An Efficient Ray Validation Technique for Ray-Tracing in Urban Microcellular Environments

Sajjad Hussain* and Conor Brennan†

Abstract

Ray-tracing propagation models for urban microcellular environments become time consuming as the validation of higher order of ray interactions presents a significant computational overhead. This letter presents a time efficient ray validation algorithm for ray-tracing models. The model presented in this letter uses a pre-computed database of building faces that are completely or partially visible to each face and vertical edge of all the buildings in the radio environment. This database is used to readily validate the ray-segments between the transmitter and receiver. The validation results show considerable time saving over previous models.

Keywords: Radio propagation, Ray tracing, Uniform Theory of Diffraction

1 Introduction

Image-based ray tracing models have been well-received for radio channel modeling in recent years [1]. These models represent the radio waves as connected straight line segments, also called rays, that follow the laws of Geometric Optics and Uniform Theory of Diffraction. A visibility algorithm is recursively applied to find an ordered list, or image-tree, of all visible surfaces that can produce the rays for the required order of ray interaction. Geometrical checks are performed to validate that each of rays' associated ray-segments is not obstructed by buildings or other objects in the environment. These geometrical checks make ray-tracing models computationally inefficient for large or dense urban environments when a high order of ray interaction must be computed to ensure accurate channel modeling.

There are several pre-processing techniques, mostly adopted from computer graphic algorithms, that can reduce the visible surfaces computations for ray tracing models [2–11]. These techniques either reduce the number of objects on

---

*S. Hussain is with National University of Sciences and Technology, Islamabad, Pakistan. Email: sajjad.hussain2@seecs.edu.pk

†Conor Brennan is with the School of Electronic Engineering, Dublin City University, Ireland. email:conor.brennan@dcu.ie.

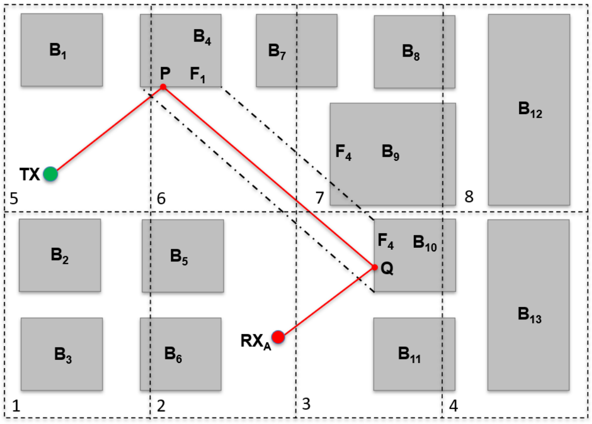

Figure 1: Example scenario for efficient ray validation test

which the ray-object intersection test is performed or accelerate the ray-object intersection test [12–15]. The angular Z-buffer (AZB) technique [2] checked only the buildings and objects inside the illuminated region formed by wall reflection or vertical edge diffraction. Alwajeeh et al. in [3] pre-computed the precise two-dimensional (2-D) polygon with the fewest nodes for each potential ray. These polygons represented the exact horizontal area in which a ray can propagate, thus eliminating the need for a ray-object intersection test. A tree structure was proposed by space partitioning techniques like Binary Space Partitioning (BSP) [4] and Linear Space Partitioning (LSP) [5] to distinguish between building surfaces (or nodes) that are visible and non-visible to a root node. Similar to this, convex cell partitioning method by Teh et al. [6] divided the radio environment into three-dimensional (3D) convex spaces that were effectively used to sort objects according to how far they were from a point. Pre-processing is required for each transmitter location for the methods presented in [2, 3] because they are source-dependent. Additionally, it is difficult to implement the models [3–6] in outdoor scenarios with many building walls and objects.

The environment was divided into grid squares using space division techniques like bounding box [7] and bounding volume hierarchy (BVH) [8], and the locations of the buildings and objects were then mapped to the appropriate grid squares. Only the buildings and objects inside the grid squares that a ray passed through needed to be checked for obstructions, which accelerated the ray validation. The number of building walls and objects that still need to be tested for ray validation, however, would result in a lower time acceleration in dense urban areas. Eid et al. in [9] applied geometric algebra (GA) methods to determine the ray-object intersection. By permuting a limited number of surfaces from the set of all surfaces in the environment, the model generated all feasible reflection sequences of the required order of ray interaction for validation using GA. Due to the high potential number of reflection sequences, the model is not feasible for dense urban areas with numerous building walls. Liu et al. in [10] pre-computed the Orientation Face Set that contains an ordered list of all the faces that are visible to each face in half-space defined by outward face normal. Ng. et al. in [11] used Potential Visibility Set (PVS) that builds a large table of "Yes" or "No" entries based on intra-visibility between the building faces. The above models in references [10, 11] speed-up the creation of image-tree, but the geometrical tests for ray validation must still be performed.

Previous works by the authors have presented efficient pre-processing techniques including image polygons mapping for mobile receivers [16, 17], intra-visibility matrix to readily compute image-tree for mobile transmitter [18], and visibility matching technique to reduce the image-tree in vehicular scenarios [19]. The methods presented in previous works are able to reduce the size of the image-tree or decrease the time it takes to compute it, but they still require that each image in the image-tree be tested for ray validation, which can be computationally intensive for dense urban environments. In this letter, a straightforward and source location-independent pre-processing method is proposed to speed up the *ray-object intersection test* in image-based ray-tracing models. The list of all the building faces that are either partially or completely visible to each building face and vertical edge (collectively referred to as nodes) of all the buildings in the environment are pre-computed. Additionally, the list of building faces that block the visibility of partially visible faces for each building node in the environment is pre-computed. This so-called reflecting surfaces identification (RSID) database reduces the number of ray-object intersection tests required for ray validation, resulting in faster ray-tracing computations. The proposed model thus complements the existing techniques as it addresses the specific issue of computational overhead during the ray validation step, which is a key bottleneck in image-based ray-tracing models. By incorporating the proposed technique into previous works [16–19], the computational complexity in radio channel modeling for both cellular and vehicular radio networks can be further reduced. Please note that while the proposed model can be used to speed up ray-tracing computations in both rural and urban areas, it is more effective in urban microcellular environments where a large number of higher-order wall reflection and vertical edge diffraction rays exist and may significantly affect received power.

2 Ray tracing Algorithm

The problem being addressed in this paper can be understood with reference to Fig. 1. A key task in ray-tracing is the validation of each ray, that is confirming that all of its constituent segments have unobstructed line-of-sight and do not intersect any buildings. A ray with N reflections will have $N+1$ such segments, and the example in Fig. 1 shows a ray with 2 reflections and hence 3 segments. This paper addresses the specific problem of validating the *building-to-building* segments (such as between reflection points Q and P), a problem which constitutes most of the work (as a ray with N reflections will contain $N-1$ of such segments). The simplest approach to validate such a ray segment is a brute force approach. In the context of our example this will involve checking whether the line segment PQ intersects with any building in the environment (in this case buildings B_1 to B_{13}, except for B_4 and B_{10} upon which the points P and Q reside). A more efficient technique is to use space division which partitions the environment using a grid. Only buildings in grid-squares through which the segment PQ passes need to be checked, in this case buildings B_7, B_8, B_9 and B_{11} which is clearly more efficient than brute force. Our proposed method makes use of RSID database. This is an extension of the environmental intra-visibility matrix introduced in [18]. The intra-visibility matrix is computed in an off-line phase and contains, for each building node, a list of all other nodes that are visible (partially or wholly) within the environment. The matrix also stores the angular range over which nodes are visible to each other. Such knowledge, computed only once for an environment, can be used to promptly compute the image-tree for each transmitter location for a mobile transmitter scenario as shown in [18].

The RSID database, proposed in this letter, extends intra-visibility matrix by adding further information which needs only be computed once in an off-line phase. For any pair of nodes that are only partially visible to each other the list of buildings that are causing this partial blockage are identified and stored. The construction of this database is discussed in section 2.1. For now we assume it is available and just concentrate on its usage. Returning to our motivational example consider Fig. 1. It can be seen that face F_4 of building B_{10} is partially visible to face F_1 of building B_4 as it is obstructed by face F_4 of building B_9. It is assumed that this visibility relation between F_4 of B_{10} and F_1 of B_4 is known *a priori* via the proposed RSID database. Under the proposed approach validating the ray-segment PQ requires checking *only* face F_4 of building B_9 for obstruction. For an N point ray involving $N-1$ building-to-building reflections, this approach can be applied to each building-to-building segment as appropriate, greatly reducing the computing time required. A flowchart depicting the ray-tracing algorithm using RSID database is shown in Fig. 2.

2.1 Reflecting Surfaces Identification Database

In order to compute the RSID database, a one-off pre-processing of all the building faces and edges in the environment is performed. It should be noted that once the RSID database is compiled for a given environment, it can be repeatedly used for any transmitter and receiver configurations in that environment. For each building face the list of both completely and partially visible faces is computed in two steps. First, a list of all the buildings that are visible to a given face or edge is determined by checking for visibility in half plane defined by the outward face normal. Then all the faces of visible buildings are checked one by one for complete visibility. Two faces will be completely visible to each other if there are no buildings inside the parallelogram formed by joining the outer edges of the faces. In case, two faces are not completely visible to each other then the building faces that obstruct the visibility are computed.

This is further explained with the help of Fig. 3. We are interested to find

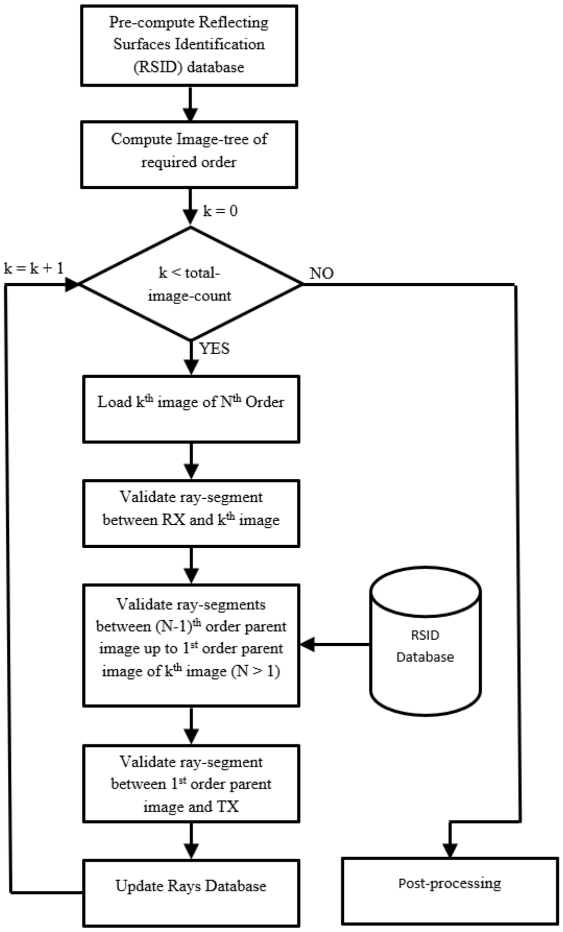

Figure 2: Flowchart for efficient ray validation using RSID database

the list of all the completely or partially visible faces to face F_1 of building B_1. It can be seen that faces F_2 and F_3 of building B_2, and F_3 of B_3 are visible to F_1 of B_1. In the next phase, each visible face is tested for complete or partial visibility by checking the region formed by joining its edges to the edges of face F_1 of B_1. Face F_3 of B_3 is partially visible as faces F_1, F_2 and F_3 of B_2 block the visibility between these faces as shown. It can also be seen that face F_3 of B_2 is completely visible to F_1 of B_1 as no building obstruct the region between these two faces as shown by solid red parallelogram. A similar approach is used to find the completely or partially visible faces to a vertical edge of a building. In order to confirm complete visibility in case of edge-to-face, the triangular region formed by joining the edge to both the edges of the face must not contain any other building or objects as shown for edge E_1 of B_1 on right side of Fig. 3.

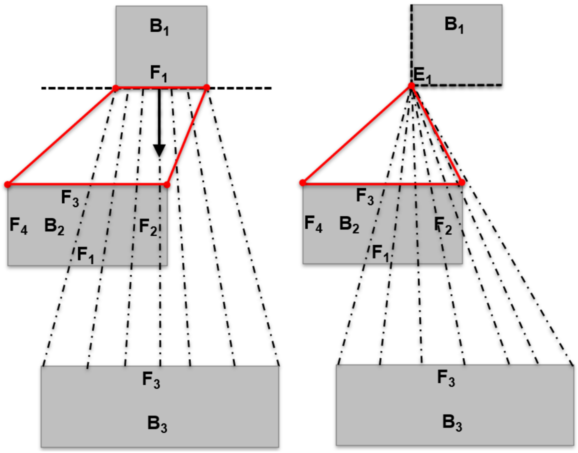

Figure 3: RSID Database computation

Faces F_1, F_2 and F_3 of B_2 block the visibility between edge E_1 of B_1 and face F_3 of B_3. The complete and partial visibility data for each face and vertical edge of all the buildings is computed and stored in the RSID database.

Table 1: Ray-tracing time comparison for different types of rays.

Time	R^2 Rays	R^3 Rays	R^4 Rays	$D-R$ Rays
Using space-division technique [8] (seconds)	38.98	120.66	261.61	$1,430.25$
Total time using this paper (seconds)	24.40	68.05	147.90	996.70
Percentage Reduction	37.40%	43.60%	43.46%	30.31%

3 Results

The method presented in this letter is used to compute radio coverage maps in an urban environment taken from Munich [20] that consists of 67 buildings as shown in Fig. 4. The height of the transmitter antenna is 10m. A half-wave dipole antenna with 20 dBm output power is assumed at 1900 MHz. A rectangular grid of $6,074$ receiver points at 5m x 5m resolution is placed over the environment. The height of the receiver points is 1.6m. The RSID database is computed for the environment. The rays up to 4th order of reflections, and diffraction followed by reflection are computed for all the receiver points. The rays are validated in the horizontal plane followed by validation in the vertical plane to compute the actual three-dimensional (3D) ray paths [17]. The proposed algorithm is also applicable for ray validation in full 3D ray-tracing models in which rays are launched in both elevation and azimuth planes. An Intel® Core™ i7 computer

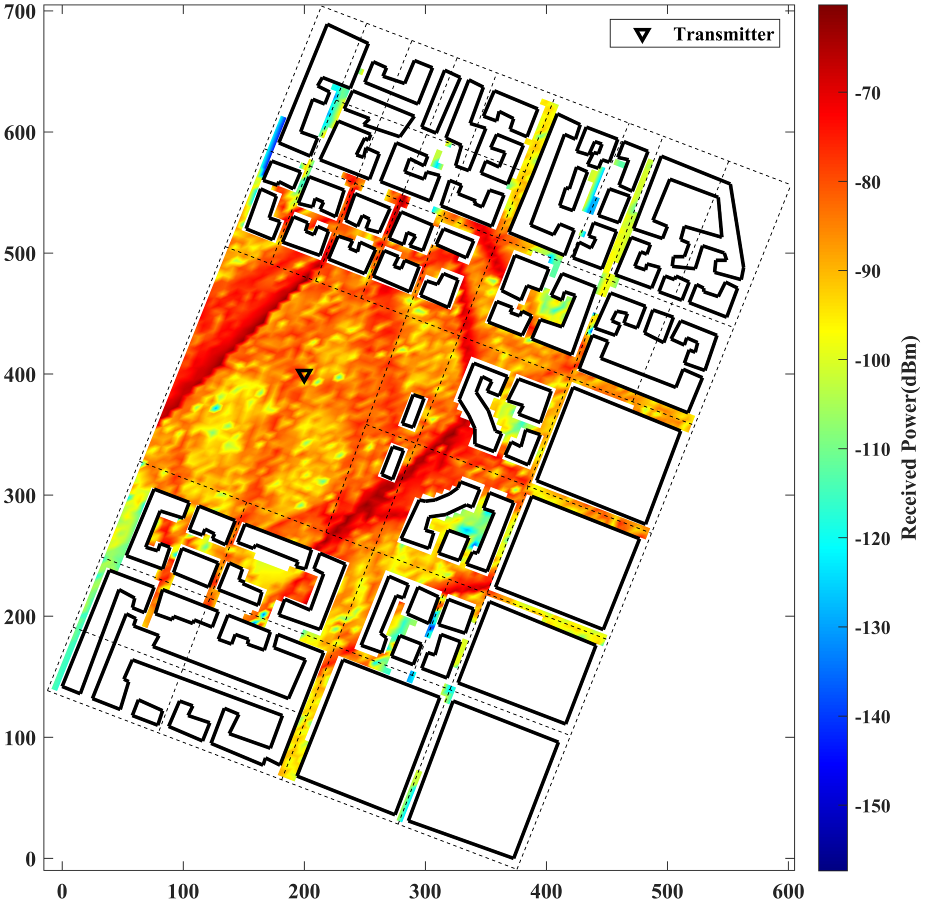

Figure 4: Radio coverage map for $D-R$ rays in an outdoor microcellular environment

with 16 GB RAM is used in simulations. Fig. 4 shows the radio coverage map computed using diffraction followed by reflection ($D-R$) rays. It can be observed that the $D-R$ rays can provide coverage even in far-off regions as expected.

In order to compare the computational gain achieved using the proposed model, the ray-tracing time is compared against that required by an adaptive multi-level space-division technique [8]. The environment is divided into a rectangular grid such that no more than three buildings are mapped to a grid square (see grid of dashed lines in Fig. 4). The time comparison for the proposed model against the space-division method for different types of rays is listed in Table 1. It can be seen that the ray-segment validation time for wall-to-wall reflections is decreased by 37.40%, 43.60%, and 43.46% for 2nd, 3rd, and 4th order reflections respectively. In the case of diffraction followed by reflection rays, the

Figure 5: Munich database for path loss computations

proposed model takes 996 seconds for ray-segments validation as compared to $1,430$ seconds for the space-division method. This corresponds to a reduction of computation time by 30.31%. The ray-segment validation time for diffraction rays is significantly longer than that for reflection rays. This is due to the fact that diffracted rays spread in all directions, resulting in a larger image-tree that must be validated for each receiver position.

In order to show the accuracy of the proposed model, path loss is compared against measured data. The measurement data was collected in Munich at 947 MHz by Mannesmann Mobilfunk GmbH as part of COST Action 231 [20]. The transmitter and receiver antenna heights were 13 m and 1.5 m, respectively. The measurement area is 2400 m by 3400 m and contains $2,088$ buildings. As seen in Fig. 5, a total of 67 receiver points are chosen. For each receiver point, rays up to the sixth order of reflections and a single diffraction are computed. These Non Line-of-sight (NLOS) receiver points are chosen because they are close to the transmitter and are expected to receive significant contributions from reflections and vertical edge diffraction rays, which the proposed model effectively computes. The comparison of simulated and measured path loss is shown in Fig. 6. The observed error has a mean of 0.49 dB and a standard deviation of 5.07 dB. This shows the proposed model can offer sufficient accuracy in path loss prediction in urban microcellular environments. Future work will include incorporating the proposed technique into previous works [18, 19] and comparing its performance in challenging V2V scenarios with novel vehicular channel prediction techniques such as [21], which use a combination of ray-

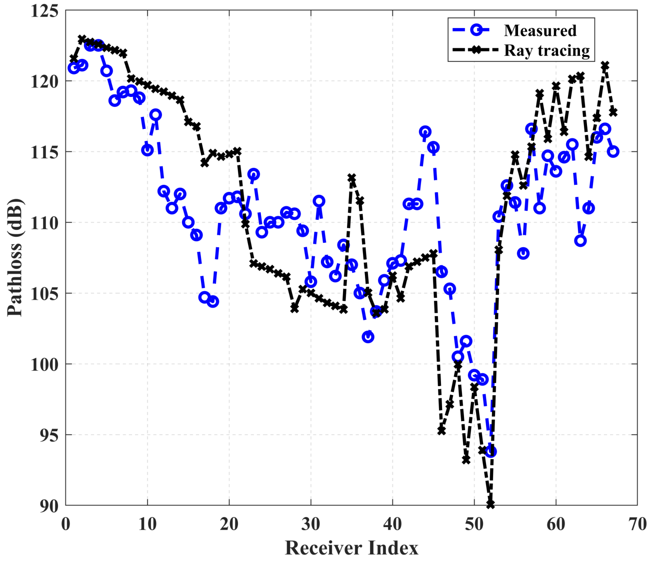

Figure 6: Comparison of predicted path loss against measured data

tracing and propagation graph model to lower computational complexity.

4 Conclusion

A new efficient ray-segment validation algorithm for radio propagation modeling is introduced in this letter. The model creates the RSID database to pre-compute the buildings faces that are completely or partially visible to each face and vertical edge of all the buildings in the environment. The model expedites the ray-tracing by only checking visibility between relevant faces and edges. The proposed model reduced the ray validation time by up to 43% for higher order of specular reflections and by up to 30% for vertical edge diffraction contributions, and predicted path loss with sufficient accuracy which makes it a suitable candidate for radio propagation modeling in urban microcellular environments.

References

[1] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong and T. Kürner, "The Design and Applications of High-Performance Ray-Tracing Simulation Platform for 5G and Beyond Wireless Communications: A Tutorial," in IEEE Communications Surveys and Tutorials, vol. 21, no. 1, 2019.

[2] C. Saeidi and F. Hodjatkashani, “Modified angular z-buffer as an acceleration technique for ray tracing,” IEEE Transactions on Antennas and Propagation, vol. 58, no. 5, pp. 1822–1825, 2010.

[3] T. Alwajeeh, P. Combeau and L. Aveneau, ”An Efficient Ray-Tracing Based Model Dedicated to Wireless Sensor Network Simulators for Smart Cities Environments,” in IEEE Access, vol. 8, pp. 206528-206547, 2020.

[4] R. P. Torres, L. Valle, M. Domingo, and S. Loredo, “An efficient ray-tracing method for radio propagation based on the modified BSP algorithm,” Vehicular Technology Conference, 1999. IEEE VTS 50th, pp. 1967–1971, 1999.

[5] J. Bittner, J. Prikryl, and P. Slavik, “Exact regional visibility using line space partitioning,” Computers and Graphics (Pergamon), vol. 27, no. 4, pp. 569–580, 2003.

[6] C. Teh, B. Chung and E. Lim, ”An Accurate and Efficient 3-D Shooting-and- Bouncing-Polygon Ray Tracer for Radio Propagation Modeling,” in IEEE Transactions on Antennas and Propagation, vol. 66, no. 12, pp. 7244-7254, Dec. 2018.

[7] F. Aguado Agelet, A. Formella, J. M. Hernando Rabanos, F. Isasi de Vicente and F. Perez Fontan, ”Efficient ray-tracing acceleration techniques for radio propagation modeling,” in IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2089-2104, Nov. 2000.

[8] F. Weinmann, ”Adaptive and Automated Multilevel Uniform Space Division for Acceleration of High-Frequency Electromagnetic Simulations [EM Programmer’s Notebook],” in IEEE Antennas and Propagation Magazine, vol. 59, no. 1, pp. 130-138, Feb. 2017.

[9] Ahmad H. Eid, Heba Y. Soliman and Sherif M. Abuelenin, ”Efficient ray-tracing procedure for radio wave propagation modeling using homogeneous geometric algebra,” in Electromagnetics, vol. 40, no. 6, pp. 388-408, 2020.

[10] Z. . -Y. Liu, T. . -L. Zhang, L. . -X. Guo and J. Li, ”Effects of Diffraction and Ground Reflection on Ray-Tracing-Based Coverage Predictions in Urban Microcellular Environments,” 2018 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2018.

[11] K. H. Ng, E. K. Tameh, A. Doufexi, M. Hunukumbure and A. R. Nix, ”Efficient Multielement Ray Tracing With Site-Specific Comparisons Using Measured MIMO Channel Data,” in IEEE Transactions on Vehicular Technology, vol. 56, no. 3, pp. 1019-1032, May 2007.

[12] S. Hussain, “Efficient ray-tracing algorithms for radio wave propagation in urban environments,” Ph.D. dissertation, Dublin City Univ., Dublin, Ireland, 2017.

[13] M. F. Catedra, J. Perez, F.Saez de Adana, O.Gutierrez, “Efficient Ray-Tracing techniques for three-dimensional analyses of propagation in mobile communication: application to picocell and microcell scenarios,” IEEE Antennas and Propagation Magazine, Vol. 40, No. 2, pp. 15-28, April 1998.

[14] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, “Speed-up techniques for ray tracing field prediction models,” in IEEE Transactions on Antennas and Propagation, Vol. 57, No 5, pp. 1469 – 1480, May 2009.

[15] Z. Yun and M. F. Iskander, ”Ray Tracing for Radio Propagation Modeling: Principles and Applications,” in IEEE Access, vol. 3, pp. 1089-1100, 2015.

[16] S. Hussain and C. Brennan, ”An efficient ray-tracing acceleration technique for mobile receivers in urban environments”, Proc. Comput. Electromagn. Int. Workshop (CEM), pp. 47-48, Jun. 2017.

[17] S. Hussain and C. Brennan, “An efficient ray tracing method for propagation prediction along a mobile route in urban environments,” Radio Science, vol. 52, no. 7, pp. 862–873, 2017.

[18] S. Hussain and C. Brennan, ”Efficient Preprocessed Ray Tracing for 5G Mobile Transmitter Scenarios in Urban Microcellular Environments,” IEEE Transactions on Antennas and Propagation, vol. 67, no. 5, pp. 3323-3333, May 2019.

[19] S. Hussain and C. Brennan, ”A Visibility Matching Technique for Efficient Millimeter-wave Vehicular Channel Modeling,” in IEEE Transactions on Antennas and Propagation, May 2022, doi: 10.1109/TAP.2022.3178130.

[20] *”COST 231 Munich digital map and measurements”*. Accessed on: March 10, 2022. [online]. Available: https://propagationtools.com/wireless/cost-231-munich-digital-map-and-measurements/

[21] M. Gan, G. Steinböck, Z. Xu, T. Pedersen and T. Zemen, ”A Hybrid Ray and Graph Model for Simulating Vehicle-to-Vehicle Channels in Tunnels,” in IEEE Transactions on Vehicular Technology, vol. 67, no. 9, pp. 7955-7968, Sept. 2018.