Position and LIDAR-Aided mmWave Beam Selection using Deep Learning

Marcus Dias*, Aldebaro Klautau*, Nuria González-Prelcic† and Robert W. Heath Jr.†
* Federal University of Pará, Belem, PA, 66075-110, Brazil, marcus.dias@itec.ufpa.br,aldebaro@ufpa.br
† The University of Texas at Austin, ECE Dept., Austin, TX, 78712-1084, USA, {ngprelcic,rheath}@utexas.edu

***Abstract*—One issue in the design of modern communication systems is how to benefit from the increasing variety of sensor signals and sophisticated machine learning algorithms. We recently described how LIDAR (light detection and ranging) on a vehicle can be used for line-of-sight detection and to reduce the overhead associated with link configuration in millimeter wave communication systems. LIDAR is widely used in autonomous driving for high resolution mapping and positioning. In this paper, we present new LIDAR-based features for machine learning and compare the previously proposed distributed architecture with two centralized schemes: using a single LIDAR located at the base station (BS) and fusing LIDAR data from neighboring vehicles at the BS. We also quantify the advantages of LIDAR-based solutions over solutions based on connected vehicles informing their positions. We use deep convolutional neural networks to process images composed of LIDAR data and/or positions. Using co-simulation of communications and LIDAR in a vehicle-to-infrastructure (V2I) scenario, we find that the distributed LIDAR-based architecture provides robust performance irrespective of car penetration rate, outperforming the single LIDAR at BS and position-based solutions. We noted that, under the simulated conditions, the benefits of a centralized data fusion over distributed processing are not significant, meaning that machine learning for line-of-sight detection and beam selection can be conveniently executed at vehicles equipped with LIDAR.**

I. Introduction

Connected and automated vehicles generate a large amount of sensor data. Sharing this data requires high rate wireless links, which can be provided by millimeter wave (*mmWave*) systems [1]. The sensor data may be used for an additional application: improving mmWave communication. Given the lack of models that relate sensor data to communication channels, it is sensible to try a data-driven approach such as deep neural networks [2]. In this paper, we propose and evaluate a framework for reducing overheads in establishing mmWave communication links in vehicular networks.

The LIDAR (light detection and ranging) is one of the most sophisticated sensors used in automated driving. A LIDAR uses a laser to scan an area and measure the time delay from the backscattered signal. This data is then converted into points in space and interpreted as three-dimensional (3D) images with pixels indicating relative positions from the sensor. LIDAR data can be exploited without additional cost for improved communications when it is already used on a automated vehicle for mapping, positioning, or obstacle detection.

Most prior work in machine learning-based beam selection has used the position information obtained from vehicles [1], [3]–[9]. That work demonstrates that position information can be used by a centralized machine learning (ML) agent, executed for example at the base station (BS), to reduce the overhead required to establish mmWave links. The performance of such centralized architectures is limited by the *penetration rate*, which is the fraction of connected with respect to all circulating vehicles [9]. We proposed a new approach in [10]: having a decentralized architecture based on LIDAR. This distributed architecture is advantageous given that fully connected is a long term scenario.

In this paper, we develop centralized LIDAR architectures, which may also include position information provided by satellite-based navigation. A single LIDAR at the BS is attractive for automated driving in that all the vehicles in the cell may share the data from the infrastructure-based LIDAR, reducing the need for expensive LIDARs on each car. We evaluate centralized and distributed architectures applied to beam selection, which more specifically consists here in choosing the set of best beam pairs. We also discuss the distinction between line-of-sight (LOS) and non-LOS (NLOS) situations. LOS detection is useful because beam selection is easier in the LOS setting. Our main conclusions are that, in spite of the lower cost of the single LIDAR at BS and position-based solutions, they are outperformed by distributed LIDAR-based architecture in the two investigated problems. We also noted that a centralized data fusion did not bring significant improvements over the distributed processing.

The rest of this paper is organized as follows. Section II describes the system model. Section III summarizes the simulation methodology, describing the machine learning modeling and how we obtained paired simulations of LIDAR and communication systems. The machine learning techniques used in this paper are described in Section IV. Section V describes the simulations results and Section VI presents the conclusions.

II. System Model

We consider a downlink orthogonal frequency-division multiplexing (OFDM) mmWave system with analog beamforming [11]. Both transmitter and receivers have antenna arrays with only one radio frequency (RF) chain and fixed beam codebooks. We use ray-tracing (RT) to simulate the channel

This material is based upon work supported in part by the National Science Foundation under Grant No. ECCS-1711702 and Grant No. CNS-1731658, as well as gifts from Nokia Bell Labs and Toyota ITC. The work of A. Klautau was supported in part by CNPq, Brazil (201493/2017-9/PDE).

and combine the RT output data with a *wideband* mmWave *geometric channel model* [8]. Assuming R_c rays (or multipath components, MPC) per transmitter / receiver pair, the information collected from the RT outputs for the r-th ray of a given pair is: complex path gain α_r, time delay τ_r, angles ϕ_r^D, θ_r^D, ϕ_r^A, θ_r^A, corresponding respectively to azimuth and elevation for departure and arrival, and whether it is a LOS or NLOS situation. The channel model at the time instant corresponding to the n-th symbol vector is [11]:

$$\mathbf{H}[n] = \kappa \sum_{r=0}^{R_c-1} \alpha_r g(nT - \tau_r)\mathbf{a}_r(\phi_r^A, \theta_r^A)\mathbf{a}_t^*(\phi_r^D, \theta_r^D), \quad (1)$$

where $\kappa = \sqrt{N_t N_r}$, N_t and N_r are the numbers of antennas at the transmitter and receiver, respectively, $g(\tau)$ is the shaping pulse (a raised cosine with roll-off of 0.1), $T = 1/B$ is the symbol period and B the bandwidth, $\mathbf{a}_r(\phi_r^A, \theta_r^A)$ and $\mathbf{a}_t^*(\phi_r^D, \theta_r^D)$ are the steering vectors at the receiver and transmitter for the r-th MPC, respectively.

To simplify notation, when using uniform planar arrays (UPAs), we adopt the corresponding Kronecker products of steering, precoding and combining vectors to represent the 2D arrays as 1D vectors [8]. Assuming OFDM with K subcarriers and that $\mathbf{H}[n]$ can be fairly represented by its first L taps, the frequency-domain channel at subcarrier k is

$$\mathbf{H}[k] = \sum_{n=0}^{L-1} \mathbf{H}[n] e^{-j\frac{2\pi k}{K}n}. \quad (2)$$

We assume beam codebooks $\mathcal{C}_t = \{\mathbf{f}_1, \cdots, \mathbf{f}_{|\mathcal{C}_t|}\}$ and $\mathcal{C}_r = \{\mathbf{w}_1, \cdots, \mathbf{w}_{|\mathcal{C}_r|}\}$ at the transmitter and the receiver sides, where $|\mathcal{C}_t|$ and $|\mathcal{C}_r|$ are their corresponding cardinalities. For a given pair (p, q) of vectors, representing precoder $\mathbf{f}_p$ and combiner $\mathbf{w}_q$, the received signal at subcarrier k is $\mathbf{s}[k] = \mathbf{w}_q^H \mathbf{H}[k]\mathbf{f}_p$, where H denotes conjugate transpose. The beam selection is guided by the normalized signal power

$$\mathbf{y}_{(p,q)} = \sum_{k=0}^{K-1} |\mathbf{w}_q^H \mathbf{H}[k]\mathbf{f}_p|^2 \quad (3)$$

and the *optimum* beam pair is $\widehat{(p,q)} = \arg\max_{(p,q)} \mathbf{y}_{(p,q)}$. In this paper, the goal of beam selection is to recommend a set $\mathcal{B} = \{(p_i, q_i)\}_{i=1}^M$ of pairs such that $\widehat{(p,q)} \in \mathcal{B}$.

III. Simulation Methodology

We executed *paired* simulations that used the same 3D environment to gather data from LIDAR and mmWave channels via ray-tracing. For that we adopted a simulation methodology as proposed in [12] using traffic, ray-tracing and LIDAR simulators: *Blender Sensor Simulation* (BlenSor) [13], *Simulation of Urban MObility* (SUMO) traffic simulator [14] and Remcom's Wireless InSite for ray-tracing, respectively. The first two are open-source softwares. A Python software (*orchestrator*) invokes SUMO and save its outputs (vehicles positions, orientations, etc.), to be utilized by the others simulators. Then, the orchestrator invokes the ray-tracing and the LIDAR simulators to obtain paired results.

The LIDAR uses a laser to scan an area and measure the time delay from the backscattered signal. This data is then converted into points in space, known as *point cloud* data. The LIDAR sensor has been mainly used for obstacle detection, navigation of autonomous ground vehicles, 3D mobile data collection and mapping applications. But the BlenSor software can be adapted to be used in the context of communication systems.

Fig. 1 illustrates examples of the generated LIDAR point clouds for LOS and NLOS, respectively. The line between transmitter and receiver is also shown and its proximity to LIDAR points indicate a NLOS condition. The circles centered at the receiver correspond to ground reflections, which are filtered out in this work. In the ML simulations, the actual LOS or NLOS class "label" is assigned by the result of the RT simulator.

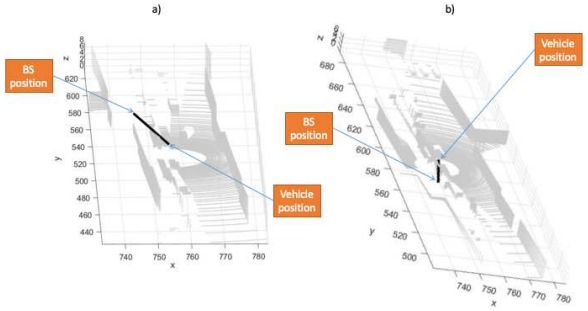

Fig. 1. Two examples of point clouds from a LIDAR located on vehicle, which also carries a mmWave radio close to the LIDAR. The line between the LIDAR and the mmWave radio at the BS is also shown. In a) this line does not encounter any LIDAR point. This suggests a mmWave LOS link, which may be checked using a ray-tracing simulation. In b) the line traversing points detected by the LIDAR indicates a NLOS condition.

The realism of the RT simulation depends on the details of the scenario, which include geometric aspects (number of object faces, etc.) and materials (electromagnetic parameters, etc.). In this paper we used an improved version of the dataset adopted in [10]. Instead of representing vehicles with boxes, the simulations used models of vehicles (cars, buses and trucks) that were created using the open source 3D creation suite Blender [15]. After a model is constructed with Blender, it is exported to the DXF format, which is supported by both Wireless InSite and Blensor. In their corresponding InSite's configuration, multiple materials were associated to the new models. For example, glass was used for the vehicle windows and metal to the rest of the vehicle, as shown in Fig. 2. It should be noted that the new (more realistic) models increase RT simulation time due to the larger number of 3D faces per object. We placed receivers and LIDARs on top of a maximum of 10 vehicles in each scene snapshot. Fig. 2 shows an example with three "connected" vehicles.

Fig. 3 depicts the adopted urban canyon 3D scenario, which is part of Wireless InSite's examples and represents a region of Rosslyn, Virginia, which was also used e. g. in [5], [16].

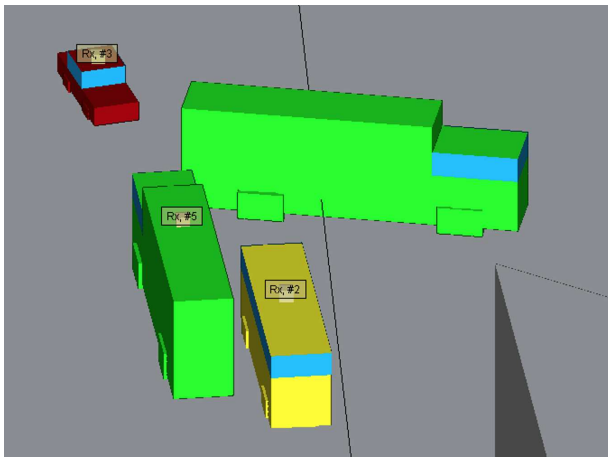

Fig. 2. This paper uses more realistic 3D models of vehicles composed with different materials (metal and glass) than the ones used in [12]. In this figure three vehicles are "connected" (indicated as Rx #) and equipped with mmWave receivers and LIDAR.

The chosen *study area* is a street corresponding to a rectangle of approximately 23×250 m^2.

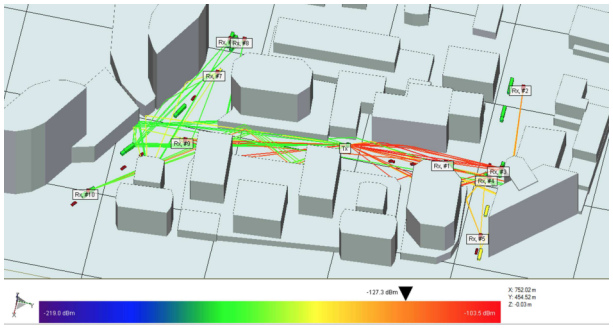

Fig. 3. 3D urban canyon scenario used for the InSite ray-tracing simulations.

Properly positioning the LIDAR at the BS and vehicles is important. Fig. 4 depicts the results considering a LIDAR at a pole with three distinct heights (1, 2 or 4 m). As expected, the LIDAR position significantly impacts the resulting scans and the number of points (obstacles) obtained. It is out of the scope of this paper to optimize this height and we assumed that the LIDAR at the BS is located at the same height of the antenna array, at $z = 4$ m. The LIDAR on a vehicle is on the top of its roof.

To collect the paired data for a given scene, we execute a single RT simulation and several LIDAR simulations. More specifically, to enable simulations of the distributed, centralized architectures, we run Blensor to obtain a LIDAR scan for each connected car in a scene, as depicted in Fig. 5. We have the same number of RT and LIDAR simulations, only for the case of the architecture with a single LIDAR at the BS. Table I summarizes the most important simulation parameters.

IV. Deep Learning Model

We evaluate LOS detection and beam selection using the following ML architectures:

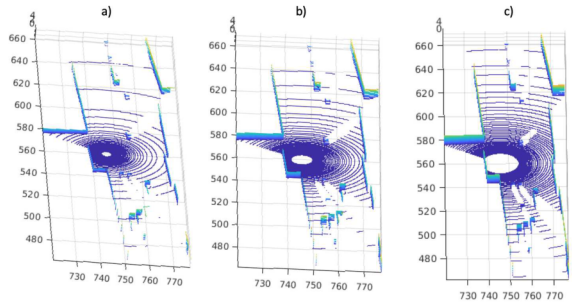

Fig. 4. Scans for a LIDAR at a BS assuming the LIDAR is located (in the center of the circles) at different heights z: a) 1 m, b) 2 m and c) 4 m.

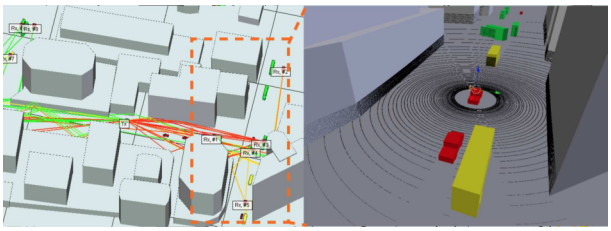

Fig. 5. Example of a Blensor LIDAR scan (right) performed in one of the scenarios of the simulation (left).

- *LIDAR distributed* [10]: each connected vehicle runs a ML algorithm and makes its decision based on its own LIDAR data and position information broadcasted by BS;
- *LIDAR centralized*: the BS runs the ML algorithm with data sent by connected vehicles and makes its decision based on the fused data;
- *LIDAR@BS*: there is a single LIDAR at the BS, and the BS runs a ML algorithm to makes decision based on its own LIDAR data;
- *Position*: the BS receives the positions of connected vehicles, executes the ML algorithm and decision.

The DNN architecture utilized in both problems (LOS detection and beam selection), is composed of 13 layers trained with Keras' *Adadelta* optimizer, in which 7 of them are 2D *convolutional* layers with decreasing kernel sizes, from 13×13 to 1×1. We utilized regularization and *dropout* to decrease the effects of overfitting. For top-M classification, the output layer had a *softmax* activation function and a *categorical cross-entropy* as loss function [17]. For binary classification, the output layer and loss were *sigmoid* and *binary cross-entropy*, respectively [17].

In this paper, all architectures use ML input features that consist of a binary 3D array of dimension $\mathbf{G} = 20 \times 200 \times 10$ points. Fig. 6 depicts the extraction of features for the LIDAR and Position architectures. In the case of the LIDAR architectures, the elements of this arrays are derived from the point cloud $\mathbf{C}$ collected by the LIDAR. For the Position architecture, the array with features is created based on the positions and dimensions reported by the vehicles. We first describe the creation of features to the LIDAR architectures.

TABLE I
SIMULATION PARAMETERS.

Ray-tracing parameters	
Carrier frequency	60 GHz
RSU transmitted power	0 dBm
RSU antenna height	4 m
Antenna (Tx and Rx)	Isotropic
Propagation model	X3D
Terrain and city material	ITU concrete 60 GHz
Vehicle material	Metal and glass
Ray spacing (degrees)	1
Num. L of strongest rays	100
Diffuse scattering model	Lambertian
DS max. reflections ($N_{\max}^{\text{DS}}$)	2
DS coefficients (S)	0.4 (concrete), 0.2 (metal)
Traffic parameters	
Number of lanes	4
Vehicles	car, truck, bus
Lengths, respectively (m)	4.645, 12.5, 9.0
Heights, respectively (m)	1.59, 4.3, 3.2
Probabilities, respectively	0.45, 0.3, 0.25
Average speed (m/s)	8.2
Sampling period T_{sam} (s)	3
LIDAR parameters	
Model	Velodyne HDL-64E2
Scan distance (m)	120
Scan angle resolution	0.17
Angle of vision	360 degrees

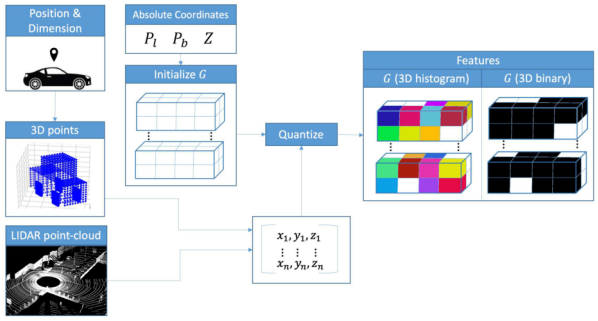

Fig. 6. Feature extraction for the Position and LIDAR architectures. The n 3D points correspond to a point-cloud (LIDAR architectures) or were created from information about vehicles (position, size and orientation). The features are represented by a 3D array $\mathbf{G}$.

The point cloud $\mathbf{C}$ provides obstacle distances with respect to the LIDAR position P_ℓ. For both the distributed and centralized architecture P_ℓ is the vehicle position P_v, and for the LIDAR@BS architecture it is the base station position P_b, which is provided by GPS. The absolute positions $\mathbf{A} = \mathbf{C} + P_\ell$ are quantized into a grid representing the BS coverage zone Z [10]. In summary, the absolute positions are mapped to a grid $\mathbf{G}$ using the provided absolute coordinates. Instead of using $\mathbf{G}$ as a 3D histogram as proposed in [10], we alternatively convert the histogram elements into binary values, using 1 whenever the histogram counter was larger than 0 or, otherwise, keeping the 0 value. This binary and 3D histogram representations provided similar results and we use the former in this paper due to its lower requirement of memory for storage.

For the Position architecture, given the positions and dimensions of the connected vehicles, the corresponding elements of $\mathbf{G}$ are represented with 1, and 0 elsewhere.

V. NUMERICAL RESULTS

The mmWave data [18] is composed by 3582 snapshots, each with several receivers. From this data we selected $N_L = 11,691$ LOS and $N_N = 4,048$ NLOS channel examples. For the beam selection experiments, as in [10], were chosen kept only the codevectors that were chosen as $\widehat{(p,q)}$ more than 100 times in the training set. This procedure led to $|\mathcal{C}_t| = 24$ and $|\mathcal{C}_r| = 11$, respectively. Hence, the number of classes for top-M classification is 264. Also, the beam selection experiments used N_L and N_N examples in the LOS and NLOS evaluations, respectively, while LOS detection used all $N_L + N_N$ examples. For all experiments we created disjoint test and training sets with 20% and 80% of the examples, respectively. The following paragraphs present results for the two LIDAR architectures, distributed and centralized, and for Position using 100% penetration.

Table II shows the accuracy of the ML systems for LOS detection. The distributed architecture outperformed the other methods. These results will be discussed after the presentation of beam selection results.

TABLE II
SIMULATION RESULTS FOR LOS VS NLOS BINARY CLASSIFICATION.

Architecture	Accuracy
LIDAR distributed	0.912
LIDAR centralized	0.799
Position (penetration=100%)	0.797
LIDAR@BS	0.702

Fig. 7 and Fig. 8 show the beam selection top-M accuracy for LOS and NLOS, respectively. In LOS cases, all architectures other than LIDAR@BS perform equivalently, given that the best beam primarily depends on the position of the receiver with respect to the BS. However, in NLOS and LOS detection in Table II, even considering a 100% penetration rate, the distributed outperformed the Position architecture. This may be related to the fact that the Position takes in account only the (connected) vehicles while a LIDAR is capable of incorporating information about the surroundings (buildings, etc.). For the LIDAR@BS architecture, the BS only collects information about the environment close to it. Vehicles far from the BS may be poorly represented or even invisible to the system.

Fig. 9 and Fig. 10 depict the corresponding *achieved throughput ratio* (RT)

$$\text{RT} = \frac{\sum_{i=1}^{N} \log_2(1 + y_{\widetilde{(p,q)}})}{\sum_{i=1}^{N} \log_2(1 + y_{\widehat{(p,q)}})}, \tag{4}$$

where N is the number of test examples and $\widetilde{(p,q)}$ is the best beam pair in $\mathcal{B}$. The overhead from beam selection for $M = 10$, decreases by a factor of 26.4, however the RT indicates a reduction to 64% of the achievable throughput for NLOS in the

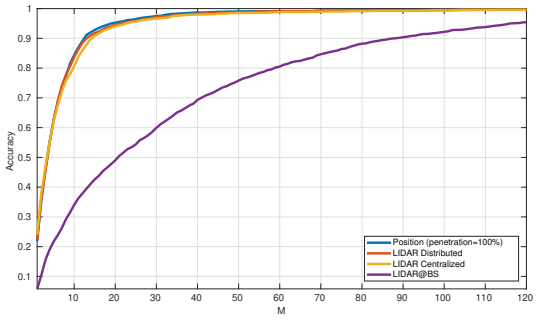

Fig. 7. LOS top-M classification for beam selection with 264 beam-pairs, for $M = 1, \ldots, 120$.

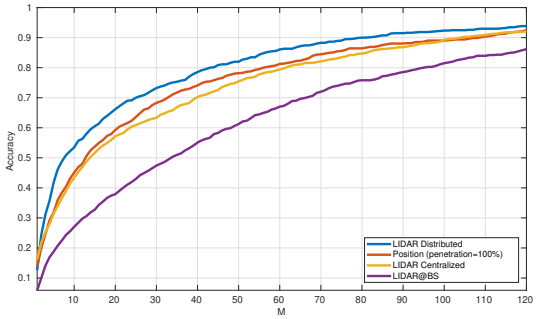

Fig. 8. LOS top-M classification for beam selection with 264 beam-pairs.

LIDAR centralized architecture. For NLOS and the Position architecture, RT reaches e. g. 92% for $M = 75$.

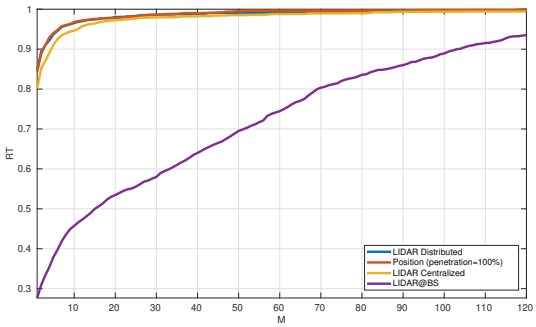

Fig. 9. LOS throughput RT for beam selection with 264 beam pairs.

VI. Conclusions

This paper expanded the work in [10] and compared the distributed LIDAR-based architecture and centralized ones. We also assessed the results obtained with LIDAR or the vehicles position provided by their GPS. The results indicated that for the NLOS case, the distributed LIDAR-based architecture outperforms the centralized and position-based alternatives. The performance of the centralized data fusion may had been limited by the adopted ML technique and coarse resolution used for the 3D grid. Future work includes developing and testing alternative feature representations and neural network models for fusing LIDAR, position and other information sources.

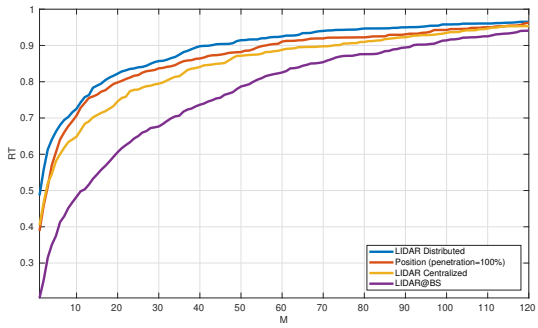

Fig. 10. NLOS throughput RT for beam selection with 264 beam pairs.

References

[1] N. González-Prelcic, A. Ali, V. Va, and R. W. Heath, "Millimeter-wave communication with out-of-band information," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 140–146, Dec. 2017.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[3] A. Capone, I. Filippini, V. Sciancalepore, and D. Tremolada, "Obstacle avoidance cell discovery using mm-waves directive antennas in 5G networks," in *Proc. IEEE 26th Ann. Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Aug. 2015, pp. 2349–2353.

[4] W. B. Abbas and M. Zorzi, "Context information based initial cell search for millimeter wave 5G cellular networks," in *2016 European Conf. on Networks and Comm. (EuCNC)*, Jun. 2016, pp. 111–116.

[5] J. C. Aviles and A. Kouki, "Position-Aided mm-Wave Beam Training Under NLOS Conditions," *IEEE Access*, vol. 4, pp. 8703–8714, 2016.

[6] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Beam design for beam switching based millimeter wave vehicle-to-infrastructure communications," in *Proc. IEEE Int. Conf. on Comm. (ICC)*, May 2016, pp. 1–6.

[7] A. Loch, A. Asadi, G. H. Sim, J. Widmer, and M. Hollick, "Mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training," in *9th COMSNETS*, Jan. 2017, pp. 1–8.

[8] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse Multipath Fingerprinting for Millimeter Wave V2I Beam Alignment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4042–4058, May 2018.

[9] Y. Wang, A. Klautau, M. Ribero, M. Narasimha, and R. Heath, "Mmwave vehicular beam training with situational awareness by machine learning," in *Proc. of IEEE GLOBECOM*, Dec. 2018.

[10] A. Klautau, N. González-Prelcic, and R. W. Heath, "LIDAR data for deep learning-based mmWave beam-selection," *IEEE Wireless Communications Letters*, pp. 1–1, 2019.

[11] A. Ali, N. González-Prelcic, and R. W. Heath, "Millimeter Wave Beam-Selection Using Out-of-Band Spatial Information," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1038–1052, 2017.

[12] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *Proc. Inf. Theory Appl. Workshop (ITA)*, 2018.

[13] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Proc. 7th Int. Symp. Advances in Visual Computing*, 2011, pp. 199–208.

[14] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.

[15] https://www.blender.org.

[16] S.-C. Kim, B. J. Guarino, T. M. Willis, V. Erceg, S. J. Fortune, R. A. Valenzuela, L. W. Thomas, J. Ling, and J. D. Moore, "Radio propagation measurements and prediction using three-dimensional ray tracing in urban environments at 908 MHz and 1.9 GHz," *IEEE Trans. Veh. Technol.*, vol. 48, no. 3, pp. 931–946, May 1999.

[17] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2017.

[18] https://www.lasse.ufpa.br/raymobtime.