Cluster Analysis of Pedestrian Mobile Channels in Measurements and Simulations

Chao Wang 1, Jianhua Zhang 2,* and Guangzhong Yu 1

1 Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China; chaowang@bupt.edu.cn (C.W.); ygz@bupt.edu.cn (G.Y.)
2 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: jhzhang@bupt.edu.cn

Received: 9 January 2019; Accepted: 26 February 2019; Published: 1 March 2019

Abstract: In wireless communication systems, channels evolve when user terminals move. To further understand channel variation, and especially the evolution of clusters in mobile channels, a set of experiments was designed. First, we performed pedestrian mobile measurements in an urban macro (UMa) scenario at 3.5 GHz, and the K-power means-Kalman filter (KPMKF) algorithm was used for clustering and tracking. By this process, the trajectory of different clusters could clearly be described during measurement. The birth and death rate of clusters per snapshot show that the change of one or two clusters in each snapshot takes more probabilities. In addition, the differences of the cluster lifetime between the clustering process with and without the Kalman filter (KF) algorithm are given to show the effect from the KF. Second, channel simulations were implemented based on the above observed results. The spatial-consistency feature was introduced to get closer to the measured channels, which is based on the primary module of International Mobile Telecommunications-2020 (IMT-2020) channel model. Comparisons among measurements and simulations with and without this feature show that adding this feature improves simulation accuracy. To explore a novel method to characterize clusters during linear movement, a gradient boosted decision-tree (GBDT) algorithm is introduced. It uses the above characteristics of clusters and channel impulse responses (CIRs) as the training and validating dataset. The root mean square error (RMSE) shows that this is promising.

Keywords: channel measurement; IMT-2020 channel model; clustering; Kalman filter; gradient boosted decision tree; massive MIMO

1. Introduction

Research on the channel model is one of the basic works in wireless communication systems. It is an abstract description of wireless propagation characteristics. Researchers in the field of wireless communications have had great interest in channel modelling in various kinds of propagation environments (such as urban, suburb, and indoor) since the 1970s. They focused on how to build a mathematical model based on theoretical and measurement analysis, which approximately simulates real channel conditions. From the 3G to the 4G era, the channel model has already changed from the tapped delay line (TDL) model in M. 1225 [1], which is known as the International Mobile Telecommunications-2000 (IMT-2000) channel model, to geometry-based stochastic channel model (GBSM) in M. 2135 [2], which is known as IMT-Advanced channel model. ITU radiocommunication sector (ITU-R) started the project to draft a new channel model, called IMT-2020, to meet the new requirements in the 5G era, which was published in M. 2412 on October 2017 [3].

As the basis of GBSM models, a cluster can reflect path characteristics in the channels, and their variation over time can show channel time-variant and spatial-variant features in mobile scenarios.
Many measurements in different scenarios were conducted, followed by theoretical analysis to study the statistical modelling of clusters. Reference [4] proposes the average number of clusters in indoor office environments, which can reflect the active number of clusters in different positions. Reference [5] shows the lifetimes of clusters in indoor-to-outdoor and outdoor-to-indoor environments, while Reference [6] reports them in an indoor hall environment. Moreover, how to continuously parameterize the clusters, and track and identify them in mobile channels have been considered in many studies. Reference [7] built an evolution model of the number of clusters from dynamic double-directional measurements in an indoor environment, and uses cluster-level parameters to describe the mobile channels. Reference [8] gives a Kuhn-Munkres-based method to model the evolution of clusters that is based on simulations of the 3rd Generation Partnership Project (3GPP) channel model.

However, cluster-tracking statistical analysis to massive Multiple Input Multiple Output (MIMO) pedestrian mobile measurements, which can detect richer scatterers in environments, needs to be further studied. Although we performed cluster analysis in References [9,10] with the K-power means (KPM) algorithm, it can only independently do clustering in each snapshot. The relationships of clusters between neighboring snapshots and their birth–death process are unknown, nor does the corresponding validation work on the primary module of the IMT-2020 channel model.

Considering this, we focused on better presenting the evolution of clusters in mobile channels in both measurements and simulations. Thus, the contributions of this paper are summarized as follows.

- The KPM-Kalman filter (KPMKF) algorithm is used for massive MIMO urban macro (UMa) pedestrian mobile scenario measurement. With this method, evolution including the birth and death rate and lifetime of clusters can be clearly shown. The difference of clustering results with and without Kalman filter (KF) part is also shown.
- The spatial-consistency feature is introduced and implemented to complement the primary module of the IMT-2020 channel model. It makes the channel evolve smoothly without discontinuity when Transmitter (Tx)/Receiver (Rx) moves. With this feature, cluster characteristics are highly correlated among neighboring moving locations, which is closer to real conditions. In the meantime, comparisons among between measurements and simulations with and without this feature are shown.
- Inspired by Reference [11], a novel clustering method based on the gradient boosted decision-tree (GBDT) algorithm is explored to train and validate the results from the above measurements and simulations. It was proven to be an effective way to characterize cluster evolution in certain scenarios during linear movement.

The remainder of this paper is organized as follows. Section 2 gives details of the measurement setup, the data process, and the analysis results. In Section 3, generation procedures of channel simulations by the IMT-2020 channel model with the spatial-consistency feature are described, as are the simulation results. Section 4 gives the framework of the GBDT-based model, and the validated results by the GBDT-based model. Finally, conclusions are drawn in Section 5.

2. Cluster Tracking in Mobile Measurements

In contrast to channels from a conventional MIMO system, massive MIMO systems give richer channel information [12–14]. Considering this, a virtual massive MIMO pedestrian mobile measurement campaign was performed to observe cluster evolution. To characterize the evolution process of clusters, which is greatly influential on cluster number in mobile channels, the tracking algorithm was considered. All details of the measurement setup, the data extraction and processing method, and data analysis are shown in the following.

2.1. Measurement Setup and Data Processing

Channel measurement took place in a square at the Beijing University of Posts and Telecommunications (see Figure 1), which can be seen as a typical UMa scenario. The used channel
sounder had a carrier frequency of 3.5 GHz and 200 MHz bandwidth and collected the raw data of channel impulse responses (CIRs). A 32-element uniform planar array (UPA) was used at the Tx side, shown in Figure 2a. At the Rx side, the dual-polarized omnidirectional array (ODA) was equipped with a 56-element antenna. In this case, Antennas 1–16 were chosen for the measurement campaign (see Figure 2b). Measurement details are listed in Table 1.

During measurement, the Rx moved along with the 8 m rail at a constant speed of 1.4 m/s (pedestrian speed). The Tx side was set on the roof of a teaching building. Two measurement routes were in line-of-sight (LoS) and non-line-of-sight (NLoS) conditions, respectively. Route R1, moving from east to west, had LoS conditions. R2, moving from north to south, had NLoS conditions, and was mainly blocked by the building and shrubs.

To form the massive MIMO array with 256 antennas, a virtual-measurement method was adopted. Figure 3 shows the scheme of 256-element combining antenna array, giving 8 continuous positions. After collecting the raw data, we combined the 8 CIRs into one group.

\[ h_i(t, \tau) = \sum_{n=1}^{N} \alpha_n^{(i)} e^{-j\phi_n^{(i)}(t)} \delta(\tau - \tau_n^{(i)}), \]  

where \( i = 1, 2, \cdots, 8 \). \( \alpha_n^{(i)} \), \( \phi_n^{(i)}(t) \) and \( \tau_n^{(i)} \) represent the amplitude, phase, and delay of the \( n \)th sample in the time domain at the \( i \)th angle bin, respectively. \( N \) is the number of collected CIRs.

Then, CIRs are combined into one group as

\[ h_{\text{virtual}}(t, \tau) = [h_1(t, \tau), h_2(t, \tau), \cdots, h_8(t, \tau)], \]  

where \( h_{\text{virtual}}(t, \tau) \) can be considered as the equivalent CIR collected from a 256 antenna UPA. The rationality of this method was proven in Reference [15]. It shows that there is a good match between the power delay profile (PDP) calculated from CIRs collected from the measurement campaigns and the combined CIRs. Furthermore, spatial angular characteristics, including elevation angle of departure (EoD), azimuth angle of departure (AoD), elevation angle of arrival (EoA), and azimuth angle of arrival (AoA), which were estimated from the combination, fit well with those from the measurement campaigns.

Figure 1. Measurement map (Red triangle is the Tx location. R1 and R2 are the measurement routes in LoS and NLoS conditions, respectively).
Figure 2. Antenna used in the measurement. (a) Tx: 4 × 4 patches, with each patch comprising a pair of cross-polarized antennas. (b) Rx: 8 adjacent sides with 3 patches each, a top surface with 4 patches; each patch contains a pair of cross-polarized antennas.

Table 1. Specifications of measurement campaign.

<table>
<thead>
<tr>
<th>Antenna Type</th>
<th>ODA(Rx)</th>
<th>UPA(Tx)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of antenna ports</td>
<td>56 (16 ports were used)</td>
<td>32</td>
</tr>
<tr>
<td>Radiation pattern</td>
<td>Omnidirectional</td>
<td>Hemispherical</td>
</tr>
<tr>
<td>Inter element spacing</td>
<td>41 mm</td>
<td>41 mm</td>
</tr>
<tr>
<td>Number of elements</td>
<td>28</td>
<td>16</td>
</tr>
<tr>
<td>Polarized</td>
<td>±45°</td>
<td>±45°</td>
</tr>
<tr>
<td>Angle range</td>
<td>Azimuth</td>
<td>−180°~180°</td>
</tr>
<tr>
<td>Elevation</td>
<td>−70°~90°</td>
<td>−70°~70°</td>
</tr>
<tr>
<td>Center frequency</td>
<td>3.5 GHz</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>200 MHz</td>
<td></td>
</tr>
<tr>
<td>PN sequence</td>
<td>255 chips</td>
<td></td>
</tr>
<tr>
<td>Speed of Rx</td>
<td>1.4 m/s (pedestrian)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Scheme of antenna combining array in virtual measurements.

After combining the 8 groups of CIRs sequences into one group of sequences, the space-alternating generalized expectation maximization (SAGE) algorithm was used to estimate the channel parameters from the raw data [16], which provides a joint estimation of parameter set \( \theta_l = \{ \tau_l, \phi_d, \Phi_l, \Omega_l, \alpha_l \} \), \( l = \{1, \ldots, L\} \). \( \tau_l, \phi_d, \Phi_l, \Omega_l, \) and \( \alpha_l \) denote propagation delay, the Doppler shift, the AoD, the AoA, and polarization of the \( l \)th propagation subpath, respectively. Specifically, \( \Phi_l = [\theta_T, \phi_T] \), \( \Omega_l = [\theta_R, \phi_R] \), where \( \theta_T, \phi_T, \theta_R, \) and \( \phi_R \) denote the EoD, AoD, EoA, and AoA, respectively. The number of estimated multipath components (MPCs) was set to 100 in each snapshot.
With this information on the MPC level, we could cluster them with similar parameters, e.g., AoA, AoD, and delay. In References [9,10], we only used KPM for clustering, and the process worked independently in each snapshot. Although the evolution of clusters could be observed along with the movement, the relationships among continuous snapshots and statistical analysis cannot be seen by KPM. For parameterizing clusters in mobile channels over time and showing their continuous evolution, KPM was used for clustering and KF was used for tracking [17,18].

The flowchart of the KPMKF algorithm is shown in Figure 4, where \( \overrightarrow{X}^{(n)} \) is the data matrix of 100 MPCs in an \( n \)th snapshot. \( \overrightarrow{X}^{(n)} = [x_1^{(n)}, x_2^{(n)}, \ldots, x_i^{(n)}, \ldots, x_{100}^{(n)}]^T \), \( x_i^{(n)} = [\tau_i^{(n)}, \theta_{Tx,i}^{(n)}, \theta_{Rx,i}^{(n)}, \phi_{c}^{(n)}, \phi_{Rx,i}^{(n)}] \), \( \theta_{Rx,i}^{(n)} \) represents the matrix of MPC power. State model \( g_c^{(n)} = [\tau_c^{(n)}, \theta_{Tx,c}^{(n)}, \theta_{Rx,c}^{(n)}, \Delta \theta_{Tx,c}^{(n)}, \Delta \theta_{Rx,c}^{(n)}, \Delta \phi_{Rx,c}^{(n)}, \Delta \phi_{Rx,c}^{(n)}]^T \). The matrix of cluster centroids is \( \mu_c^{(n)} = [\tau_c^{(n)}, \theta_{Tx,c}^{(n)}, \theta_{Rx,c}^{(n)}, \phi_{c}^{(n)}, \phi_{Rx,c}^{(n)}] \).

In this framework, \( \overrightarrow{X}^{(1)} \) and \( \overrightarrow{P}^{(1)} \) were used as initial input to the KPM part. After clustering in the first snapshot, we obtained \( \mu_c^{(1)} \) and initial state \( g_{c_1}^{(10)} \). Then, KF uses these two parameters to update the state as \( g_{c_1}^{(11)} \), and predict the state as \( g_{c_1}^{(21)} \) for the second snapshot. Then, we input the \( \overrightarrow{X}^{(2)} \) and \( \overrightarrow{P}^{(2)} \) to perform clustering in the second snapshot. Then, KF takes both the predicted and KPM-processed values into consideration to get \( g_{c_2}^{(32)} \). The process cycle keeps working until the last snapshot is considered. The details of KPM and KF are given as follows, respectively.

2.1.1. K-Power Means Part

Generally, KPM minimizes the total weight of multipath component distance (MCD) to the corresponding cluster centroid by iterations [19]. The MCD between the \( i \)th and \( j \)th MPC is calculated by

\[
MCD_{ij} = \sqrt{\left\| MCD_{\tau,ij} \right\|^2 + \left\| MCD_{\theta,ij} \right\|^2 + \left\| MCD_{\phi,ij} \right\|^2}
\] (3)

of which \( MCD_{\tau,ij} \) is the delay distance given as follows:

\[
MCD_{\tau,ij} = \eta \cdot \frac{|\tau_i - \tau_j|}{\Delta \tau} \cdot \frac{\tau_d}{\Delta \tau}
\] (4)

\( \eta \) is a scaling factor to adjust the weight of delay in the distance function. \( \Delta \tau \) means the range of delay, and \( \Delta \tau = \max_{i,j} |\tau_i - \tau_j| \). \( \tau_d \) is the standard deviation of delay.
Angle distance $MCD_{Tx/Rx,ij}$ is given as

$$MCD_{Tx/Rx,ij} = \frac{1}{2} \begin{pmatrix} \sin(\theta_{Tx/Rx,i})\cos(\phi_{Tx/Rx,j}) \\ \sin(\theta_{Tx/Rx,i})\sin(\phi_{Tx/Rx,j}) \\ \cos(\theta_{Tx/Rx,i}) \end{pmatrix} - \begin{pmatrix} \sin(\theta_{Tx/Rx,j})\cos(\phi_{Tx/Rx,i}) \\ \sin(\theta_{Tx/Rx,j})\sin(\phi_{Tx/Rx,i}) \\ \cos(\theta_{Tx/Rx,j}) \end{pmatrix}$$ (5)

First, we randomly choose the $K$ positions of cluster centroids $[c_1^{(0)}, c_2^{(0)}, \ldots, c_K^{(0)}]$ as the initial state. Then, we start iterations in the following steps:

1. Assign MPCs to cluster centroids and store indices:
   \[ F_1^{(i)} = \arg\min(P_1 \cdot MCD(X_l, c_k^{(i-1)})) \]
   \[ F^{(i)} = [F_1^{(i)}, F_2^{(i)}, \ldots, F_L^{(i)}], c_k^{(i)} = \text{Indices}(F_l = k) \] (6)
   where $F_l^{(i)}$ is the cluster index of the $l$th MPC in the $i$th iteration. $c_k^{(i)}$ is the set of MPCs indices belonging to the $k$th cluster in the $i$th iteration.

2. Recalculate positions of cluster centroids $c_k^{(i)}$ from the allocated MPCs to coincide with the clusters’ centers of gravity:
   \[ c_k^{(i)} = \frac{\sum_{j \in c_k^{(i)}} (P_j \cdot X_j)}{\sum_{j \in c_k^{(i)}} P_j} \] (7)

3. If $c_k^{(i)} = c_k^{(i-1)}$ for all $k = 1, \ldots, K$, then we go out of the iterations. Otherwise, we keep performing iterations up to the setting maximum iteration times.

Finally, we obtain the optimal cluster set. Here, the maximum iteration times were set as 100.

2.1.2. Kalman Filter Part

State model $g_c^{(n)}$ is assumed as follows:

$$g_c^{(n)} = \Phi g_c^{(n-1)} + N_1 = I_5 \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot g_c^{(n-1)} + \Delta^{(n)[n-1]} + N_1$$ (8)

where $\Phi$ is the state-transition matrix, $I_i$ is the identity matrix with $i$ dimensions, $\otimes$ is the product of the Kronecker matrix, $N_1$ is the state noise, and $\Delta^{(n)[n-1]}$ is the gap-compensation matrix, which is given as

$$\Delta^{(n)[n-1]} = [\Delta T_{i,1}^{(n)[n-1]} \Delta \theta_{T,i}^{(n)[n-1]} \Delta \phi_{T,i}^{(n)[n-1]} \Delta \theta_{R,i}^{(n)[n-1]} \Delta \phi_{R,i}^{(n)[n-1]}] \otimes [0 \ 1]^T$$ (9)

The observation model is the information of cluster centroids themselves, $\mu_c^{(n)}$, which is given as

$$\mu_c^{(n)} = I g_c^{(n)} + N_2 = I_5 \otimes [1 \ 0] \cdot g_c^{(n)} + N_2$$ (10)

where $N_2$ is the observation noise.

The update step of KF is

$$K_k = P_\Phi^{(n)[n-1]} f^T (f P_\Phi^{(n)[n-1]} f^T + R)^{-1}$$ (11)

$$S_c^{(n)} \approx S_c^{(n)[n-1]} + K_k (\mu_c^{(n)} - I g_c^{(n)})$$ (12)
\[ P_{\Phi}^{(n)} = (I - K_k) P_{\Phi}^{(n|n-1)} \]  \hspace{1cm} (13)

The predict step of KF is
\[ S_{\Phi}^{(n|n-1)} \approx \Phi S_{\Phi}^{(n-1)} + \Delta^{(n|n-1)} \]  \hspace{1cm} (14)

\[ P_{\Phi}^{(n|n-1)} = \Phi P_{\Phi}^{(n-1)} \Phi^T + Q \]  \hspace{1cm} (15)

where \( P_{\Phi} \) is the error covariance matrix, \( K_k \) is the optimal Kalman gain, and \( Q \) and \( R \) are the noise-covariance matrices.

It is noted that the state space model was chosen for the basic model description since it is suitable for linear movement [18]. Therefore, in the processing, \( Q \), \( R \), and \( P_{\Phi}^{(0)} \) were set as the identify matrices.

2.2. Clustering and Tracking Results

The spatial distributions of clusters in LoS and NLoS conditions are shown in Figure 5a,b, respectively. In the LoS condition, one cluster survives during the whole movement near the main path on 0°. In the NLoS condition, on the other hand, no cluster can keep surviving during the whole movement. Some clusters with a shorter lifetime can be observed. Their angles and power smoothly change with the movement. It is noted that those clusters with less than 5% normalized power were removed in the figures to make the main contributed clusters clearer for observation. Ignored clusters’ contributions to the channels are slight, and most of them have a short lifetime that only exists only in 1 snapshot. However, in the following statistical works, all of them are considered.

Figure 5. Clustering and tracking profiles of measurement clusters. (a) Cluster-tracking profiles in LoS condition. (b) Cluster-tracking profiles in NLoS condition. (c) Birth and death histogram of clusters per snapshot. (d) cumulative density functions (CDF) of cluster lifetime distances (m).
From Figure 5c, we see the birth and death histograms of clusters per snapshot that were calculated, which show the birth and death rate of clusters of all snapshots during movement. The total number of newly appeared clusters per snapshot is slightly larger than that of newly disappeared clusters per snapshot. Obviously, the change of 1 or 2 clusters per snapshot takes more probabilities. The polynomial fitting of both birth and death rate is also given in the figure.

The cumulative density functions (CDFs) of cluster lifetime distances are shown in Figure 5d. In our former work, Reference [9], the sizes of visibility regions (VRs) are calculated from the results by KPM algorithm, which are shown in blue and green curves in this figure. VR is a concept proposed in COST 2100 [20], which is an indicator of where corresponding clusters are active. In linear mobile measurements, VR diameters are the same as the lifetime distances of clusters. From this figure, we can see the effect of the newly introduced KF algorithm. Especially in LoS, most clusters show shorter lifetime distances, less than 1 m. In addition, the corresponding log-normal fit is given.

3. Mobile Channel Simulations by IMT-2020 Channel Model

The channel model is used to meet the requirements of evaluating IMT-2020 candidate radio-interface technologies (RITs) by allowing the realistic modelling of propagation conditions for radio transmissions in different environments [3]. Generally, a GBSM channel model is built in a 2D/3D co-ordinate system. The path loss model, shadow fading, small-scale parameters of clusters in typical scenarios (such as indoor hotspots (InH), UMa, urban micro (UMi), and rural macro (RMa)) are analyzed and collected from many empirical measurements. This can simulate and reproduce channel information that is similar with that in real conditions. In this section, the IMT-2020 channel model is used to simulate the measured channels. It follows the typical GBSM channel-model procedures in ITU-R M. 2135 [2], 3GPP TR 38. 900 [21], and TR 36. 873 [22]. To get closer to the mobile channels, the spatial-consistency feature was implemented, which is based on the primary module of the IMT-2020 channel model. Simulation results, including moving trajectory, path loss model, and angle evolution of clusters, are given, as are comparisons between measurements and simulations with and without this feature.

3.1. Simulation Generation Procedures

The generation of a channel model is divided in three main parts: general parameters, small-scale parameters, and coefficient generation (see Figure 6).
The first part, the generation of general parameters, includes the initial setup of the whole system, and how to find the corresponding path loss model and its large-scale parameters. Users must choose one of four typical scenarios: InH, UMa, UMi, and RMa. Base station (BS) and user terminal (UT) antenna-array details need to be set, such as the number of antennas, 3D locations, antenna field patterns, and array geometries. The speed and direction of UT motion, center frequency, and bandwidth must be set, too. Then, the corresponding path loss model with the formulas is formed. Finally, large-scale parameters are calculated, including root mean square delay spread, root mean square angular spreads, Ricean K-factor, and shadow fading.

In the second part, small-scale parameters are calculated. This is a group of parameters that represents the characteristics of dynamic clusters in the model, which include the delay, power, AoA, AoD, EoA, and EoD of each cluster. After this step, we can obtain the full information of the model clusters.

In the third part, initial phases are generated randomly at first. Each ray $n$ of cluster $m$ has four initial phases: $\{\Phi^{\theta}_{n,m}, \Phi^{\phi}_{n,m}, \Phi^{\theta}_{n,m}, \Phi^{\phi}_{n,m}\}$ for four different polarization combinations ($\theta$, $\phi$, $\theta$, $\phi$). Their distribution is uniform within ($-\pi, \pi$). Then, CIRs, also known as channel coefficients, are calculated. Finally, we apply path loss and shadowing to the CIR generation formula. For a channel simulation with $S$ antenna elements at the Tx side and $U$ antenna elements at the Rx side, the equation of generating a channel CIR is expressed as

$$H_{u,s} = \frac{F_{r,s}}{M} \sum_{m=1}^{N} \left[ F_{t,s} \theta(\theta_{n,m,EA}, \Phi_{n,m,AA}) \right]^T \left[ \exp(j\Phi^{\theta}_{n,m}) \sqrt{k-1} \exp(j\Phi^{\phi}_{n,m}) \right] \left[ F_{t,s} \theta(\theta_{n,m,EO}, \Phi_{n,m,AO}) \right] \left[ \exp(j\Phi^{\theta}_{n,m}) \sqrt{k-1} \exp(j\Phi^{\phi}_{n,m}) \right]$$

$$\cdot \exp(j2\pi \frac{\hat{r}_{t,s} \cdot \hat{d}_{t,s}}{\lambda_0}) \exp(j2\pi \frac{\hat{r}_{t,s} \cdot \hat{d}_{t,s}}{\lambda_0})$$

where $F_{r,s}$ and $F_{t,s}$ are the field patterns of receiver antenna element $u$ in the direction of spherical basis vectors $r$ and $s$, respectively, while $F_{t,s}$ and $F_{t,s}$ are the field patterns of transmitter antenna element $s$ in the direction of spherical basis vectors $t$ and $s$, respectively. $\hat{r}_{t,s}$ and $\hat{r}_{t,s}$ are the spherical unit vector with azimuth departure angle $\Phi_{n,m,EO}$ and elevation departure angle $\theta_{n,m,EA}$ at the Tx side, and with azimuth arrival angle $\Phi_{n,m,AA}$ and elevation arrival angle $\theta_{n,m,EO}$ at the Rx side, respectively. The Doppler frequency component depends on arrival angles AoA and AoE, and UT velocity vector $\vec{v}$ with speed $v$, travel azimuth angle $\phi_{v}$, elevation angle $\theta_{v}$, and other details can be found in Reference [3].

To bring the channel model closer to a realistic mobile scenario, a feature called spatial consistency was selected. The initial states of cluster delay, power, and angles were generated based on the steps of the primary module. Then, at moment $t_{k}$, the delay of the $n$th cluster is given as

$$\tau_{n}(t_{k}) = \tau_{n}(t_{k-1}) - \frac{\hat{r}_{r,s}(t_{k-1}) \cdot \hat{v}(t_{k-1})}{c} \Delta t$$

(17)
\[
\tau_n(t_k) = \tau_n(t_k) - \min \{ \tau_n(t_k) \}
\]

where \( \dot{r}_{rx,n}(t_{k-1}) \) is the spherical unit vector that consists of \( \theta_{n,EoA}(t_{k-1}) \), and \( \phi_{n,AoA}(t_{k-1}) \)

\[
\dot{r}_{rx,n}(t_{k-1}) = \begin{bmatrix}
\sin(\theta_{n,EoA}(t_{k-1})) \cos(\phi_{n,AoA}(t_{k-1})) \\
\sin(\theta_{n,EoA}(t_{k-1})) \sin(\phi_{n,AoA}(t_{k-1})) \\
\cos(\theta_{n,EoA}(t_{k-1}))
\end{bmatrix}
\]

\( \sigma(t_{k-1}) \) is the velocity of Tx in 3D, which is given as

\[
\sigma(t_{k-1}) = [V_X(t_{k-1}) \ V_Y(t_{k-1}) \ V_Z(t_{k-1})]^T
\]

There are no more updates to cluster-power generation. This changes by updating delays. With regard to angle generation, the updated AoD and EoD formulations are given as

\[
\phi_{n,AoD}(t_k) = \phi_{n,AoD}(t_{k-1}) + \sigma'(t_{k-1}) \cdot \alpha \frac{180}{\pi} \Delta t
\]

\[
\theta_{n,EoD}(t_k) = \theta_{n,EoD}(t_{k-1}) + \sigma'(t_{k-1}) \cdot \beta \frac{180}{\pi} \Delta t
\]

where \( \alpha, \beta, \sigma'(t_{k-1}), \) and \( r'_{n,3D}(t_{k-1}) \) are

\[
\alpha = \begin{bmatrix}
-sin(\phi_{n,AoD}(t_{k-1})) \\
\cos(\phi_{n,AoD}(t_{k-1})) \\
0
\end{bmatrix}
\]

\[
\beta = \begin{bmatrix}
\cos(\phi_{n,AoD}(t_{k-1})) \cos(\theta_{n,EoD}(t_{k-1})) \\
\cos(\phi_{n,AoD}(t_{k-1})) \sin(\phi_{n,AoD}(t_{k-1})) \\
-sin(\theta_{n,EoD}(t_{k-1}))
\end{bmatrix}
\]

\[
\sigma'(t_{k-1}) = [V_X'(t_{k-1}) \ V_Y'(t_{k-1}) \ V_Z'(t_{k-1})]^T
\]

\[
r'_{n,3D}(t_{k-1}) = r_{n,3D}(t_{k-1}) \sin(\theta_{n,EoD}(t_{k-1}))
\]

More details of other angles’ generation can be found in Reference [3]. After updating all this cluster information, the channel coefficient can be formed with Equation (16).

### 3.2. Simulation Results

A single-cell MIMO system was assumed at 3.5 GHz with 200 MHz bandwidth in a UMa NLoS scenario. At the Tx side, a UPA with 128 cross-polarized antennas was used; at the Rx side, a single-antenna ODA was used. Channel realizations in simulations were based on 12,000 drops. It is noted that the number of antennas is 1 on the Rx side since only the angles at the departure side could be observed. Moreover, the multiantenna array used at the Rx side in our measurement can be seen as an equivalent single antenna with an omnidirectional antenna pattern.

The MS moving trajectory is shown in Figure 7a. The Rx moves along in the direction of the arrow. Each direction takes 3000 drops. Based on Tx and Rx locations and the setting scenario, the path loss model is plotted in Figure 7b. The slope of the path loss model changes with the turning of directions. Blue bubbles show corresponding shadowing fading \( c_{SF} = 6 \) dB. Figure 7c shows the AoD and EoD evolution profiles of the first three clusters (with the three largest powers). Cluster angles change direction when the moving route changes, and the evolution process turns smoothly and
continuously. We also calculated the standard deviations (STD) of the delay, and the AoD and EoD from the measurements and simulations with and without the spatial-consistency feature (see Figure 7d). The simulation group without the spatial-consistency feature always has the largest values. Compared to the STD from the simulation group without this feature, the simulation group with it gets closer to the measurement group, which implies that spatial consistency improves simulation performance.

Figure 7. Profiles of the IMT-2020 channel simulations in UMa, 3.5 GHz, and STD comparisons among measurements and simulations with and without spatial-consistency feature. Note: AoD, azimuth angle of departure; EoD, elevation angle of departure; STD, standard deviation. (a) MS moving trajectory. (b) Simulation path loss model. (c) AoD and EoD of first three clusters (three largest powers). (d) Delay STD, AoD, and EoD.

4. Gradient Boosted Decision Tree-Based Model

4.1. Model Description

Generally, a decision-tree algorithm can be drawn as a flowchart that is easy to read and understand, and it can mainly be divided into two kinds: classification tree and regression tree. Boosting algorithms are a family of algorithms that can promote weak learners to strong learners. The gradient boosting machine (GBM) can use different learning methods than basic learners. GBDT is a GBM that uses the decision tree as the basic learner. It is a machine learning algorithm that is used for regression issues, originated and developed by L. Breiman and J. H. Friedman [25–27]. It is an ensemble learning method that is used to explain and predict statistical models, and it is flexible enough to fit different types of data, such as nonlinearities and interactions [28]. It sets each feature in each leaf node for stating the different characteristics of the dataset, and it goes through all child
datasets to build the tree for predicting. In contrast to other decision-tree algorithms, GBDT reduces the residual in each training iteration by compared to previous training results. It creates a new mathematical model in each iteration to lower the residual. There, it can better lower the possibility of overfitting that another decision tree has [29].

GBDT can be seen as an addictive model $F(x; w)$ that is accumulated by $T$ trees

$$F(x; w) = \sum_{i=0}^{T} a_i h_i(x; w) = \sum_{i=0}^{T} f_i(x; w) \tag{27}$$

where $x$ are the input samples and $w$ are the splitting variables, $h$ is the regression-tree function, and $a$ is the weight of each tree.

Optimized model $F^*$ can be obtained by minimizing loss function $L(\cdot)$

$$F^* = \arg \min_F \sum_{i=0}^{N} L(y_i, F(x_i; w)) \tag{28}$$

where $y_i$ represents the $i$th output from the model.

According to the first-order Taylor expansion, the best steepest-descent step direction $y_i$ can be calculated as

$$\tilde{y}_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)} \tag{29}$$

Optimized solution $w^*$ can be obtained from

$$w^* = \arg \min_{w, \beta} \sum_{i=1}^{N} (\tilde{y}_i - \beta h_i(x_i; w))^2 \tag{30}$$

Then, we can obtain optimized line search $\rho^*$ by

$$\rho^* = \arg \min_{\rho} \sum_{i=0}^{N} L(y_i, F_{t-1}(x_i) + \rho h_i(x_i; w^*)) \tag{31}$$

Finally, the approximation is updated by

$$f_t = \rho^* h_t(x; w^*)$$
$$F_t = F_{t-1} + f_t \tag{32}$$

4.2. Numerical Results

Simulation and measurement data were used to train and validate the GBDT-based model. 80% of the data are training samples, and 20% are validating samples. Training samples are used to train the model, and validating samples are used to check its accuracy. To train the simulation data, we had to reconstruct CIRs in the time domain with MPC delay and power as input. Training output is the cluster delay, AoD, and EoD. To train the data collected by measurement, CIRs are ready-made as input, and the output is from the results given by the KPMKF process, as shown in Section 2. The depth of trees was set as 100, and the maximum iteration is set as 200.

To validate the GBDT-based model, RMSE was used to measure the differences between predicted and observed values, which is calculated by

$$R = \sqrt{\frac{\sum_{n=1}^{N}(y(n) - \hat{y}_{sim/mca}(n))^2}{N}} \tag{33}$$
$y^{(n)}$ is the $n$th results predicted by the GBDT-based model, while $y^{(n)}_{\text{sim/nea}}$ is the $n$th results from simulations/measurements. $N$ is the number of drops. The smaller the RMSE is, the better the fit between the predicted and observed values is.

Figure 8a displays an example of AoD results by GBDT to show the degree of fitting. The training and validating samples, and the AoDs of the three clusters, are from the above Section 3 simulation in NLoS conditions, which are shown as solid curves. The dotted lines are the fitted results calculated by the GBDT-based model. The fitted curves are very close to the samples.

The RMSEs of the measurements and simulations are shown in Figure 8b. This shows that the GBDT validating results generally fit well with simulation/measurement results. Compared to the measurement RMSEs, those of the simulations are smaller. Moreover, the GBDT-based model performs better in an LoS condition than in an NLoS condition.

![Figure 8a](image1.png)

**Figure 8a.** AoD example of degree of fitting.

![Figure 8b](image2.png)

**Figure 8b.** RMSE.

5. Conclusions

In this paper, a series of cluster analyses of mobile UMa channels are presented, which were based on measurements and simulations. A UMa massive MIMO pedestrian mobile measurement at 3.5 GHz, which was performed with a virtual-measurement method, was described and analyzed. With the KPMKF algorithm for clustering and tracking, the spatial distributions of clusters in LoS and NLoS conditions were shown, respectively. From the figures, we could observe the cluster birth and death process and their spatial evolution during movement. The cluster birth and death histogram per snapshot was shown. The total snapshots of newly appeared clusters per snapshot was slightly larger than that of newly disappeared clusters per snapshot. The change of one or two clusters per snapshot takes more probabilities. In addition, the CDF of lifetime distances and VR diameters in Reference [9] were shown. In an LoS condition, most clusters showed shorter lifetime distances, which were less than 1 m.

To get closer to the above channel conditions, channel simulations were implemented with the IMT-2020 channel model using the newly introduced spatial-consistency feature. With the simulation results, the evolution of generated clusters could be observed. They evolve smoothly when UT turns the direction. Additionally, comparisons between measurements and simulations with and without this feature were given by STD. The results showed that simulation with this feature was closer to the measured channels.

The GBDT-based model, which is used for training and validating the above results from measurements and simulations, was introduced. We used the RMSE to check the accuracy of the model. Compared to the measurements, simulation RMSEs were smaller. Additionally, the GBDT-based model had better performance in an LoS condition than in an NLoS condition. Generally, it is a promising model for characterizing cluster evolution in linear mobile scenarios.
Author Contributions: Conceptualization, C.W. and J.Z.; methodology, C.W. and J.Z.; software, C.W. and G.Y.; validation, C.W., J.Z., and G.Y.; formal analysis, C.W. and J.Z.; investigation, C.W. and J.Z.; resources, C.W. and J.Z.; data curation, C.W. and J.Z.; writing—original draft preparation, C.W.; writing—review and editing, C.W., J.Z., and G.Y.; visualization, C.W.; supervision, J.Z.; project administration, J.Z.; funding acquisition, J.Z.

Funding: The research was supported by the National Science and Technology Major Project of the Ministry of Science and Technology (Grant No. 2018ZX03001031), the Key Program of the Beijing Municipal Natural Science Foundation (Grant No. L172030), and the project from the Beijing Municipal Science and Technology Commission (Grant No. Z181100003218007).

Acknowledgments: The authors thank Fredrik Tufvesson, Jose Flordelis, Huilui Ma, and Jian Zhang for active discussions.

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

References


