo, B.N. A Solution for Large-scale Multi-object Tracking. *arXiv* **2018**, arXiv:1804.06622.
7. Chen, H.; Li, X.R.; Bar-Shalom, Y. On joint track initiation and parameter estimation under measurement origin uncertainty. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *40*, 675–694. [CrossRef]
8. Huang, Y.; Zhang, Y.; Shi, P.; Wu, Z.; Qian, J.; Chambers, J.A. Robust Kalman Filters Based on Gaussian Scale Mixture Distributions with Application to Target Tracking. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, 1–15. [CrossRef]
9. Oh, S.; Russell, S.; Sastry, S. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Trans. Autom. Control* **2009**, *54*, 481–497.
10. Vo, B.N.; Ma, W.K. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Trans. Signal Process.* **2006**, *54*, 4091–4104. [CrossRef]
11. Vo, B.N.; Vo, B.T.; Phung, D. Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter. *IEEE Trans. Signal Process.* **2014**, *62*, 6554–6567. [CrossRef]
12. Bar-Shalom, Y.; Li, X.R. Multitarget-Multisensor Tracking: Principles and Techniques. *IEEE Aerosp. Electron. Syst. Mag.* **1995**, *11*, 41–44.
13. Jia, B.; Pham, K.D.; Blasch, E.; Chen, G. Multiple space object tracking via a space-based optical sensor. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–10.
14. Wang, F.; Man, Y.; Man, L. Intelligent optimization approach for the k shortest paths problem based on genetic algorithm. In Proceedings of the International Conference on Natural Computation, Xiamen, China, 19–21 August 2014; pp. 219–224.
15. Chong, C.Y. Graph approaches for data association. In Proceedings of the International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 1578–1585.

16. Ren, X.; Huang, Z.; Sun, S.; Liu, D. An Efficient MHT Implementation Using GRASP. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 86–101. [CrossRef]
17. Goldberg, A.V. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algorithms* **1997**, *22*, 1–29. [CrossRef]
18. Zhang, L.; Li, Y.; Nevatia, R. Global data association for multi-object tracking using network flows. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
19. Pirsiavash, H.; Ramanan, D.; Fowlkes, C.C. Globally-optimal greedy algorithms for tracking a variable number of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Springs, CO, USA, 20–25 June 2011; pp. 1201–1208.
20. Xi, Z.; Liu, H.; Liu, H.; Yang, B. Multiple Object Tracking Using the Shortest Path Faster Association Algorithm. *Sci. World J.* **2014**, *2014*, 481719. [CrossRef] [PubMed]
21. Zhenghao, X.I.; Hongbo, L.I.; Liu, H.; Sun, F. Multiple object tracking using A* algorithm optimization. *Qinghua Daxue Xuebao/J. Tsinghua Univ.* **2014**, *54*, 1549–1554.
22. Ahuja, R.K.; Magnanti, T.L.; Orlin, J.B. *Network Flows: Theory, Algorithms, and Applications*; Prentice Hall: Upper Saddle River, NJ, USA, 1993.
23. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Pearson: London, UK, 2010.
24. Chabini, I.; Lan, S. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 60–74. [CrossRef]
25. Kim, J.; Min, J.; Kweon, I.S.; Lin, Z. Fusing Multiple Independent Estimates via Spectral Clustering for Robust Visual Tracking. *IEEE Signal Process. Lett.* **2012**, *19*, 527–530. [CrossRef]
26. Wang, X.; Qian, B.; Davidson, I. On constrained spectral clustering and its applications. *Data Min. Knowl. Discov.* **2014**, *28*, 1–30. [CrossRef]
27. Vautard, F.; Xu, L.; Drzal, L.T. Study on the Determination and Application of the Cluster Number of the Images Based on the Spectral Graph Theory. *Gongcheng Shuxue Xuebao/Chin. J. Eng. Math.* **2012**, *29*, 27–50.
28. Zhang, F. *Matrix Theory*; Springer: New York, NY, USA, 2011.
29. Särkkä, S. Continuous-time and continuous-discrete-time unscented Rauch–Tung–Striebel smoothers. *Signal Process.* **2010**, *90*, 225–235. [CrossRef]
30. Huang, Y.; Zhang, Y.; Zhao, Y.; Mihaylova, L.; Chambers, J. A Novel Robust Rauch–Tung–Striebel Smoother Based on Slash and Generalized Hyperbolic Skew Student's T-Distributions. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 369–376.
31. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457. [CrossRef]
32. Kristan, M.; Matas, J. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2137–2155. [CrossRef] [PubMed]
33. Zhang, Y.; Brady, M.; Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging* **2001**, *20*, 45–57. [CrossRef] [PubMed]
34. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364.
35. Garcia-Fernandez, A.F.; Morelande, M.R.; Grajal, J. Bayesian Sequential Track Formation. *IEEE Trans. Signal Process.* **2014**, *62*, 6366–6379. [CrossRef]