A Track Association Algorithm Based on Leader-Follower On-line Clustering in Dense Target Environments

Li XU [1], Shuilin JIN [2], Guisheng YIN [1]

[1] College of Computer Science and Technology, Harbin Engineering University, 150001 Harbin, China
[2] Dept. of Mathematics, Harbin Institute of Technology, 150001 Harbin, China

xuli@hrbeu.edu.cn, jinsl@hit.edu.cn, yinguisheng@hrbeu.edu.cn

Abstract. *The imbalance between accuracy and computational cost is a defect in track association. In response to the defect, the track association problem is transformed into an on-line clustering problem with constraints, and a novel track association algorithm is proposed based on Leader-Follower online clustering. In the algorithm, we take a track as a Leader or a Follower based on its type and make Followers and Leaders clustered, which greatly reduces the track pairs associated. In addition, the association relationships between Leaders and Followers are acquired by introducing a function of association degree, which is characterized by small computational cost and no requirements on the distribution of sensor data. The fused Leader-Follower forms a new Leader, which combines Leader generation and track fusion. When sensor tracks are updated, their Leaders will be changed and the other Leaders will be retained, by which the associated results obtain a good stability.*

Keywords

Track association, Leader-Follower on-line clustering, multi-sensor, multi-target.

1. Introduction

In distributed multi-sensor track fusion systems, track association is a core technique. The association accuracy would directly impact on the performance of the whole fusion system.

Over the last few years, a lot of significant works on the track association problem have been done. In particular, many researches focused on the problem in challenging scenarios, such as overlapping coverage scenarios [1], cluttered environments [2,3] and scenarios with temporarily undetectable targets [4]. Besides, scholars put forward many solutions to the track association problem under different conditions. These conditions are much closer to the real world, and the track association problem under these conditions becomes more complex. For instances, Maurer attempted to quantify the effect on associating two independently developed track sets with various types of information [5]. Chen et al. examined several track association methods with different assumptions on the target distribution [6]. Panakkal et al. demonstrated an effective data association scheme for closely moving targets [7]. Ouyang et al. derived a modified cost function for the passive sensor data association [8]. Sigalov et al. applied the cross entropy method and its recent MinxEnt variant to the multi-scan version of the data association problem and obtained state-of-the-art performance in the presence of misdetections, false alarms and unknown number of targets [9]. Han et al. applied the optimal Bayes joint decision and estimation to the track association in the presence of sensor bias [10].

In addition, Kaplan et al. investigated different versions of the likelihood that more than two tracks represented the same target, and compared the performance of all likelihood versions [11]. Papageorgiou et al. derived a closed-form expression for computing "pure" track association likelihoods and presented an alternative formulation of the track association problem, which facilitated system-level track ambiguity management [12]. Sathyan et al. proposed two new assignment-based association algorithms, which improved tracking performance, while requiring considerably less computations [13]. Roy et al. emphasized nearest neighborhood approach for track association, which was carried out after the target motion analysis solution stabilized [14]. These works all contribute to the research of the track association problem.

In multi-sensor environments, if there is only one target in the surveillance region, then track fusion mainly concerns how to obtain precise track state estimations of the target quickly. And if there are multiple targets in the surveillance region, track association is one of the most important problems in track fusion. Especially in dense target environments, if all the local tracks from different sensors are association-judged in pairs, the system burden will be heavy. The track association in dense target environments has become a very challenging and significant task. To solve the problem, a novel track association algorithm is presented based on Leader-Follower on-line clustering in this paper.

2. Track Management

Track association involves two kinds of tracks: system tracks and local tracks.

When the local track of target t from sensor i is updated:

(1) If the database of local tracks is empty or doesn't include the local tracks from sensor i, write sensor i and its local tracks into the database of local tracks.

(2) If the database of local tracks includes the local tracks from sensor i, but doesn't include the local track of target t, which means that sensor i has detected a new target. Write the local track into the database of local tracks under sensor i.

(3) If the database of local tracks includes the local track of target t from sensor i, which means that the state estimations of the local track is updated. Write the updated state estimations to the end of the local track.

(4) If the database of system tracks is empty, then write all local tracks in the database of local tracks into the database of system tracks.

(5) If the local track is judged to be unassociated with any system track, write the local track into the database of system tracks.

(6) If the local track is judged to be associated with a system track, then fuse the local track with the system track, and write the fusion track as a new system track into the database of system tracks, and delete the original system track from the database of system tracks.

As time goes on, the local tracks in the database of local tracks become more and more, but some local tracks will no longer be updated. So it is necessary to remove these local tracks from the database of local tracks. If all the original local tracks of a system track are deleted, then remove the system track from the database of system tracks.

3. Track Association and Clustering

In distributed multi-sensor track fusion systems, each sensor independently processes its local measurements and forms local tracks. The fusion center fuses the local tracks of the same targets and forms system tracks. In fact, track association is to divide the local tracks from different sensors into different groups, so that the local tracks in the same group originate from the same target. Clustering is to classify samples into several categories, so that the samples in the same category have similar attributes. Both methods can be used in information classification.

Through the above analysis, we can understand the track association problem from a new perspective. The local tracks from different sensors can be viewed as samples to be clustered, and the goal is to classify these samples into several categories according to their attributes. In this way, the track association problem is transformed into a clustering problem.

However, for a certain sensor, its local tracks sent to the fusion center are from different targets, so the local tracks from the same sensor cannot belong to a category. Besides, one local track from a sensor is associated with one local track from another sensor at the most, so the track association problem can be viewed as a clustering problem with constrains. The task of track association is to cluster the local tracks from different sensors under the constraints.

For the track association problem, neither the number of targets is known, nor all the track data is got before clustering, which make some clustering algorithms have some defects in solving the track association problem, such as unstable cluster structure. If the arrival of new sample results in a large reconfiguration of the cluster structure, then it will make the problem-solving become more complex. This is due to some clustering algorithms use global standards. The new sample will affect all the cluster centers no matter how far the sample is from the cluster centers. So the Leader-Follower online clustering is adopted in this paper.

4. The Track Association Algorithm Based on Leader-Follower On-line Clustering

The track association algorithm is based on Leader-Follower on-line clustering, abbreviated as Leader-Follower algorithm in the paper.

In the algorithm, each system track is viewed as a Leader, and each local track is viewed as a Follower. Each track is a discrete sequence evolving over time, thus the Leader-Follower on-line clustering will be carried out among the state estimations at each time step.

In the Leader-Follower algorithm, the similarity between the features of Leader-Follow at time k is acquired by an entropy function, whose value range is from 0 to 1. The smaller the value is, the smaller the similarity is. Then "min" operation takes the minimum in the entropy values of all the features. Namely, we choose the feature with the smallest similarity to reflect the association degree of the Leader-Follow at time k. If the smallest similarity is larger than the threshold we set, then the similarities of all the features will be larger than the threshold and the Leader-Follow can be recognized to be associated at time k.

4.1 Algorithm Description

Step 1. Initialization

Assume that the set of Leaders consists of n system tracks, which are denoted as:

$$x_l = \{x_{l1}, x_{l2}, \cdots, x_{ln}\} \ . \tag{1}$$

The state estimate vector $\boldsymbol{x}_{li}$ at time k is denoted as:

$$x_{li}(k) = (r_{i1}(k), r_{i2}(k), \cdots, r_{ip}(k))^{\mathrm{T}} \quad i = 1,2,\cdots,n \tag{2}$$

where $r_{iq}\,(1 \le q \le p)$ is the feature of the track, p is the number of the features.

Assume that the set of Followers from sensor s consist of m local tracks, which are denoted as:

$$x_s = \{x_{s1}, x_{s2}, \cdots, x_{sm}\} \ . \tag{3}$$

The state estimate vector $\boldsymbol{x}_{sj}$ at time k is denoted as:

$$x_{sj}(k) = (r_{j1}(k), r_{j2}(k), \cdots, r_{jp}(k))^{\mathrm{T}} \quad j = 1,2,\cdots,m. \tag{4}$$

Step 2. Calculate the association degrees between Followers and Leaders at time k

Define the function of association degree.

$$\begin{aligned} \mathrm{a}_{ij}(k) = \min_{1\le t\le p}(&-\frac{r_{it}(k)}{r_{it}(k)+r_{jt}(k)}\log_2\frac{r_{it}(k)}{r_{it}(k)+r_{jt}(k)} \\ &-\frac{r_{jt}(k)}{r_{it}(k)+r_{jt}(k)}\log_2\frac{r_{jt}(k)}{r_{it}(k)+r_{jt}(k)}) \end{aligned} \tag{5}$$

By (5), the association degree of a Leader-Follower at time k is got.

Step 3. Association judgment

By (6), assign a Follower to the clustering which its nearest Leader represents:

$$i' = \arg\max_i (a_{ij}(k)) \ , \ i = 1,2,\cdots,n_l \ . \tag{6}$$

One Follower can be associated with one Leader at the most. And target location is a critical attribute in the association judgment. So if i' is not unique, the Leader with minimum distance from the Follower will be selected to cluster.

If the above clustering is done once, the N_a increases 1. Judge $\max_i(\mathrm{a}_{ij}(k)) > \mathrm{K}$. If it is true, Q_a increases 1. Otherwise, Q_a remains unchanged. If N_a equals N, Q_a is compared with Q. If $Q_a \ge$ Q, then the Leader-Follower is associated. Otherwise, the Leader-Follower is unassociated. The association judgments are done between the Followers from sensor s and each Leader at time k.

Here, K is a constant, which represents the threshold of association degree. Q_a is the association quality of a Leader-Follower, which is a non-negative integer and its initial value is 0. If the association degree is larger than K, then Q_a will increase 1, which indicates that the Leader-Follower is associated at the time step. Otherwise Q_a remains unchanged, which indicates that the Follower is unassociated with any Leader at the time step. N_a is the association step, which is a non-negative integer and its initial value is 0. When the association judgment is done once, its value increases 1. N is a constant, which represents the threshold of the association step. If $N_a >$ N, the association judgment is not done any more and Q_a is compared with Q. Q is a constant, which represents the threshold of the association quality.

Step 4. Ambiguity processing

One Follower from a sensor can be associated with one Leader at the most. So if a Follower is judged to be associated with multiple Leaders, the ambiguity processing is needed. Choose the Leader-Follower with the maximum association quality as the associated track pair. If the maximum is not unique, choose the Leader-Follower with the minimum average distance as the associated track pair. After ambiguity processing, each Follower is associated with one Leader at the most. The ambiguity processing of Leaders can also be done by the above procedure.

After track association, the associated Leader-Follower will be fused.

The global state estimation at k is:

$$\begin{aligned} \hat{x}(k) &= P_j(k)(P_i(k)+P_j(k))^{-1}x_i(k) \\ &\quad + P_i(k)(P_i(k)+P_j(k))^{-1}x_j(k) \end{aligned} \tag{7}$$

The global error covariance at k is:

$$P(k) = P_i(k)(P_i(k)+P_j(k))^{-1}P_j(k) \tag{8}$$

where $x_i(k)$ is the local state estimation and $P_i(k)$ is the local error covariance of the track from sensor i at time k.

In addition, if a Leader-Follower is judged to be associated, a counter should be started. After a certain time, the Leader-Follower is separated, and the association judgment is done again. The aim is to prevent error associations from long-standing in the system.

4.2 Theoretical Analysis

In the surveillance region, suppose M sensors observe T targets. If there are no missing measurements, $M\times T$ local tracks will be got. After a correct track association, $M\times T$ local tracks are divided into T categories, each category consist of M local tracks from different sensors.

If the local tracks from different sensors are supposed to be points and the association operations are viewed as edges, then the topology of the association in pairs among the local tracks from different sensors is shown in Fig. 1 and the topology of the association using Leader-Follower algorithm is shown in Fig. 2. The node in the two figures represents the local track or the system track, and the local tracks are from different sensors.

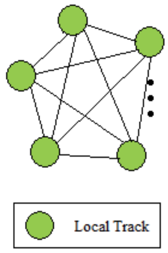

Fig. 1. Association in pairs among local tracks.

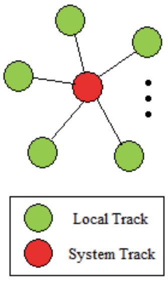

Fig. 2. Association using Leader-Follower algorithm.

If the local tracks from different sensors are associated in pairs, then the association times T_{ll} are:

$$T_{ll} = C_M^2 \times T^2 = \frac{M \times (M-1)}{2} \times T^2 . \tag{9}$$

If the Leader-Follower algorithm is adopted, then the association times T_{LF} are:

$$T_{LF} = (M-1) \times T^2 . \tag{10}$$

Comparing the above two results, we can see if the number of sensors M is more than 2, then $T_{LF} < T_{ll}$. It means the Leader-Follower algorithm requires less association times. And its superiority becomes more obvious as the number of sensors increases. In applications, a multi-sensor system includes multiple sensors, i.e. $M > 2$, so the Leader-Follower algorithm can significantly reduce the association times.

5. Experiments and Remarks

5.1 Initial Setup

In order to facilitate problem discussion, suppose all the state estimations sent to the fusion center are in the same coordinate system, all the sensors sample synchronously, and the delay time of data transmission is 0.

In the simulation, four sensors are designed to observe the targets at the same time. The targets run at a variable speed in three-dimensional space. In order to validate the algorithm performance, the algorithm is simulated 100 times with Monte Carlo method. The value range of K is approximately acquired by the normal approximation method. Then the value of K is fine-tuned by the simulation experiments, and is determined as 0.92 at last. And take $N = 12$ and $Q = 10$.

Three cases are set in the simulation. Case 1 is the sparse target environment in which 30 targets enter the surveillance space. Case 2 is the medium density target environment in which 60 targets enter the surveillance space. And Case 3 is the dense target environment in which 120 targets enter the surveillance space. Fig. 3, Fig. 4 and Fig. 5 show the tracks with the target number 30, 60 and 120 respectively.

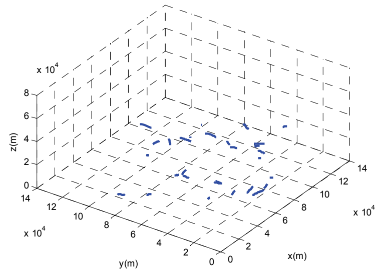

Fig. 3. 30 target tracks.

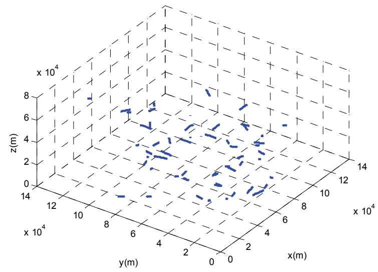

Fig. 4. 60 target tracks.

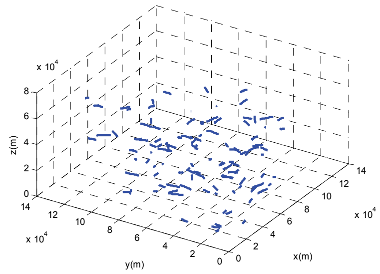

Fig. 5. 120 target tracks.

5.2 Experimental Results and Analysis

The algorithm performance is evaluated in terms of the following indexes: the probability of correct association (P_{ca}), the probability of missing association (P_{ma}), the probability of correct separation (P_{cs}) , the probability of error association(P_{ea}) and the association time.

In order to define P_{ca}, P_{ma}, P_{cs}, and P_{ea}, we take the following assumptions. In the process of a track association, sensor 1 has A tracks and sensor 2 has B tracks, then there are $A*B$ association track pairs. In them, N_c association track pairs from the same targets. After association, N_1 pairs are correctly judged to be associated, N_2 pairs are wrongly judged to be unassociated, and $N_c = N_1 + N_2$. N_s association track pairs from the different targets. After association, N_3 pairs are correctly judged to be unassociated, N_4 pairs are wrongly judged to be associated, and $N_s = N_3 + N_4$. N_f association track pairs consist of an invalid track at least. After association, N_5 pairs are correctly judged to be unassociated, N_6 pairs are wrongly judged to be associated, and $N_f = N_5 + N_6$. $A{*}B = N_c + N_s + N_f$. Then P_{ca}, P_{ma}, P_{cs}, and P_{ea} are defined as follow.

P_{ca} is the probability that the association track pairs from the same tracks is correctly judged to be associated, which is defined as:

$$P_{ca} = N_1 / N_c \tag{11}$$

P_{ma} is the probability that the association track pairs from the same tracks is wrongly judged to be unassociated, which is defined as:

$$P_{ma} = N_2 / N_c\,. \tag{12}$$

P_{cs} is the probability that the association track pairs from the different tracks is correctly judged to be unassociated, which is defined as:

$$P_{cs} = (N_2 + N_5)/(N_s + N_f)\,. \tag{13}$$

P_{ea} is the probability that the association track pairs from the different tracks is wrongly judged to be associated, which is defined as:

$$P_{ea} = (N_4 + N_6)/(N_s + N_f)\,. \tag{14}$$

Tab. 1, Tab. 2 and Tab. 3 show the comparisons on the average P_{ca}, P_{ma}, P_{cs}, and P_{ea} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 1, Case 2 and Case 3.

Algorithm	P_{ca}	P_{ma}	P_{cs}	P_{ea}
WTA	0.8831	0.1169	0.8239	0.1761
FCM	0.9232	0.0768	0.9425	0.0575
Leader-Follower	0.9201	0.0799	0.9366	0.0634

Tab. 1. The performance comparisons in Case 1.

Algorithm	P_{ca}	P_{ma}	P_{cs}	P_{ea}
WTA	0.7250	0.2750	0.7115	0.2885
FCM	0.8436	0.1564	0.8573	0.1427
Leader-Follower	0.8589	0.1411	0.9056	0.0944

Tab. 2. The performance comparisons in Case 2.

Algorithm	P_{ca}	P_{ma}	P_{cs}	P_{ea}
WTA	0.4869	0.5131	0.3234	0.6766
FCM	0.7894	0.2106	0.8367	0.1633
Leader-Follower	0.8036	0.1964	0.8850	0.1150

Tab. 3. The performance comparisons in Case 3.

Tab. 4 shows the average time for associating all targets by WTA, FCM and Leader-Follower in Case 1, Case 2 and Case 3.

We mainly focus on the algorithm performance in the dense target environment. The comparisons on the average P_{ca}, P_{ma}, P_{cs}, and P_{ea} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 3 are shown in Fig. 6, Fig. 7, Fig. 8 and Fig. 9 respectively.

Algorithm	The average association time(s)		
	Case1	Case2	Case3
WTA	1.3627	4.3183	14.5161
FCM	7.6775	25.9536	100.1188
Leader-Follower	4.2946	14.7320	55.7374

Tab. 4. The comparisons on association time.

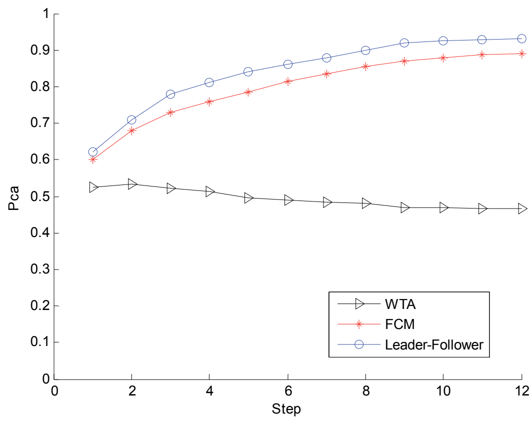

Fig. 6. The comparisons on P_{ca} in Case 3.

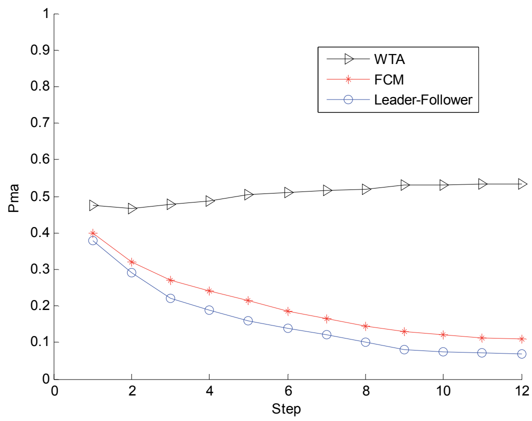

Fig. 7. The comparisons on P_{ma} in Case 3.

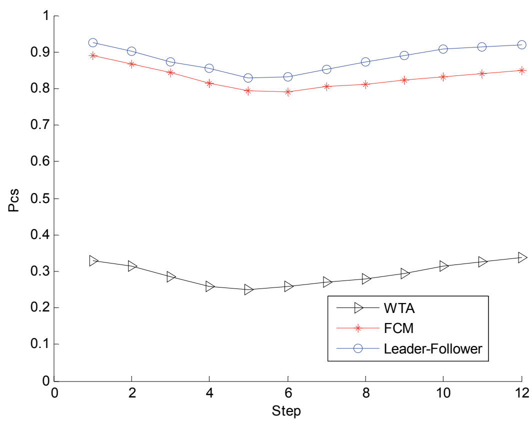

Fig. 8. The comparisons on P_{cs} in Case 3.

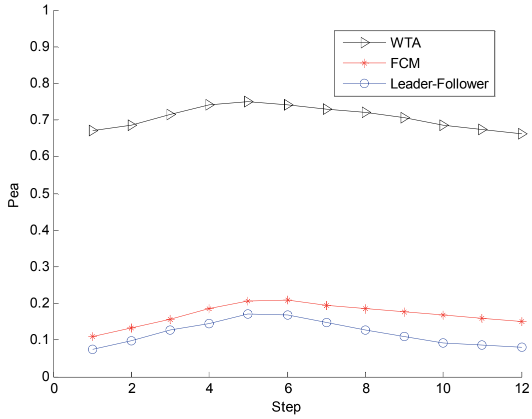

Fig. 9. The comparisons on P_{ea} in Case 3.

Fig. 6 is the comparisons on the average P_{ca} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 3. In Fig. 6, the curve representing WTA is on the bottom, the curve representing FCM is in the middle, and the curve representing Leader-Follower is at the top, which shows the average P_{ca} of WTA is lowest, that of FCM is medium, and that of Leader-Follower is highest.

Fig. 7 is the comparisons on the average P_{ma} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 3. In Fig. 7, the curve representing WTA is at the top, the curve representing FCM is in the middle, and the curve representing Leader-Follower is on the bottom, which shows the average P_{ma} of WTA is highest, that of FCM is medium, and that of Leader-Follower is lowest.

Fig. 8 is the comparisons on the average P_{cs} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 3. In Fig. 8, the curve representing WTA is on the bottom, the curve representing FCM is in the middle, and the curve representing Leader-Follower is at the top, which shows the average P_{cs} of WTA is lowest, that of FCM is medium, and that of Leader-Follower is highest.

Fig. 9 is the comparisons on the average P_{ea} of 100 times simulation for associating all targets by WTA, FCM and Leader-Follower in Case 3. In Fig. 9, the curve representing WTA is at the top, the curve representing FCM is in the middle, and the curve representing Leader-Follower is on the bottom, which shows the average P_{ea} of WTA is highest, that of FCM is medium, and that of Leader-Follower is lowest.

From the simulation results, we find that in the dense target environment, the speed of WTA is fastest, but its association accuracy gradually declines as the time step increases. FCM has a higher association accuracy than WTA, but it pays out a heavy computational cost. WTA and FCM both have the serious imbalance between accuracy and computational cost. Compared with WTA and FCM, Leader-Follower algorithm acquires the highest association accuracy with an acceptable computational cost.

6. Conclusion

In the paper, the track association problem is transformed into the on-line clustering problem with constraints, and the track association algorithm based on Leader-Follower online clustering is given. By doing association judgment between Leaders and Followers, the association times are reduced. Furthermore, by fusing the associated Leader-Follower to acquire a new Leader, fusion process and Leader generation is combined, which improved the efficiency of the fusion system. Simulation results show that in dense target environments, the algorithm balances the conflict between association accuracy and association speed.

Acknowledgements

The work described in this paper is supported by the National Natural Science Foundation of China under Grant No. 61201084 and No. 11301110, China Postdoctoral Science Foundation funded project under Grant No. 2013M541346, the State Scholarship Fund from the China Scholarship Council(CSC) under Grant No. 201203070290 and No. 201206125024, and the Fundamental Research Funds for the Central Universities under Grant No. HEUCF100604.

References

[1] AZIZ, A. M. Fuzzy track-to-track association and track fusion approach in distributed multisensory multi-target multiple-attribute environment. *Signal Processing*, 2007, vol. 87, no. 6, p. 1474 to 1492.

[2] OH, S., RUSSELL, S., SHANKAR, S. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 2009, vol. 54, no. 3, p. 481-497.

[3] HABTEMARIAM, B., THARMARASA, R., THAYAPARAN, T. A multiple-detection joint probabilistic data association filter. *IEEE Journal of Selected Topics in Signal Processing*, 2013, vol. 7, no. 3, p. 461-471.

[4] SINHA, A., DING, Z., KIRUBARAJAN, T., FAROOQ, M. Track quality based multitarget tracking approach for global nearest-neighbor association. *IEEE Transactions on Aerospace and Electronic Systems*, 2012, vol. 48, no. 2, p. 1179-1191.

[5] MAURER, D. E. Information handover for track-to-track correlation. *Information Fusion*, 2003, vol. 4, no. 4, p. 281-295.

[6] CHEN, H. M., BAR-SHALOM, Y. Track association and fusion with heterogeneous local trackers. In *Proceedings of the 46th IEEE Conference on Decision & Control*. New Orleans (USA), 2007, p. 2675-2680.

[7] PANAKKAL, V. P., VELMURUGAN, R. Effective data association scheme for tracking closely moving targets using factor graphs. In *2011 National Conference on Communications*. Bangalore (India), 2011, p.1-5.

[8] OUYANG, C., JI, H. Modified cost function for passive sensor data association. *Electronics Letters*, 2011, vol. 47, no. 6, p. 383 to 385.

[9] SIGALOV, D., SHIMKIN, N. Cross entropy algorithms for data association in multi-target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 2011, vol. 47, no. 2, p. 1166-1185.

[10] HAN, Y., ZHU, H., HAN, C. Track-to-track association in the presence of sensor bias and the relative bias estimation. In *15th International Conference on Information Fusion*. Singapore (Singapore), 2012, p. 1044-1050.

[11] KAPLAN, L. M., BAR-SHALOM, Y., BLAIR, W. D. Assignment costs for multiple sensor track-to-track association. *IEEE Transactions on Aerospace and Electronic Systems*, 2008, vol. 44, no. 2, p. 655-677.

[12] PAPAGEORGIOU, D. J., HOLENDER, M. Track-to-track association and ambiguity management in the presence of sensor bias. In *12th International Conference on Information Fusion*. Seattle (USA), 2009, p. 2012-2019.

[13] SATHYAN, T., SINHA, A., KIRUBARAJAN, T., McDONALD, M., LANG, T. MDA-based data association with prior track information for passive multi-target tracking. *IEEE Trans. Aerospace and Electronic Systems*, 2011, vol. 47, no. 1, p. 539 to 556.

[14] ROY, A. K., RAO, S. K. Multi-track association and fusion. In *2011 International Conference on Communications and Signal Processing*. Kerala (India), 2011, p. 131-135.

About Authors ...

Li XU was born in Shuangyashan, China. She received her Ph. D. in Computer Application Technology from Harbin Institute of Technology in 2012. Now she serves as an assistant professor at Harbin Engineering University. Her research interests include information fusion, intelligent planning and plan recognition.

Shuilin JIN was born in Xingtai, China. He received his Ph. D. in Mathematics from Jilin University in 2009. Now he serves as an assistant professor at Harbin Institute of Technology. His research interests include information fusion and operator algebra.

Guisheng YIN was born in Taixing, China. He received his Ph. D. in Control Theory and Control Engineering from Harbin Engineering University. Now he serves as a professor at Harbin Engineering University. His research interests include information fusion, database and virtual reality.