Trajectory-Joint Clustering Algorithm for Time-Varying Channel Modeling

Chen Huang, *Student Member, IEEE,* Andreas F. Molisch, *Fellow, IEEE,* Yangli-Ao Geng, Ruisi He, *Senior Member, IEEE,* Bo Ai, *Senior Member, IEEE,* Zhangdui Zhong, *Senior Member, IEEE*

***Abstract*—Clustering of multipath components (MPCs) is an important aspect of propagation channel modeling. When a time series of measurements, based on movement of transmitter and/or receiver, is available, the temporal evolution of MPCs can be used as a basis for clustering. We present an algorithm that bases clustering not only on distance of MPCs in the delay/angle space, but also how similar the temporal evolution of their parameters are. Sample results obtained from a vehicle-to-vehicle measurement campaign show good performance of the proposed algorithm.**

***Index Terms*—Clustering, tracking, multipath component, channel measurement and modeling, machine learning.**

I. Introduction

Accurate yet reasonably simple channel models are an essential requirement for the design and testing of wireless communications systems. The most widely used channel models aim to characterize the parameters of the multipath components (MPCs), such as amplitude, delay, and/or direction, in various scenarios. Extensive channel measurements have found that the MPCs generally occur in groups, also known as clusters. This motivates the widespread use of clustered channel models, which separately describe the inter-cluster and intra-cluster properties, thus greatly reducing complexity with minimal loss of accuracy, e.g., COST 2100 [1], 3GPP Spatial Channel Model [2], and WINNER [3]. Parameterization of the models from measurements requires estimation of the MPC parameters, and subsequent clustering.

Most measurements in the literature are done at several, isolated, spatial points, so that the MPCs are extracted from a single spatial "snapshot". The subsequent clustering then defines a cluster as a group of MPCs that have delay/angle channel parameters that are similar amongst each other, but significantly different from those of MPCs in other clusters. While these "static clustering" algorithms are popular [4]–[7], results can be sensitive to arbitrary threshold parameters. Examples for such algorithms are the [4] and the Kernel-power-density (KPD)-based clustering algorithm [5]. Instead of identifying clusters based on the MPCs, [6] identifies the cluster directly from the power angle spectrum of the current snapshot, which is less accurate but with less computation complexity as well.

In any case, most of the current clustering solutions only consider the distribution of MPCs in each snapshot separately and ignore the evolution pattern of clusters and MPCs over time, which naturally leads to the birth and death of clusters and MPCs. Nevertheless, the birth and death process of clusters and MPCs is a very typical and important characteristic of channels that are time-varying due to movement of transmitter (Tx) and/or receiver (Rx). For example, either a stationary object (e.g., tree) or moving object like a vehicle, that acts as a reflector/scatterer, appears and disappears, thus resulting in birth and death of a certain cluster. In this case, specifically when measurements on continuous *trajectories* of Tx/Rx are available, we argue that a more physically meaningful definition of clusters is the following: MPCs whose delays and angles show a similar evolution over time constitute a cluster, or more precisely, a trajectory-defined cluster. For this type of clustering, suitable algorithms generally fall into two categories:

- *Clustering first and tracking after*: To evaluate the time-varying characteristic, the most intuitive solution is to first identify the clusters in each snapshot, by using the solutions of *Static Clustering*, and then track the identified clusters over time. These solutions separate the clustering and tracking procedures, e.g., the power-angle-spectrum based clustering and tracking algorithm [6] and the multipath clustering and tracking algorithm [8];
- *Jointly clustering and tracking*: Since the moving pattern of each MPC is actually an important characteristic for clustering, clusters can be based on the moving trajectories of MPCs, e.g., Kalman-filter-based tracking and clustering algorithm [9] and the tracking-based clustering algorithm [10].

Measuring the difference of evolution between different MPCs' trajectories is crucial for identifying the trajectory-defined cluster, which, however, has not been well addressed

This study is supported by the National Key R&D Program of China under Grant 2016YFE0200900 and 2018YFF0212103, the National Natural Science Foundation of China under Grant 61922012, 61771037, 61725101 and U1834210, the Beijing Natural Science Foundation under Grant 4182047, the Royal Society Newton Advanced Fellowship under Grant NA191006, the Fundamental Research Funds for the Central Universities under Grant 2017RC031 and L172020, and the State Key Lab of Rail Traffic Control and Safety under Grant 2017JBM332 andRCS2018ZZ007. (*Corresponding authors: Ruisi He and Bo Ai.*)

C. Huang and Y.-A. Geng are with the School of Computer and Information Technology, School of Computer and Information Technology, Beijing, 100044, China (email: morning@bjtu.edu.cn).

A. F. Molisch is with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 (email: molisch@usc.edu).

R. He, B. Ai, and Z. Zhong are with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (email: ruisi.he@bjtu.edu.cn; boai@bjtu.edu.cn; zhdzhong@bjtu.edu.cn).

in the past. The Kalman-filter-based tracking and clustering algorithm in [9] only uses the trajectory for initialization of the next snapshot clustering, whereas the tracking-based clustering algorithm in [10] simply considers the moving probability of MPCs as a feature for clustering.

In this paper, we propose an alternative approach, in which we track the MPCs first, and cluster them subsequently based on the temporal evolutions. The main contributions of this paper are the following.

- To the authors' best knowledge, this paper is the first to develop a trajectory-based clustering algorithm for time-varying channels. The proposed algorithm naturally fits to the definition of clusters as MPCs with similar temporal evolution.
- We propose a novel distance measure function to measure the distance between trajectories in time dimension, which considers all the differences between the shape, the position, and the length of trajectories.
- We demonstrate the effectiveness of the approach on results from a vehicle-to-vehicle (V2V) channel measurement campaign, since such channels exhibit strong time variations and are thus an especially interesting case study.

II. Problem Description and Framework of Proposed Algorithm

In any wireless channel, the signal propagates from the Tx to the Rx via different paths, giving rise to the different MPCs. The parameters of MPCs in each snapshot can be extracted by using high-resolution-parameter-estimation (HRPE) algorithms, e.g., Rimax [11] or the space-alternating generalized expectation-maximization (SAGE) [12]. The most general channel representation is then the double-directional channel model [13], which represents the channel as the sum of the MPCs with complex amplitude α, delay τ, azimuth of departure (AOD) ϕ_T, elevation of departure θ_T, azimuth of arrival ϕ_R, elevation of arrival (EOA) θ_R, and Doppler Δf of the MPCs[1]. We consider M snapshots of data, $m = 1, 2, \cdots, M$, where each snapshot contains a number of N^m MPCs.[2] Thus, the n-th MPC in the m-th snapshot can be represented by the multi-dimensional parameter vector $\mathbf{x}_n^m = [\alpha_n^m, \tau_n^m, \phi_{T,n}^m, \theta_{T,n}^m, \phi_{R,n}^m, \theta_{R,n}^m, \Delta f_n^m], n = 1, 2, \cdots, N^m$.

The goal of the algorithm is to identify dynamic clusters in the time-varying channels, which requires both tracking and clustering of the MPCs extracted in each snapshot. As mentioned in Sec. I, utilizing the trajectories of the MPCs in the parameter space can improve the accuracy of clustering. Therefore, the proposed algorithm identifies the trajectory of each MPC first, then clusters MPCs based on the identified trajectory.

Thus, the MPCs are clustered considering not only the current parameters but also the parameters in the past as well as the future (for the case that clustering is done off-line for the evaluation of a measurement campaign). This allows to identify different clusters having a close distance in delay/angle space but with different evolution paths. Note that, extraction and tracking of MPCs can be either separate (HRPE + tracking) or joint (Extended Kalman-filter based parameter estimation and tracking). In the following, we propose a clustering algorithm for MPC based on the former approach.

[1]Note that the elevation domain may not be considered in some cases, e.g., the data are collected by a horizontal uniform linear array.

[2]Note that superscript m denotes an index, not an exponent.

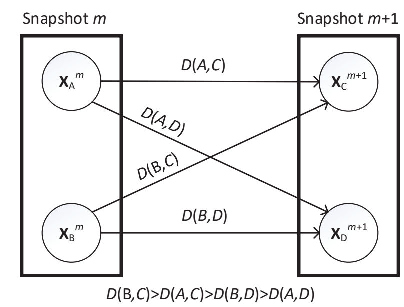


Fig. 1. Illustration of different trajectories between the MPCs. $\mathbf{x}_i^m$ and $\mathbf{x}_i^{m+1}$ are the MPCs in snapshot m and snapshot $m+1$, respectively, whereas D_i is the distance between different MPCs.

III. Tracking Joint Clustering Algorithm

This section presents the procedure of the proposed algorithm, which has two main stages: *Trajectory identification* and *Trajectory clustering*.

A. Trajectory identification

To identify the trajectory (temporal evolution of an MPC), the MPCs needs to be tracked over consecutive snapshots. Existing tracking algorithms for MPCs can be roughly classified into: *threshold-based*, i.e., MPCs are associated based on a fixed distance threshold [8], and *minimum distance-based*, i.e., MPC pairs are associated based on the minimum distance among each pair. For time-varying channels, it is hard to determine a fixed threshold for MPC tracking since the distribution of parameters continuously evolves with the dynamic environment. Therefore, we use a minimum distance-based solution in our research. Nevertheless, most of the past minimum distance-based solutions use the local minimum distance for tracking, which means a pair of MPCs having the smallest distance is primarily linked. This usually leads to a locally optimum result, as shown as $\{D(A, D), D(B, C)\}$ in Fig. 1, where the pair of MPCs $\mathbf{x}_1^m$ and $\mathbf{x}_2^{m+1}$ has the local minimum distance $D(A, D)$. However, the globally optimum result of Fig. 1 should be $\{D(A, C), D(B, D)\}$.

To achieve better accuracy, the trajectory of the MPC is identified by seeking the global minimum distance of all MPC pairs, as follows

$$\mathcal{D}_{min}^{m,m+1} = \arg \min_{\mathcal{D}^{m,m+1}} \sum_{\substack{i=1\\j=1}}^{N^m} D(\mathbf{x}_i^m, \mathbf{x}_j^{m+1}), \quad i \neq j \qquad (1)$$

where $D(\mathbf{x}_i^m, \mathbf{x}_j^{m+1})$ is the distance between the i-th MPC and the j-th MPC in two consecutive snapshots, $\mathcal{D}^{m,m+1}$ is the set of all possible trajectories between these snapshots. The distance $D(\mathbf{x}_i^m, \mathbf{x}_j^{m+1})$ thus is computed as a normalized Euclidean distance of angle, delay, and power [14].

To solve the problem in (1), we use the Kuhn-Munkres algorithm, which is able to find the minimized weight-perfect matching in a bipartite graph of a general assignment problem. The details of the Kuhn-Munkres-based tracking algorithm can be found in [6]. It is noteworthy that the number of MPCs in two snapshots is allowed to be different, i.e., $N^m \neq N^{m+1}$, so that the MPCs that do not associate to others are considered as dying MPCs or born MPCs.

By using the above approach, we can track the MPCs in consecutive snapshots. However, as indicated above, there are also births and deaths in the MPC evolutions. If an MPC $\mathbf{x}_i^m$ disappears in the current snapshot and another MPC $\mathbf{x}_j^{m+1}$ appears close to $\mathbf{x}_i^m$, they might be classified as the same MPC, which may lead to a jump in the trajectory (henceforth called "hopping point"). In this case, we exploit an angular hopping threshold $\mathbf{\Gamma}_n^m$ as a function of Doppler to determine whether the hopping point should be interpreted as separating two distinct trajectories, as follows:

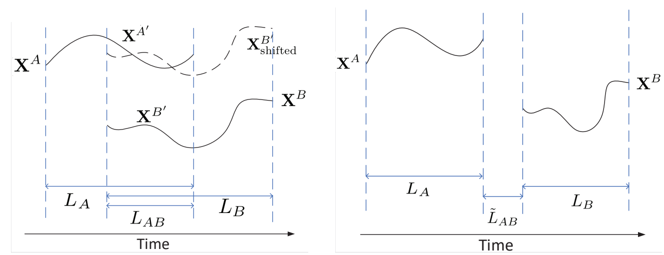
$$\mathbf{\Gamma}_n^m = \mathbf{\Upsilon}|\Delta f_n^m T_s| + \mathbf{\Gamma}_{\text{baseline}} \tag{2}$$

where $\mathbf{\Upsilon}$ is a vector of the scale coefficient for MPCs' parameters, i.e., delay, AOA, and AOD, a larger scale coefficient indicates a larger $\mathbf{\Gamma}_n^m$ for the same Doppler Δf_n^m, and T_s is the time interval between two samples. If two consecutive MPCs ($\mathbf{x}_i^m$, $\mathbf{x}_j^{m+1}$) in one trajectory have a variance of each parameter more than a threshold $\mathbf{\Gamma}_n^m$, then $\mathbf{x}_j^{m+1}$ is considered as a start of a new trajectory. The principle here is that for an MPC with a larger Doppler, which usually indicates a faster-moving scatterer, we allow a bigger variance during evolution, where $\mathbf{\Gamma}_{\text{baseline}}$ is the minimum variance of different parameters, e.g., due to noise in the observations.

B. Trajectory clustering

The past clustering algorithms mostly identified the cluster based on the distribution of MPCs' parameters in the current snapshot, where the evolution of the MPC has not been taken into account. Instead, our approach directly clusters the trajectories of the MPCs. In other words, the MPCs that have a similar evolution pattern and are close to each other, are identified as the same cluster. Considering its good overall clustering performance, the KPD algorithm [5] is used here to cluster the trajectories, where the clustering of the objects is based on the density of parameters in the multi-dimensional parameter space. Due to the space constraints, we refer for details of KPD to [5]. The main change is that we cluster trajectories rather than MPCs in a snapshot, therefore, how to measure the distance among trajectories is a key factor determining performance.

For any two time-varying trajectories, there are two possible cases: i) two trajectories overlap each other in a certain time period, which means both of them exist during a period of time, i.e., Fig. 2(a); and ii) two trajectories are separated in time, i.e., Fig. 2(b). Let $\mathbf{X}^A$ and $\mathbf{X}^B$ represent the two trajectories to compare, $\mathbf{X}^{A/B} = \left[\mathbf{x}_1^{A/B}, \mathbf{x}_2^{A/B}, \cdots, \mathbf{x}_{L_{A/B}}^{A/B}\right]$, where L_A and L_B are the lengths of the two trajectories, $\mathbf{x}_i^A$ and $\mathbf{x}_i^B$ are the i-th MPC in the two trajectories. For the overlapping case, $\mathbf{X}^{A'}$ and $\mathbf{X}^{B'}$ are the overlapping segment of $\mathbf{X}^A$ and $\mathbf{X}^B$, respectively, with a length of L_{AB}. For the separated case, $\tilde{L}_{AB}$ is the length of the time interval between two trajectories. We use different strategies of distance calculation for these two case. For the overlapping case, the trajectory distance is calculated based on the shape distance and the actual distance in the overlapping segment. Specifically, the shape distance is defined as



(a) (b)

Fig. 2. Illustration of position of two trajectories, which (a) overlaps each other (both exist during a period of time) and (b) are separated in time.

$$D(\mathbf{X}^A, \mathbf{X}^B)_{\text{sp}} = \min_{\mu} \sum_{i=1}^{L_{AB}} ||\mathbf{x}_i^{A'} - (\mathbf{x}_i^{B'} - \mu)||_2 \tag{3}$$

where μ is an offset used for shifting the $\mathbf{X}^{B'}$ to the position of $\mathbf{X}^{A'}$ for comparing the difference of the shape. The norm function is a convex function, thus we can obtain the closed form solution of μ and re-write (3) as:

$$D(\mathbf{X}^A, \mathbf{X}^B)_{\text{sp}} = \sum_{i=1}^{L_{AB}} ||\mathbf{x}_i^{A'} - (\mathbf{x}_i^{B'} - \sum_{i=1}^{L_{AB}} \frac{\mathbf{x}_i^{A'} - \mathbf{x}_i^{B'}}{L_{AB}})||_2. \tag{4}$$

By applying μ, $\mathbf{X}^{B'}$ plotted in Fig. 2(a) is shifted to $\mathbf{X}^{B'}_{\text{shifted}}$ for comparing the shape difference with $\mathbf{X}^{A'}$. The actual distance calculation can be expressed as

$$D(\mathbf{X}^A, \mathbf{X}^B)_{\text{at}} = \sum_{i=1}^{L_{AB}} ||\mathbf{x}_i^{A'} - \mathbf{x}_i^{B'}||_2. \tag{5}$$

Based on the obtained shape distance and actual distance, the overall trajectory distance between trajectories can be expressed as:

$$D(\mathbf{X}^A, \mathbf{X}^B) = \zeta \left(\chi_1 D(\mathbf{X}^A, \mathbf{X}^B)_{\text{sp}} + \chi_2 D(\mathbf{X}^A, \mathbf{X}^B)_{\text{at}}\right) \tag{6}$$

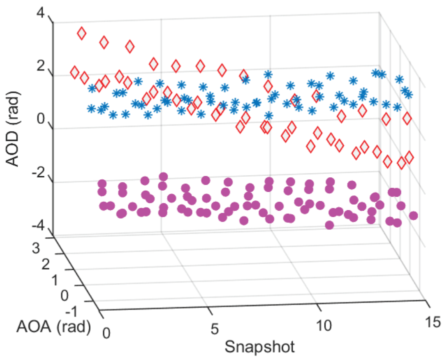
where $\chi_{1,2}$ are factors determining the weight of the shape distance and the actual distance and we set them to $\chi_1 = \chi_2 = 0.5$ in our examples. ζ is a length based factor:

$$\zeta = \arctan(\frac{\min(L_A, L_B) + \tilde{L}_{AB}}{L_{AB} + 1}). \tag{7}$$

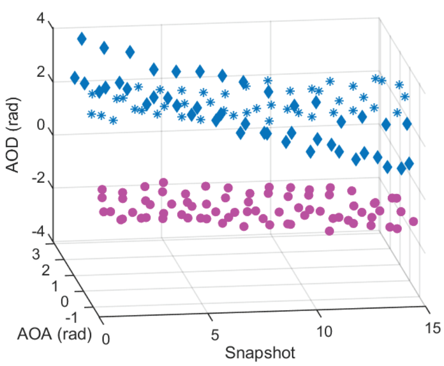
For the separated case, the trajectory distance is calculated as the difference between the mean value of the MPCs in the two trajectories.

$$D(\mathbf{X}^A, \mathbf{X}^B) = \zeta || \frac{\sum_{m=1}^{L_A} e^{m - L_A} \mathbf{x}^A}{\sum_{m=1}^{L_A} e^{m - L_A}} - \frac{\sum_{m=1}^{L_B} e^{1-m} \mathbf{x}^B}{\sum_{m=1}^{L_B} e^{1-m}} ||_2 \tag{8}$$

where $e^{(\cdot)}$ is a decaying factor that makes the MPCs closer



(a)



(b)

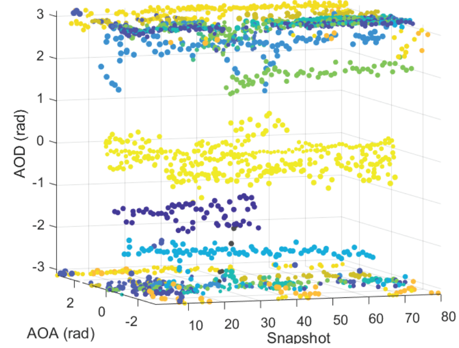
Fig. 3. Clustering evaluation based on synthetic data, where the ground truth of clusters is given by different marks, and the clustering results of (a) the proposed algorithm and (b) the Kalman-filter-based algorithm are given by different colors.

to the end of $\mathbf{X}_A$ and the beginning of $\mathbf{X}_B$ have a higher weight. By applying a factor ζ, the distances between each two trajectories are weighted considering their relative positions. Specifically, for the overlapping case, ζ decreases with a longer overlapping segment L_{AB} of the compared trajectories, i.e., the trajectories having longer overlapping segment will have relatively smaller distance and vice versa. For the separated case, ζ decreases with a shorter trajectory interval $\tilde{L}_{AB}$, i.e., the trajectories with shorter interval will have relatively smaller distance.

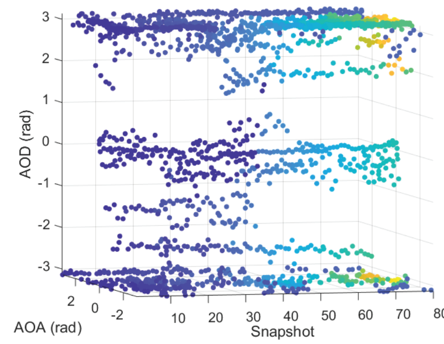
IV. Performance Validation

A. Clustering Accuracy

To evaluate the clustering performance, a synthetic channel is built to generate MPCs with a known ground truth of clustering. Both our algorithm and the Kalman-filter-based clustering algorithm in [9] are applied for comparison. As shown in Figs. 3(a) and (b), the ground truth of cluster identification is presented by using different marks, i.e., square, diamond, and circle. The clustering results of (a) the proposed algorithm and (b) the Kalman-filter-based clustering algorithm are given by different colors. It can be seen that the Kalman-filter-based method cannot distinguish two clusters if they are close to each other, e.g., the cluster marked by red edge, whereas the proposed algorithm can well recognize them due to their different moving patterns, e.g., two clusters plotted in blue and red.



(a)



(b)

Fig. 4. Clustering results obtained by using (a) the proposed algorithm, (b) the Kalman-filter based clustering algorithm, whereas (c) gives the raw extracted MPC in dB. The MPCs of the same cluster are plotted by the same color dot and the size of the dot indicates the delay of the MPC.

TABLE I
PARAMETERS SETTINGS

$\mathbf{\Gamma}_{\text{baseline}}$			$\mathbf{\Upsilon}$		
Γ_{delay}	Γ_{AOA}	Γ_{AOD}	Υ_{delay}	Υ_{AOA}	Υ_{AOD}
$0.5ns$	$15°$	$15°$	2	60	60

For a practical validation, the proposed algorithm is tested on V2V measurement data, which were collected by a self-built real-time MIMO channel sounder. The details of the V2V measurement campaign can be found in [16]. Table I gives the parameter settings for the tracking threshold, which is used in (2).

Figs. 4(a)-(c) give clustering results obtained by using (a) the proposed algorithm, (b) the Kalman-filter-based clustering algorithm, whereas (c) gives the raw extracted MPCs. The MPCs belonging to the same cluster are plotted by the same color dot and the size of the dot indicates the delay of the MPC. The Kalman-filter-based clustering algorithm only uses the tracking result as an initial position for KpowerMeans clustering, and the cluster members keep changing during the iteration, which results in new cluster IDs and causes the gradient color in Fig. 4(b).

To better evaluate the clustering performance, we use the *Xie-Beni index* [17] to evaluate the two algorithms. The Xie-Beni index is an objective validation method, which has been shown to be a good measure of performance for evaluating clustering algorithm for MPCs. This index is calculated based

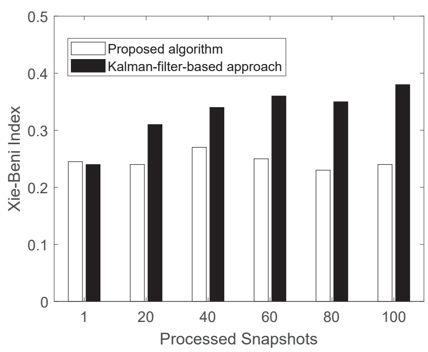


Fig. 5. Performance evaluation of the proposed algorithm and the Kalman-filter-based clustering algorithm by using the Xie-Beni index.

on the intra-cluster and inter-clusters distances, as follows:

$$V_{\mathrm{XB}} = \frac{\sum_{j}^{N_c} \sum_{i}^{N_x^{c_j}} (D(\mathbf{x}_i, \mathbf{C}_j))^2}{N \times [\min_{i \neq j} \{D(\mathbf{C}_i, \mathbf{C}_j) | i, j \in 1, \cdots N_c\}]^2} \quad (9)$$

where $\mathbf{C}_j$ is the j-th cluster, N_c is the number of clusters, N is the total number of clustered MPCs, and $N_x^{c_j}$ is the number of MPCs in the j-th cluster. The distance function $D(\cdot)$ is a normalized Euclidean distance following (1). The general idea of the Xie-Beni index is to measure the ratio of intra-cluster distance and inter-cluster distance. Hence, a small value of the Xie-Beni index indicates a better clustering performance. It is noteworthy that to validate the time-varying clustering algorithm, the time dimension of the data is treated as another feature dimension. Fig. 5 gives the performance evaluation of the proposed algorithm and the Kalman-filter-based clustering with different numbers of processed snapshots. It is found that, for clustering the MPCs in a single snapshot, the compared two algorithms show similar performance. With the increase of the number of processed snapshots, the performance of the Kalman-filter-based approach keeps decreasing, whereas the proposed algorithm can maintain a good performance due to the utilization of the MPCs' evolution pattern during the clustering.

B. Computation Complexity

Besides, the difference in clustering performance, the computational complexity of a clustering algorithm is also an important performance indicator. Since the joint tracking and clustering algorithms consist of two procedures, i.e., tracking and clustering, the computational complexity is determined based on these two parts. Specifically, the computational complexity of the proposed algorithm is determined by the K-M algorithm and the KPD algorithm, i.e., $\mathcal{O}(M \cdot \bar{N}^3 + N_L^2 \cdot d)$, where $\bar{N}$ is the average number of MPCs in each snapshot, N_L is the total number of the trajectories in M snapshots and d is the number of feature dimensions. Similarly, the computational complexity for the Kalman-filter-based algorithm is $\mathcal{O}(M \cdot (\bar{N}^2 + d^3 + d \cdot \bar{N} \cdot K \cdot T))$, where T is the number of iterations in the clustering procedure and K is the number of clusters. In this sense, the proposed algorithm has higher but polynomial computational complexity, which conversely provides a better clustering accuracy.

V. Conclusion

In this paper, we proposed a trajectory-based clustering algorithm. The algorithm identifies the cluster based on the difference between MPCs' evolution pattern to improve the clustering accuracy and robustness. Data from a V2V measurement campaign are used for performance evaluation. In the investigated examples, the proposed algorithm achieves a better performance than the conventional Kalman-filter-based clustering approach.

References

[1] L. Liu et al., "The COST 2100 MIMO channel model," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 92–99, Dec. 2012.

[2] 3GPP TR 25.996, "Spatial channel model for multiple input multiple output (MIMO) simulations," Jul. 2018.

[3] J. Meinila, P. Kyosti, T. Jamsa, and L. Hentila, "WINNER II channel models," in *Radio Technologies Concepts IMT-Advanced. Hoboken*, NJ, USA: Wiley, pp. 39–92, 2009.

[4] N. Czink, P. Cera, J. Salo, E. Bonek, J. Nuutinen and J. Ylitalo, "A framework for automatic clustering of parametric MIMO channel data including path powers," in *Proc. IEEE Vehicular Technology Conference*, Montreal, Que., 2006, pp. 1–5.

[5] R. He et al., "A kernel-power-density-based algorithm for channel multipath components clustering," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7138-7151, Nov. 2017.

[6] C. Huang et al., "A power-angle-spectrum based clustering and tracking algorithm for time-varying radio channels," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 291-305, Jan. 2019.

[7] C. Huang et al., "Machine learning-based data processing techniques for vehicle to vehicle channel modeling," *IEEE Communications Magazine*, accepted, 2019.

[8] P. Hanpinitsak, K. Saito, J. Takada, M. Kim and L. Materum, "Multipath clustering and cluster tracking for geometry-based stochastic channel modeling," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 11, pp. 6015-6028, Nov. 2017.

[9] N. Czink et al., "Tracking time-variant cluster parameters in MIMO channel measurements: algorithm and results" *COST 2100* TD(07)336, 2007.

[10] C. Huang, R. He, Z. Zhong, Y. Geng, Q. Li and Z. Zhong, "A novel tracking-based multipath component clustering algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 2679–2683, 2017.

[11] A. Richter, "Estimation of radio channel parameters: Models and algorithms," PhD dissertation, Technischen Universitat Ilmenau, Ilmenau, Germany, Dec. 2005.

[12] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus and K. Ingeman Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 3, pp. 434-450, March 1999.

[13] M. Steinbauer, A. F. Molisch and E. Bonek, "The double-directional radio channel," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 4, pp. 51-63, Aug. 2001.

[14] C. Lai, R. Sun, C. Gentile, P. B. Papazian, J. Wang and J. Senic, "Methodology for multipath-component tracking in millimeter-wave channel modeling," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 3, pp. 1826–1836, March 2019.

[15] N. Czink, P. Cera, J. Salo, E. Bonek, J.-P. Nuutinen, and J. Ylitalo, "Automatic clustering of MIMO channel parameters using the multipath component distance measure," in *Proc. WPMC05*, Aalborg, Denmark, Sept. 2005.

[16] R. Wang, C. U. Bas, O. Renaudin, S. Sangodoyin, U. T. Virk and A. F. Molisch, "A real-time MIMO channel sounder for vehicle-to-vehicle propagation channel at 5.9 GHz," In *Proc. 2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1–6.

[17] S. Mota, F. Perez-Fontan and A. Rocha, "Estimation of the Number of Clusters in Multipath Radio Channel Data Sets," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 5, pp. 2879–2883, May 2013.