IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 July 2023, accepted 21 August 2023, date of publication 31 August 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3310825

RESEARCH ARTICLE

Design and Implementation of an Innovative High-Performance Radio Propagation Simulation Tool

JOSEFA GÓMEZ, CARLOS J. HELLÍN, ADRIÁN VALLEDOR, MARCOS BARRANQUERO, JUAN J. CUADRADO-GALLEGO, AND ABDELHAMID TAYEBI

Computer Science Department, University of Alcalá, 28805 Alcalá de Henares, Spain

Corresponding author: Abdelhamid Tayebi (hamid.tayebi@uah.es)

This work was supported by the Program "Programa de Estímulo a la Investigación de Jóvenes Investigadores" of Vice Rectorate for Research and Knowledge Transfer of the University of Alcala and the Comunidad de Madrid, Spain, under Project CM/JIN/2019-028 and Project CM/JIN/2021-033.

ABSTRACT This paper presents the development and improvement of a high-performance validated radio propagation simulation platform. The design and implementation process of the proposed wireless network planning tool aims to achieve the best human-computer interaction. It takes as a base a previously developed version that allowed the visualization of geospatial data and the use of empirical and semi-empirical algorithms to analyze macrocellular environments. Now, the application also allows the use of the ray launching technique to obtain accurate predictions in urban environments, visualize multiple raster data and display the results dynamically while they are being calculated. The implementation of a 3D geometry reconstruction system has also been included to allow the automatic loading of geospatial information provided by open-source tools such as OpenStreetMap and LiDAR. The 3D reconstruction system allows the user to run simulations of any part of the world. The proposed platform can optimize the position of base stations providing the best coverage based on the constraints provided. The optimization module allows the design and planning of new cellular networks and also the extension of existing networks. The result is an innovative high-performance radio propagation simulation platform that can provide multiple functionalities through a friendly and interactive web interface.

INDEX TERMS Ray launching, user experience design, radio propagation, radio network planning.

I. INTRODUCTION

Recent years have seen a huge increase in the use of simulation tools for radio network planning and this trend seems to continue in the future. Software applications are used as an alternative to the expensive and humdrum measurement campaigns that, in the past, would have been required to study signal propagation in a particular area. Some of these applications have been developed [1], [2] using empirical and deterministic algorithms. Traditionally, empirical algorithms [3] have been chosen over deterministic ones to model mobile telecommunication systems for both rural and urban zones. With these algorithms, specific data of the area under analysis is interpreted statistically using variables like the average building height, average street width, etc. Some assumptions are made to simplify the models: antennas are assumed to be placed over elevated terrain and the receptors are treated as if they were behind a certain number of obstacles like buildings or mountains. In these situations, empirical algorithms provide reasonably good predictions. However, when the assumptions are critically opposed to reality, such as when the antenna is placed low enough, statistical simplifications do not work as well. In these cases, signal propagation is obtained through in-site measurements or using deterministic models, which require a realistic geometric and physical analysis of the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengqing Yun.

environment. One of the main types of asymptotic techniques for full-wave computational electromagnetic modeling is constituted by the ray tracing approach. This is based on the theory of geometrical optics. According to this theory, the electric field is discretized into rays and two different methods can be used in the ray tracing simulations: the image theory method and the shooting and bouncing ray method. The image theory method is more accurate since it computes the exact propagation paths from the transmitting antenna to the receiving antenna. The computational complexity is $O(N^k)$ where k is the number of specular reflections and N is the number of facets in the geometrical model. This is why the image theory method is not suitable to simulate large scenarios. On the other hand, the shooting and bouncing ray method launches rays from the transmitting antenna in every direction. This is also called the ray launching method. The rays propagate according to Snell's law and only the rays that intersect a receiving sphere are taken into account in the electric field calculation. A disadvantage of this method is that, since the receiving sphere has a certain radius, some ray paths can be inexact. Regarding the computational complexity, it is $O(NK)$, where K is the number of specular reflections and N is the number of rays generated. It is substantially lower than the complexity of the image theory method. In addition, in situations of non-line-of-sight, ray launching methods are chosen since the image theory method usually includes only a limited number of specular reflections [4]. Also, while ray tracing has typically been used for point-to-point multipath simulations, ray launching is more suitable for multipoint calculations. Another advantage of the shooting and bouncing ray method is that it allows the inclusion of diffraction and refraction, which the image theory method alone cannot accomplish. All this makes the shooting and bouncing ray method the best choice when analyzing large scenarios. This work focuses on the improvement of the user experience in cases when the user needs to analyze a large, complex environment. In these cases, the user has to wait to visualize the results because the calculations may take some time.

The proposed work aims to address this inconvenience by developing an interactive system that can display partial results (25%, 50%, 75%) even before the whole simulation has finished. As an example, one of the modules of the developed web simulation tool presented in this work provides antenna positioning optimization mechanisms, whose convergence process can be visualized in real-time. The paper describes how the developed simulation tool can perform this type of calculation and visualize the results interactively for the user, with a user-friendly and clean interface.

The first version of the radio propagation simulation platform developed by the authors was validated and presented in [5]. The main advantage of this tool was that the information on the environment was obtained from OpenStreetMap [6]. This platform provides very detailed geographic data from any part of the world required to calculate the propagation losses. Although it was initially based on a semi-empirical approach, it was improved by including several empirical methods and a ray launching module. This module takes advantage of the ray tracing programmable framework OptiX [7], through the PlotOptiX API (Application Programming Interface) package [8], optimized for massively parallel processing on GPUs (Graphics Processing Unit).

This paper describes the design and implementation process of the developed simulation platform from a human-computer interaction point of view. A review of the different tools available to improve the visualization of both geospatial data and the results of the radio propagation calculation is presented. The rest of the paper is organized as follows. Section II describes the state of the art of similar tools and equivalent simulators, both commercially and publicly available, and focuses on the contribution of the proposed simulation platform. Section III-B presents the main features of the high-performance radio propagation simulation platform and discusses the technology stack and libraries used. Finally, Section IV presents conclusions and future lines of work.

II. STATE OF ART

Besides the academic ray tracing simulators published in the literature, there are several recognized commercial simulators used by both industry and academia. The tutorial by He et al. [9] summarizes the features of the most relevant commercial and publicly available academic ray tracing-based simulators. There are also many professional software tools, both commercial [10], [11] and free [12], [13] normally based on geographic information systems [14] that use empirical methods to calculate the radio propagation. Table 1 includes a comparison of other similar applications that offer similar functionality, according to the following criteria:

1) Provides free access
2) Provides web or multiplatform access. It can be accessed from any web browser and also from mobile devices.
3) Allows interactive use
4) Allows easy to handle and edit map data.
5) Includes empirical or semi-empirical radio propagation algorithms
6) Includes ray launching or ray tracing algorithms
7) Uses graphics or visual methods to show simulation results
8) Considers multiple transmitting antennas
9) Uses 3D maps
10) Optimizes antenna positioning

According to these ten analyzed items, the most complete tool is the one proposed in this paper, since it combines all the evaluated functionalities. As it can be seen in Table 1, the optimization of the antenna's positioning (item number 10) is a feature included in the proposed tool that is not present in most of the other tools. That option has been implemented to design and plan new cellular networks and also to extend existing networks. It allows for optimizing the position of

certain antennas while keeping one or more antennas in fixed locations. The only tool in Table 1 that includes this feature is Forsk [15], rated with 8 points, which provides an automatic cell planning module for radio access network planning. The next highest-rated tool is WinProp [16], a commercial tool that allows representations on 2D and 3D maps and provides multiplatform access. Like Forsk, WinProp uses empirical and deterministic techniques to analyze both urban and indoor environments. A quite similar tool to the one developed is CloudRF [17], rated with 7 points. It offers a limited free web version but can be extended by paying a fee. After studying it, it is concluded that it does not offer a great user experience: the edition of elements that have already been created is quite difficult, and it is easier to delete them rather than edit them. On the other hand, it includes a large number of empirical algorithms that can be configured by the user. It also allows different layers to be displayed, but not in a very efficient and user-friendly way. Another similar tool is Xirio [18], rated also with 7 points. Like CloudRF, Xirio does not include ray-based methods to calculate radio propagation. It also includes various empirical models. The free version of the tool provides limited functionalities. One of its strong points is the option of working on several projects simultaneously. Among the free web-accessible tools are also Radio Mobile [19] and TowerCoverage [20]. The disadvantage of these tools is that they display the radio propagation results in only two colors, so scarce information is obtained from the simulations. In addition, when multiple base stations are considered in a simulation, the results are not displayed clearly due to the overlapping values. The results in those areas covered by two or more antennas are displayed in dark colors that do not correspond to any color scale. After analyzing some of the existing similar tools, the goal of the proposed tool is to cover all the evaluated functionalities and also to improve the user experience by developing an interactive system that overcomes the limitations of the other tools described above.

TABLE 1. Comparative table of similar tools and the developed tool.

Tool	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Total
Radio Mobile [19]	1	1			1						3
TowerCoverage [20]	1	1		1	1		1				5
Xirio [18]	1	1	1	1	1		1	1			7
QRadioPredict [21]	1			1	1						3
WaveSight [22]		1				1	1		1		4
WinProp [16]		1	1	1	1	1	1	1	1		8
RadioPlanner [23]			1	1	1		1	1			5
Probe [24]			1	1	1		1	1			5
Cellular Expert [10]		1	1	1	1		1	1	1		7
Wireless InSite [25]			1	1	1	1	1	1	1		7
Ranplan Professional [26]			1	1		1	1	1	1		6
GRASS-RaPlaT [27]	1	1	1	1	1		1				6
CloudRF [17]	1	1	1	1	1		1	1			7
CRFS [28]			1	1	1		1				4
Infovista [29]			1	1		1	1	1	1		6
Forsk [15]			1	1	1	1	1	1	1	1	8
CellPlan [30]			1	1	1	1	1	1	1		7
Edx Wireless [31]			1	1	1	1	1	1	1		7
Ibwave [32]			1	1	1	1	1	1	1		7
TAP [33]			1	1	1		1	1	1		6
Siradel [34]			1	1		1	1	1	1		6
IQ Link [35]	1		1	1	1		1	1	1		7
Pathloss [36]			1	1	1		1	1	1		6
Proposed Tool	1	1	1	1	1	1	1	1	1	1	10

III. OVERVIEW OF THE HIGH-PERFORMANCE RADIO PROPAGATION SIMULATION PLATFORM

The first version of the radio propagation simulation platform was validated with measurements in a real environment. Two different routes were measured and simulated and a good agreement between simulations and measurements results was found [5]. The first version included several empirical and semi-empirical models [2] and pursued the goals analyzed in Table 1. In the current version, the following functionalities were implemented to provide added value and to differentiate the proposed tool from the rest of the tools analyzed:

1) Display terrain-related information, e.g., height values, population density, terrain type, or any other raster input that the algorithms could require [14].
2) Improve the visualization of the results of radio propagation simulations in macrocellular environments based on empirical and semi-empirical algorithms. Represent the path loss on the map using color scales and graphs.
3) Make the interface more interactive by allowing interaction with the map and the configurations dynamically.
4) Make the manipulation of the elements created on the map easier.
5) Perform advanced antenna positioning optimization, by means of several types of genetic algorithms using additional libraries like jMetal, to provide the best coverage on a certain area [37].
6) Support incremental gathering of the data. For large calculations, the data may be processed in batches containing partial results. These batches are displayed on the map as they are being obtained so that the user can see the progress of the calculations.
7) Include a fast and accurate ray launching method to simulate both indoor and outdoor environments.
8) Reconstruct and visualize 3D maps for urban areas, in the case of using a ray launching propagation calculation.
9) Visualize ray launching simulation results on the 3D environment where they are calculated.
10) Display the elements involved in a ray launching simulation: emitter/s, receiver/s, a geometrical model

of the 3D environment, and the different types of rays: direct, reflected, transmitted, or diffracted.

11) Support interactions with the ray launching results. The user can inspect the rays through manual selection on the web interface. Filtering them with a simple query to only display the queried rays is also supported.

In the following subsections, these improvements are described in detail.

A. INTERFACE INTERACTIVITY IMPROVEMENTS

It is important to remark that the element manipulation in the current version is more flexible than in the first version. In the first version, the user created the areas and could not edit them, only delete them. Additionally, there was a problem with raster requests when two areas were overlapping. In the case of two or more large overlapped areas, they contained duplicated and redundant data, overlapping the images and creating a worse user experience. In the current version, if two areas are overlapping, they can be merged if the use case makes sense. For example, in requests for terrain heights or path loss. The areas can also be treated separately. This improves memory usage and reduces requests since only the data that are not already in the newly drawn area will be requested.

Regarding the interactive interface, many similar tools have an interface based on filling in forms with the required data and sending it to the tool's server. The results arrive in a report detailing the characteristics based on the request in an atomic way. However, in the current tool version, this interaction is dynamic. The users select and drag where they want to place the intervening elements. They can also draw the areas where the calculation will be performed. Geometries such as lines, circumferences, or rectangles encapsulating the desired area are used. Additionally, the application supports the use of forms to fill in the details of the simulation: geopositioning of the antennas or intervening elements, frequency of the antennas, polarization, and some additional characteristics. There is a button to switch between the use of forms and the interface on the map. Thus, the users are free to use the most convenient way to interact according to their use case.

This interface has been made possible thanks to the new stack of technologies employed. OpenLayers is used for moving the image processing responsibility from the server to the client. React is used to control and switch between different panels dynamically to provide a better user experience. The microservices that provide raster and optimization data have been refined. They can provide the data in different response types, so the web client can request partial responses for large areas.

B. ADVANCED ANTENNA POSITIONING OPTIMIZATION

Concerning the optimization of antenna positioning, it is performed by genetic algorithms. A back-end server receives the area where the transmitting antennas can be positioned, and the area to be optimized, providing the best coverage and the minimum path loss. This functionality allows for optimizing the position of certain antennas while keeping one or more antennas in fixed locations. It is important to point out that, since there can be several transmitting antennas, signals from these antennas may be out of phase with the candidate antenna, leading to destructive interference.

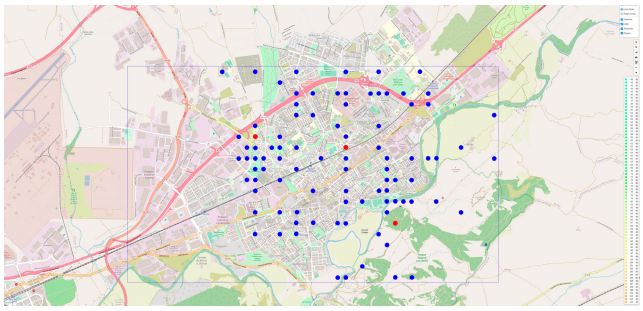

FIGURE 1. The blue points represent the positions that have been analyzed and the red points represent the current antenna positions that are being analyzed.

Genetic algorithms have been used to optimize the position of the base stations. The objective function is defined to minimize the path losses in the selected area, which are provided by the empirical/semi-empirical algorithms. The candidate positions are established in a grid throughout the area indicated by the user. This means that the candidate antenna positions can be any point within the area indicated by the user. Genetic algorithms are especially suitable for this type of application since input and output parameters are modeled by gens (the candidate locations). To carry out an optimization, the user selects the area to be optimized, which will display a form to be filled in with the necessary data. Additionally, the user can draw the area on the interactive 2D map.

Once the form is filled or the area is selected on the map, the data are sent to the server. The server returns the result, displaying a new area representing the partial optimization results, as shown in Figs. 1 and 2. As can be seen, the optimizations are displayed in real-time as they are being performed, showing the current best antenna locations. Note that the red points represent the current antenna positions, whereas the blue points represent the positions that have been previously analyzed during the optimization process. For this particular example, the positions of three antennas have been optimized.

The tool offers the possibility of selecting these two different areas since, normally, the transmitting antenna locations are restricted to certain zones. When performing the simulation, the genetic algorithm server sends instances of the evolution of the algorithm used. The client processes them and displays them dynamically and interactively on the map for the client's visualization. In the current version, the server uses the jMetal library [43], [44], [45], [46], [47], [48], a Java framework for multi-objective optimization using metaheuristics, for the computation of genetic algorithms.

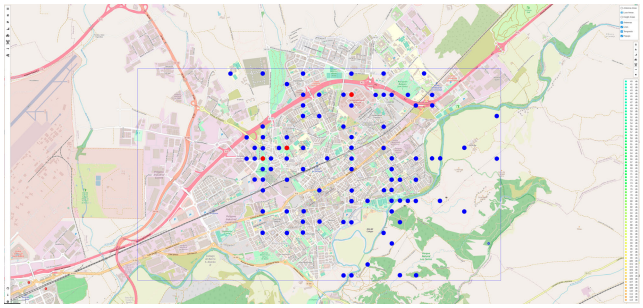

FIGURE 2. Optimization animation in another instant. The minimum distance between two candidate antenna positions in the optimization process is 30m.

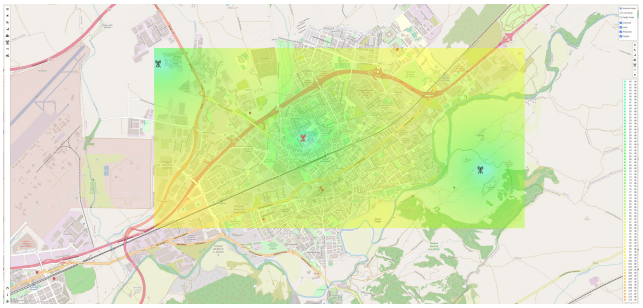

FIGURE 3. The position of a new antenna has been optimized.

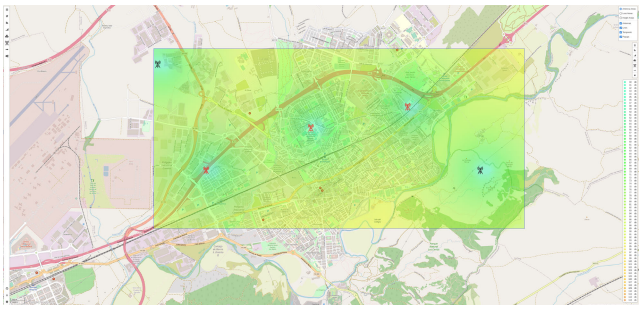

FIGURE 4. Results of the antenna optimization process.

The use of jMetal allows the use of evolutionary and genetic algorithms that are already developed and tested. jMetal is an open source project (https://github.com/jMetal). This implies that the algorithms have already been tried and tested not only by the creators themselves, but by other people who can contribute by improving the code, reporting bugs, etc. Another advantage is that development time is shorter, since that if these algorithms had to be programmed from scratch, it would take much longer. In addition, the algorithms included in jMetal have been proved to be efficient for the optimization of antenna locations. As demonstrated in [48], where a radio network design problem is addressed, satisfactory results are yielded using a multi-objective version of the algorithm CHC, which is a metaheuristic included in the jMetal framework. The experiments carried out in that work reveal that the algorithm provided by the framework is particularly adequate, because the existing results have been improved: the optimal solutions are obtained using a lower number of function evaluations and, instead of a single solution, the Pareto optimal set is obtained, thus allowing the decision maker to choose the best coverage solution.

Figs. 3 and 4 show two examples of this advanced functionality. In both cases, there is a fixed existing antenna depicted in black color. In Fig. 3 the best position for a new antenna (represented in red color) has been found and in Fig. 4 three new antennas have been located so that the path losses are minimum in the selected area.

C. RECONSTRUCTION OF THE 3D ENVIRONMENT

With respect to the 3D maps required to perform the ray launching simulations, they are integrated into the 2D maps of the application in a user-friendly way. Thus, if the users want to run a ray launching simulation, they only have to select the area to analyze, the position of the emitter/s, and the position of the receiver/s. When a user selects a rectangular area within an urban environment on the map, the buildings adjacent to the boundary of this rectangle are included since buildings and other objects near the boundary can have an impact on radio coverage. Additionally, the user can configure basic and advanced simulation parameters such as the frequency, polarization, or electromagnetic properties (electrical conductivity and relative permittivity) of the materials that compose the buildings. Then, the application downloads the urban data of the selected area from OSMBuildings [49] and LiDAR [50].

These data are in 2D, so a 3D reconstruction of the geometries is applied to obtain the 3D scenario that is going to be simulated [39]. An example of this functionality is displayed in Figs. 5 and 6, where a comparison between the 3D reconstructed geometry and the real view of the center of Madrid can be observed. The main benefit of the developed method is that any area of the world can be recreated in 3D almost instantaneously. Therefore, any part of the world can be analyzed by applying the ray launching approach without the need of having the geometrical model of the scenario under analysis; the simulation tool automatically generates it if there are OSM data available for that area. After this, the simulation parameters and the reconstructed 3D urban environment are sent to the ray launching server. The server processes the data and runs the simulation. After that, it returns the results to the client interactively (25%, 50%, etc.) until the results are received completely. Since this process may take several seconds or even a minute, the user experience is improved by receiving the results in this way. When the simulation is completed, the user receives a notification. If the notification is clicked, the 3D environment will be shown displaying the results (path loss or received power). The 3D visualizer shows the rays launched from the emitter/s that have reached the receiver/s.

D. DATA AND REQUESTS HANDLING

For the application to achieve the defined accomplishments, there must be communication between the web application and the different servers that provide data.

FIGURE 5. 3D view obtained from google earth.

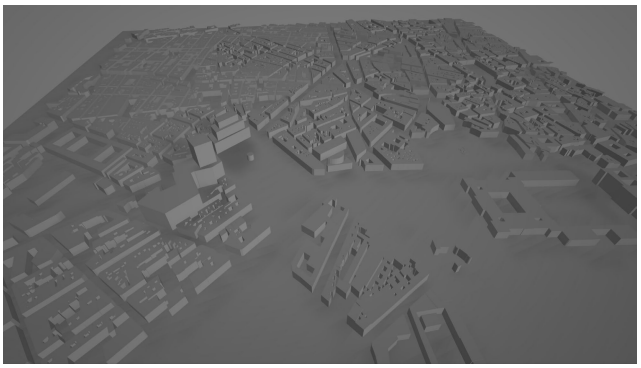

FIGURE 6. 3D reconstructed geometry.

This communication interface uses REST requests to gather data from the different servers in a single response and WebSockets [51] to request the data in fragments as they are being calculated. In the previous version, all communications were executed over REST APIs and HTTP. This limits communication in cases where partial information is to be sent. The WebSockets protocol allows sending partial information to the client. This provides more flexibility in the communication between the back-end and the front-end.

To manage REST requests, a module has been developed in which these requests are processed through different queues. This allows having a priority queue for requests, as well as managing attempts and failed requests, adding error resilience. This sub-module has been implemented in all Docker servers. Thus, request queues are used to allow retries in case the requests fail. These retries are configurable, so that the time interval between retries, the maximum number of retries, etc. can be defined. Each possible action on the map has a defined request associated with it. For example, drawing an area to display geospatial data is associated with a request for geospatial data for that area to the corresponding server. Encapsulating request handling and retries in a single module that can work with both REST and WebSocket requests allows implementing this functionality once, and reusing it multiple times along the application both in the front-end and in the back-end servers.

To manage the storage efficiently, a DataStore module has been developed. This module acts as a database in the browser. The development and implementation of the DataStore provide three main advantages:

- Unlike conventional browser storage systems, such as *LocalStorage*, the DataStore storage is not limited. The limit becomes the available local memory of the device from which the application is accessed.
- The DataStore has a notification system. When a request is made, a result object is registered in the DataStore. This object supports subscriptions that execute a call-back function when its state changes. Therefore, when the results arrive from the server, the state of the object changes from waiting to completed and the results are written in the DataStore, so the client can read them. Then, those classes subscribed and waiting for results are notified and execute the associated callback.
- The subscription system of the DataStore allows subscribing to groups of objects that match certain criteria as well as to a single object. This is ideal to maintain updated lists of objects of the same type in the user interface, such as a record of the simulation results or a record of the optimizations that have been carried out.

Using the DataStore, the lifecycle of animations and drawings on the map is managed efficiently. In the case of a request to the server of geospatial data to display the terrain heights, the following sequence of actions would be followed:

1) The area is drawn on the map.
2) A "waiting for data" animation is executed.
3) The associated heights object is added to the DataStore.
4) The area drawn on the map is subscribed to the height object in the DataStore.
5) The request is made to the server by the controller linked to the area.
6) When the data arrives, the DataStore object is updated, and the subscribed objects are notified.
7) The area waiting to be drawn will be updated when the data are available, so it reads the data and draws the image representing the height information on the map.

E. USE OF GeoJSON

To represent the areas that are displayed on the 2D interactive map, the GeoJSON specification is used. This same data representation is also used to store the geometries in a file to save the current state of the application and to communicate with the back-end servers. The GeoJSON data format allows for representing 2D vector geometries such as lines, rectangles, circles, polylines, and polygons. Thus, another benefit of using this format is that it allows indicating concrete areas with irregular shapes on the map that need to be analyzed, as shown in Fig. 7.

To include the results of the calculations, the properties section of the GeoJSON objects is used. This section is extended with a data object that can contain a matrix of data. That matrix represents the raster information alongside some metadata that describes the resolution of the raster, and the type of information that it contains, such as heights, path loss, population density, or terrain type. The properties' field is

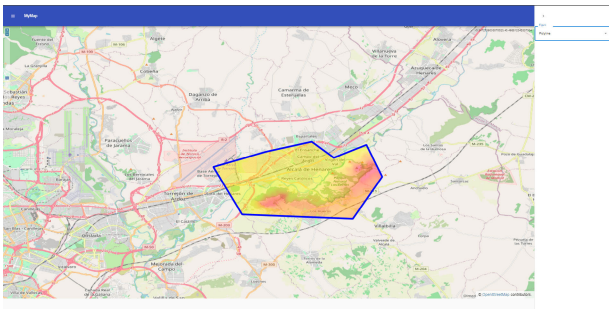

FIGURE 7. Example of an irregular polygon represented on the map to obtain results on a particular area.

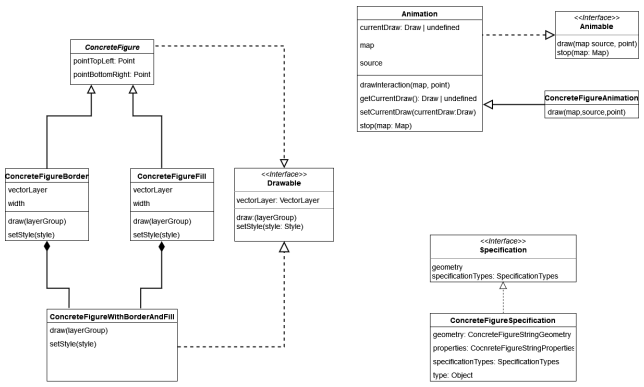

FIGURE 8. The specification objects are used to describe any 2D vector geometry.

also used to represent circular areas, which are expressed as a Point type geometry with a radius field on the properties object.

On the other hand, specification objects are used to create and describe the 2D vector geometries. With the same specification, multiple objects can be created, such as animations, areas to be drawn on the 2D map, or requests to the server's data for that specific area in that specific shape. The data model of the specification objects is shown in Fig. 8.

F. CREATING NEW AREAS

In the current version, the workflow to create new areas has been slightly modified. This has been possible due to the use of OpenLayers, the introduction of the DataStore, and the use of a single specification format for the 2D vector geometries.

To create a new area, which for the interactive 2D map is presented as a Drawable object, a builder is used. This builder can be controlled from direct interaction on the map or from a form that is being filled. While the Drawable is under construction, an animation will exist representing the current state of the partially created area. The builder will store the information of the new area in a specification, which when completed will be translated into a Drawable object by making use of a factory, as can be seen in Fig. 9.

Once an area has been created, it is ready to be included in the DataStore. The area will be linked to a controller that will subscribe to changes over it and to one or more Drawables that represent the area on a layer of the interactive

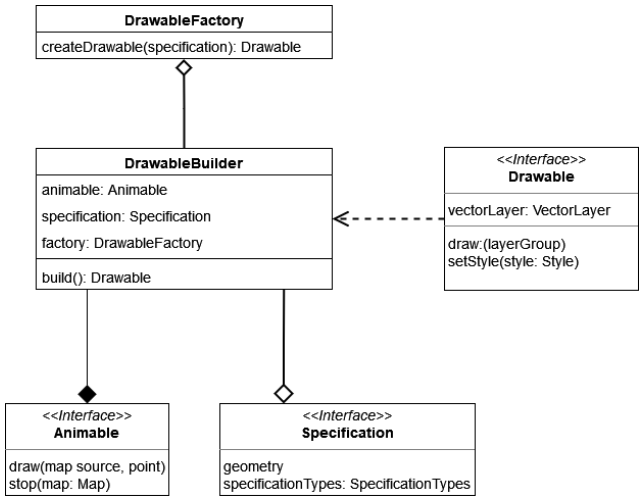

FIGURE 9. Diagram concerning the creating a new drawable object.

2D map. Depending on the state of the area, the controller will make sure that the missing raster data that the area requires are requested and that the Drawables are displaying the most updated version of the data. Other components of the application can also make use of the DataStore to receive updated information on the areas. The most common use cases are displaying the current state of an area on the interface and displaying a list of the areas that currently exist on the application. The subscribed components will be notified with updated information once an area changes or when a new area is created.

This new architecture allows components to be upgraded in a clean, integrated, and scalable way. Instead of having to employ polling techniques or use dependencies that add load on the client side, the DataStore mechanism provides a resource-efficient, easy-to-scale, and easy-to-add new functionality for updating areas and components.

G. RAY LAUNCHING VISUALIZATION MODULE

As mentioned before, the application allows the calculation of the radio propagation in microcellular environments using a fast ray launching algorithm.

The ray launching server receives a 3D environment in an .OBJ format. It also receives the coordinates of the transmitter(s) and the receiver(s), along with other simulation parameters. The most important ones are the frequency, the maximum number of ray segments to consider, the polarization, and the electromagnetic properties of the materials that compose the buildings.

Each transmitter is implemented as a point source of rays, while the receiver has to be a solid object or at least a plane to allow the ray-launching algorithm to find an intersection with it. Two options for the receiver are provided:

1) A sphere or various spheres allow the calculation of the field at a narrow location, approximating the point-like receiver with the small radius of the sphere (see Fig. 10).

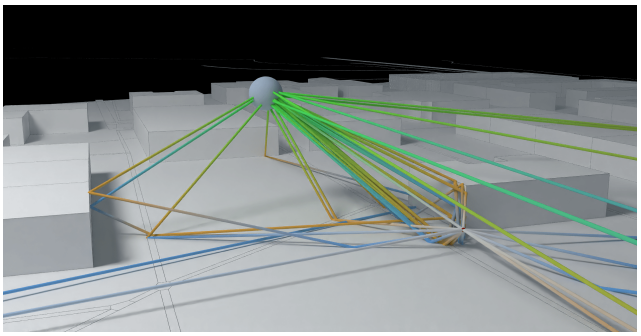

FIGURE 10. Using a sphere as a target.

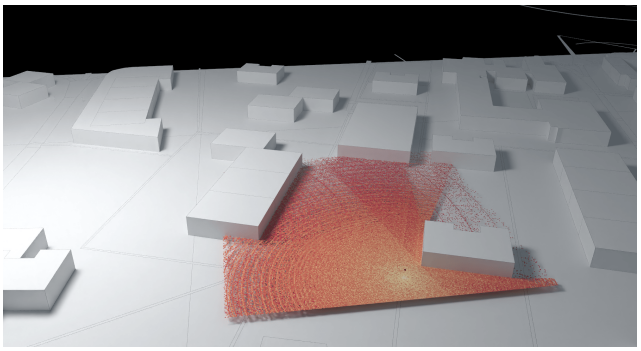

FIGURE 11. Using a plane as a target.

2) A plane allowing for the collection of large statistics of rays over the selected area. This option is computationally more efficient due to the higher probability of hitting the receiver. The receiver is transparent and can be hit multiple times in a single path. The received power along the measurement plane can be displayed as shown in Fig. 11. This capability provides direct observation of coverage in a typical urban environment and takes into account the 3-D nature of buildings and other structures in the propagation environment.

To implement the ray launching algorithm, the server makes use of the PlotOptix library [8]. This library, written in Python, is an interface that allows running Nvidia Optix [7] code. Nvidia Optix is a framework that encapsulates a ray tracing engine, making use of the GPUs to perform and accelerate ray tracing in a scalable, efficient, and faster way than other alternatives [52]. PlotOptix allows the combination of components prototyped in a high-level language with the optimized workflow of the underlying OptiX ray tracing engine. Nvidia OptiX handles compute-extensive tasks such as traversal of the scene and provides the GPU-oriented infrastructure for launching user-defined rays or executing custom code to determine the ray-object intersection and calculations when the intersection is found. PlotOptiX API allows for interaction with the OptiX engine from the level of the Python code. It handles initialization, provides functionality to define custom blocks of ray tracing code, and collects data required for further processing of the field coverage [57].

PlotOptix also includes a visualization tool to display the 3D environment that has been received as input and to show the rays that have been used by the algorithm to calculate the path loss or the received power at the receiver locations. However, this visualization tool cannot be integrated into the developed web tool as it is, since it is a desktop application. The most important disadvantage is that the interaction with the user is limited: it does not support either moving the camera or zooming in or out.

Because of these limitations, a tailored web interface has been implemented to allow the visualization of the simulation results and to enable users to interact with them. Once the results are calculated on the ray launching server, they are returned to the web application. The web application displays the simulation results on the 3D model of the environment. The main benefit of this new module is that it allows direct interaction with the results. The rays are colored depending on the type of segment: reflected, diffracted, transmitted, or direct. Users can select individual rays to obtain detailed information about them. The type of ray, path length, propagation time, electric field, direction of arrival, and other properties are displayed on a panel. Users can also filter rays according to several criteria, such as the type of ray or the number of bounces. Another advantage of this developed visualization module is that users can visualize the results in the 3D environment in a user-friendly way. Dragging and dropping are supported to move the camera, and zooming in/out with the wheel mouse, among other integrations. All these new functionalities, and specifically the possibility of hiding certain rays, help to better understand the results. This is especially helpful when visualizing a simulation that may look messy if all the rays are shown at the same time.

The ray launching visualization module has been designed so that graphical representations can be easily created and displayed on the web browser using JavaScript. Internally, it uses WebGL [53], [54], which is a graphics API that is used in web applications and is based on the OpenGL integrated graphics language, as its rendering engine.

To represent the rays, cylindrical geometries have been used instead of 3D lines. The rendering engine has limitations when it comes to adding additional features to the 3D line, mainly adding thickness to the line. It is desirable that the user of the application can interact with the rendered environment. A line thickness of 1 pixel complicates that interaction. The mechanism by which the user interacts with the geometries is called ray casting, and it works as follows:

1) When the user clicks on some point in the rendered space, from the camera coordinates where the user is located, multiple rays are cast to that point.
2) Any geometry that is intersected with the launched rays is received in an array.
3) The closest geometry to the camera that has been intersected is extracted.

Due to the way ray casting works, using lines of 1-pixel thickness, a millimeter precision is required from the

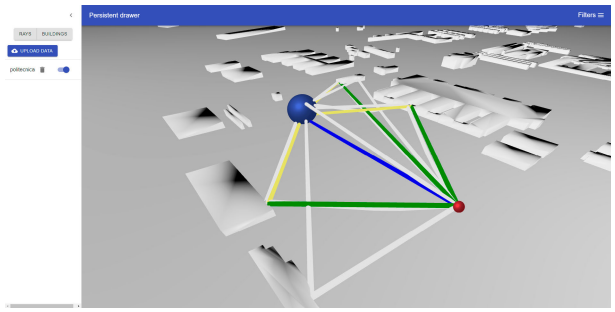

FIGURE 12. Ray launching visualization module.

user. To solve this problem, cylindrical geometries have been used to represent each of the ray segments. These cylindrical geometries have a configurable width, by default, small enough to appear to be a line but thick enough to provide adequate interaction with the user. The resulting web interface can be seen in Fig. 12.

Finally, the interactions with the 3D visualization module are performed using a simple query to select the filtered rays. Then, the tool displays only the selected rays and hides the rest. The filtering system is relatively simple and is based on set logic. An example query to show the rays that suffer three bounces coming from the second antenna and the diffracted rays coming from the third antenna would be the following: *antenna = 2 and bounces = 3 or antenna = 3 and rays = diffracted*.

The library that has been used to implement this part of the platform is ThreeJS [55]. ThreeJS is a JavaScript library to create and display static and animated 3D computer graphics in a web browser using WebGL. Some of its features are the implementation of cameras, lights, materials, shaders, objects, common geometries and import and export utilities.

Another benefit of using ThreeJS is that it allows loading and displaying, among other file formats, Wavefront (.obj) files [56]. OBJ Wavefront files are text files that contain mesh geometry data. This feature is particularly relevant for the developed tool since the ray launching server receives the 3D environment that is going to be analyzed in an .OBJ file format.

H. REQUIRED SIMULATION TIME

The simulation time required for coverage prediction mainly depends on the type of algorithm applied. When empirical/semi-empirical algorithms are applied, the results are obtained instantaneously. However, when a ray-launching simulation is run, the results are obtained after a few seconds or minutes. In this case, the required time mainly depends on the size of the area under analysis, the number of buildings within the area, and the maximum number of segments per ray. Regarding the optimizations, the required time also depends on the population size and the number of generations.

Table 2 includes a comparison of the required time to perform optimizations in the four areas depicted in Fig. 13.

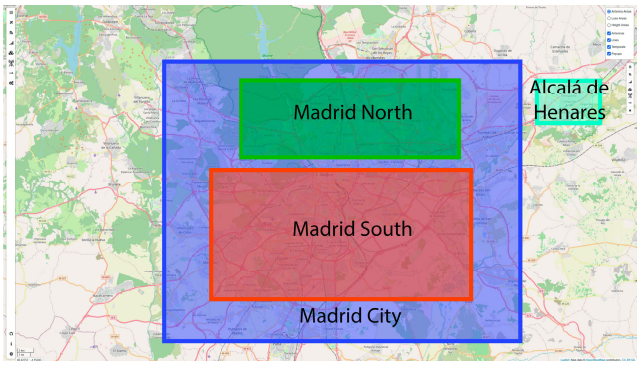

FIGURE 13. Madrid areas used for measurement.

TABLE 2. Comparison of the required time in seconds to perform a optimization in the areas depicted in Fig. 13.

	Alcalá de Henares	Madrid North	Madrid South	Madrid City
Simple config	Inmediate	Inmediate	1	6
Medium config	Inmediate	Inmediate	3	13
Complex config	Inmediate	1	6	26

Alcalá de Henares area covers 61.19 km^2, Madrid North covers 210.00 km^2, Madrid South covers 950 km^2, and the entire Madrid City area spans 1215 km^2.

As it can be seen in Table 2, three different configurations were evaluated: a simple configuration with a population size of 10 individuals and 10 generations, an intermediate configuration with a population size of 50 individuals and 50 generations and a complex configuration with a population size of 100 individuals and 100 generations. The results are immediately obtained for small and medium areas. A few seconds are needed to obtain the results in the biggest area.

On the other hand, ray launching simulations were performed in the city of Cartagena, Spain. A 500m x 500m area was considered and two different point to point simulations were conducted. In the first one, a total of 2,295 successful paths were obtained between transmitter and receiver setting 5 maximum ray bounces (5 segments per ray). The consumed time was 10 seconds. In the second one, a total of 27,085 successful paths were obtained setting 16 maximum ray bounces (16 segments per ray). The consumed time in this case was 56 seconds.

IV. CONCLUSION AND FUTURE WORK

In this paper, the improvement of an advanced radio propagation simulation platform based on ray launching and empirical/semi-empirical methods to provide a better user experience is presented.

First, the different options considered for its implementation and improvements have been reviewed. From this analysis, it was concluded that the next step for the development of the new tool should be to start making use of the library OpenLayers, together with the use of deterministic methods in order to provide more accurate results.

In addition, new design solutions have been developed, improving the functionality and architecture of the application. Some of them are the inclusion of elements that allow the handling of data and requests, like the DataStore. The DataStore allows storing and managing multiple types of data through a seamless subscription-based interface. The DataStore is flexible enough to handle the current data types but can be extended to support any datatype that may become necessary during the continuation of the development. Also, a noteworthy feature is the request system developed. The presented request system integrates well with the DataStore, allowing it to manage errors on the requests through a reasonable interface.

Other new design solutions that were introduced in the second version have been the use of specifications to create, store, and transmit spatial information using the GeoJSON format. These specifications can directly be translated into specific elements of the data model, like the areas or the Drawables.

Finally, the solution adopted to represent ray launching results on the map using the library ThreeJS has been shown. As a result, an innovative high-performance radio propagation simulation platform has been implemented to provide fast and accurate results in a user-friendly way.

The next steps for the application will be to keep improving the 3D module to be closely integrated with the interactive 2D map so that both can work seamlessly in an integrated way. Additionally, the models of the buildings that are rendered on the 3D view could be enhanced to be displayed over the actual terrain that it is underneath them instead of over a flat surface. Mesh material data could be added too, providing information about the surface of the surroundings, like glass or road materials. This would provide more precision in the simulation and improve the visualization of the results. This could also greatly improve the results of the ray launching algorithm used to calculate the path loss when the selected area is large enough. Moreover, the application needs to keep improving its optimization algorithms for antenna placement, which could make use of the ray-launching simulation results as input. At the moment, they only use empirical/semi-empirical path loss calculations from the first version of the tool.

REFERENCES

[1] L. Lozano, M. J. Algar, I. Gonzalez, and F. Catedra, "FASANT: A versatile tool to analyze antennas and propagation in complex environments," in *Proc. 3rd Eur. Conf. Antennas Propag.*, Berlin, Germany, 2009, pp. 2088–2092.

[2] F. S. de Adana, F. J. Fernández, J. L. Loranca, and R. Kronberger, "Covermap: Computer tool to calculate the propagation in open areas importing data from GoogleMaps," in *Proc. Loughborough Antennas Propag. Conf.*, Nov. 2009, pp. 229–232.

[3] D. Parsons, *The Mobile Radio Propagation Channel*. London, U.K.: Pentech Press Limited, 1992.

[4] S. Y. Tan and H. S. Tan, "Propagation model for microcellular communications applied to path loss measurements in Ottawa city streets," *IEEE Trans. Veh. Technol.*, vol. 44, no. 2, pp. 313–317, May 1995, doi: 10.1109/25.385924.

[5] A. Tayebi, J. Gomez, F. S. de Adana, O. Gutierrez, and M. F. de Sevilla, "Development of a web-based simulation tool to estimate the path loss in outdoor environments using OpenStreetMaps [wireless corner]," *IEEE Antennas Propag. Mag.*, vol. 61, no. 1, pp. 123–129, Feb. 2019, doi: 10.1109/MAP.2018.2883088.

[6] (May 2022). *OpenStreetMap*. [Online]. Available: https://www.openstreetmap.or

[7] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "OptiX: A general purpose ray tracing engine," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–13, Jul. 2010, doi: 10.1145/1778765.1778803.

[8] R. Sulej. *PlotOptiX: Ray Tracing and Data Visualization Package for Python*. Accessed: Jun. 3, 2023. [Online]. Available: https://plotoptix.rnd.team

[9] 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," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 10–27, 1st Quart., 2019, doi: 10.1109/COMST.2018.2865724.

[10] *Cellular Expert*. Accessed: Jun. 3, 2023. [Online]. Available: http://www.cellular-expert.com/

[11] Radio Science Lab UBC. (2022). *Directory of Wireless System Planning Tools*. [Online]. Available: http://rsl.ece.ubc.ca/planning.html#MentumPlanet

[12] A. Hrovat, A. Vilhar, I. Ozimek, T. Javornik, and E. Kocan, "GRASS-RaPlaT–radio planning tool for GRASS GIS system," in *Proc. ICECom*, Dubrovnik, Croatia, Oct. 2013, pp. 1–5.

[13] (May 2022). *Radio Mobile*. [Online]. Available: http://www.ve2dbe.com/english1.html

[14] F. S. De Adana, J. Gómez, A. Tayebi, and J. Casado, *Applications of Geographic Information Systems for Wireless Network Planning*. Norwood, MA, USA: Artech House, 2020.

[15] M. I. Nashiruddin, P. Aulia Fadhila, and M. A. Nugraha, "4G LTE-advanced network planning and simulation study in an urban region utilizing the 700 MHz frequency," in *Proc. IEEE 12th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Dec. 2021, pp. 0880–0886, doi: 10.1109/UEMCON53757.2021.9666529.

[16] S. Liu, "Simulation and analysis of civil aircraft cabin wireless network based on winprop," in *Proc. 4th Int. Conf. Inf. Technol. Elect. Eng.*, 2021, pp. 1–5, doi: 10.1145/3513142.3513225.

[17] Cloud-RFT. (2022). *Online RF Planning Software—Cloud-RFT*. [Online]. Available: https://cloudrf.com

[18] J. E. A. Peña, "Simulation of radiopropagation coverage in a fixed reception network of DVB-T2 digital terrestrial television: Metropolitan scenario of Bogotá D. C. (Colombia)," in *Proc. Congreso Internacional de Innovacion y Tendencias Ingenieria (CONIITI)*, Oct. 2017, pp. 1–6, doi: 10.1109/CONIITI.2017.8273349.

[19] J. Mendez-Rangel and C. Lozano-Garzon, "A network design methodology proposal for e-health in rural areas of developing countries," in *Proc. 6th Euro Amer. Conf. Telematics Inf. Syst.*, 2012, p. 339,345, doi: 10.1145/2261605.2261657.

[20] R. Fujdiak, P. Mlynek, J. Misurec, and M. Strajt, "Simulated coverage estimation of single gateway LoRaWAN network," in *Proc. 25th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jun. 2018, pp. 1–4, doi: 10.1109/IWSSIP.2018.8439232.

[21] M. Zennaro, M. Rainone, and E. Pietrosemoli, "Radio link planning made easy with a telegram bot," in *Int. Conf. Smart Objects Technol. Social Good (GOODTECHS)*, Venice, Italy, Dec. 2017, pp. 295–304, doi: 10.1007/978-3-319-61949-1_31.

[22] A. Omorinoye and Q.-T. Vien, "On the optimisation of practical wireless indoor and outdoor microcells subject to QoS constraints," *Appl. Sci.*, vol. 7, no. 9, p. 948, Sep. 2017, doi: 10.3390/app7090948.

[23] A. Tolegenova, A. Zhanys, S. Nurkasymova, and L. Soboleva, "Overview of 4G, 5G radio spectrum spectrum in the world and Kazakhstan," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 934, no. 1, p. 12055, 2020, doi: 10.1088/1757-899X/934/1/012055.

[24] *Probe*. Accessed: Jun. 3, 2023. [Online]. Available: https://www.v-soft.com/probe

[25] L. Gotszald, "Novel tracing algorithm vs remcom wireless InSite," *Int. J. Electron. Telecommun.*, vol. 61, no. 3, pp. 273–279, Sep. 2015.

[26] K. Qiu, S. Bakirtzis, H. Song, J. Zhang, and I. Wassell, "Pseudo ray-tracing: Deep leaning assisted outdoor mm-Wave path loss prediction," *IEEE Wireless Commun. Lett.*, vol. 11, no. 8, pp. 1699–1702, Aug. 2022, doi: 10.1109/LWC.2022.3175091.

[27] A. Hrovat, A. Vilhar, I. Ozimek, T. Javornik, and E. Kocan, "GRASS-RaPlaT—Radio planning tool for GRASS GIS system," in *Proc. ICECom*, 2013, pp. 1–5, doi: 10.1109/ICECom.2013.6684714.
[28] (2022). *CRFS—Spectrum Monitoring and Geolocation*. [Online]. Available: https://www.crfs.com/
[29] D. Jackson, "FirstNet turns to InfoVista for network-planning tool," Urgent Commun., Infovista Technol., Billerica, MA, USA, Tech. Rep., Feb. 2014.
[30] L. Korowajczuk, "Full apron visibility design," in *Proc. Integr. Commun. Navigat. Surveill. Conf. (ICNS)*, Sep. 2020, pp. 4G2-1–4G2-15, doi: 10.1109/ICNS50378.2020.9222922.
[31] EDX Wireless. (2022). *Best Radio/Wireless Planning Software—EDX Wireless*. [Online]. Available: https://edx.com
[32] D. Applegate, A. Archer, D. S. Johnson, E. Nikolova, M. Thorup, and G. Yang, "Wireless coverage prediction via parametric shortest paths," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 1–12, doi: 10.1145/3209582.3209605.
[33] M. Malkawi, K. Al-Zoubi, and A. Shatnawi, "Quasi real-time intermodulation interference method: Analysis and performance," *Int. J. Commun. Netw. Inf. Secur.*, vol. 13, no. 1, pp. 125–135, Apr. 2022.
[34] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRA low-power wide-area networks scale?" in *Proc. 19th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Nov. 2016, pp. 59–67, doi: 10.1145/2988287.2989163.
[35] Comsearch. (2022). *iQ.Link® Microwave Design Software*. [Online]. Available: https://www.comsearch.com/products/planning-tools/iqlink
[36] J. Llerena, A. Méndez, and F. Sánchez, "Analysis of the factors that condition the implementation of a backhaul transport network in a wireless ISP in an unlicensed 5 GHz band, in the Los Tubos sector of the Durán Canton," in *Proc. Int. Conf. Inf. Syst. Comput. Sci. (INCISCOS)*, Nov. 2019, pp. 15–22, doi: 10.1109/INCISCOS49368.2019.00012.
[37] J. Casado, J. González, A. Tayebi, J. Gómez, and F. S. de Adana, "Application of bioinspired algorithms for the optimization of a radio propagation system simulator based on OpenStreetMap," in *Proc. 4th Int. Conf. Adv. Comput., Commun. Services*, Nice, France, 2019, pp. 8–11.
[38] T. F. Eibert and P. Kuhlmann, "Notes on semiempirical terrestrial wave propagation modeling for macrocellular environments–comparisons with measurements," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2252–2259, Sep. 2003.
[39] J. Gómez, A. Tayebi, and J. Casado, "Extraction and use of geometry data to obtain 3D buildings on a web map," in *Proc. ACCSE 5th Int. Conf. Adv. Comput., Commun. Services*, Lisbon, Portugal, 2020, pp. 1–5.
[40] M. Barranquero, J. Casado, J. Gómez, A. Tayebi, and J. A. Jiménez, "Docker security in web simulation tools: A layered approach," in *Proc. Eurasiaweb Int. Conf., Marrakesh (Morocco)*, Aug. 2021, pp. 11–14.
[41] W. Benger, D. Hildenbrand, and W. Dobler, "Optimizing refined geometric primitive's leaflet visibility for interactive 3D visualization via geometric algebra," in *Proc. Comput. Graph. Int.*, Jun. 2018, pp. 267–272, doi: 10.1145/3208159.3208194.
[42] O. Zavala-Romero, E. P. Chassignet, J. Zavala-Hidalgo, P. Velissariou, H. Pandav, and A. Meyer-Baese, "OWGIS 2.0: Open source Java application that builds web GIS interfaces for desktop and mobile devices," in *Proc. 22nd ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2014, pp. 311–320, doi: 10.1145/2666310.2666381.
[43] J. J. Durillo, A. J. Nebro, and E. Alba, "The jMetal framework for multi-objective optimization: Design and architecture," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8, doi: 10.1109/CEC.2010.5586354.
[44] J. J. Durillo and A. J. Nebro, "JMetal: A Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965997811001219
[45] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal multi-objective optimization framework," in *Proc. Companion Publication Annu. Conf. Genetic Evol. Comput.*, Jul. 2015, pp. 1093–1100, doi: 10.1145/2739482.2768462.
[46] A. J. Nebro, M. López-Ibáñez, C. Barba-González, and J. García-Nieto, "Automatic configuration of NSGA-II with jMetal and irace," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2019, pp. 1374–1381, doi: 10.1145/3319619.3326832.
[47] A. J. Nebro, J. Pérez-Abad, J. F. Aldana-Martin, and J. García-Nieto, "Evolving a multi-objective optimization framework," in *Applied Optimization and Swarm Intelligence* (Springer Tracts in Nature-Inspired Computing), E. Osaba and X. S. Yang, Eds. Singapore: Springer, 2021, doi: 10.1007/978-981-16-0662-5_9.
[48] A. J. Nebro, E. Alba, G. Molina, F. Chicano, F. Luna, and J. J. Durillo, "Optimal antenna placement using a new multi-objective Chc algorithm," in *Proc. 9th Annu. Conf. Genetic Evol. Comput.*, Jul. 2007, pp. 876–883, doi: 10.1145/1276958.1277128.
[49] J. Wendel, A. Simons, A. Nichersu, and S. M. Murshed, "Rapid development of semantic 3D city models for urban energy analysis based on free and open data sources and software," in *Proc. 3rd ACM SIGSPATIAL Workshop Smart Cities Urban Analytics*, New York, NY, USA, Nov. 2017, pp. 1–7, doi: 10.1145/3152178.3152193.
[50] M. Barranquero, A. Olmedo, J. Gómez, A. Tayebi, C. J. Hellín, and F. S. de Adana, "Automatic 3D building reconstruction from OpenStreetMap and LiDAR using convolutional neural networks," *Sensors*, vol. 23, no. 5, p. 2444, Feb. 2023, doi: 10.3390/s23052444.
[51] J. Gómez, A. Tayebi, and J. Casado, "On the use of websockets to maintain temporal states in stateless applications," in *Proc. 15th Int. Conf. Internet Web Appl. Services (ICIW)*, 2020, pp. 21–24.
[52] F. Mauch, M. Gronle, W. Lyda, and W. Osten, "Open-source graphics processing unit–accelerated ray tracer for optical simulation," *Opt. Eng.*, vol. 52, no. 5, May 2013, Art. no. 053004.
[53] D. Limberger, M. Pursche, J. Klimke, and J. Döllner, "Progressive high-quality rendering for interactive information cartography using WebGL," in *Proc. 22nd Int. Conf. 3D Web Technol.*, Jun. 2017, pp. 1–4, doi: 10.1145/3055624.3075951.
[54] M. Krämer and R. Gutbell, "A case study on 3D geospatial applications in the web using state-of-the-art WebGL frameworks," in *Proc. 20th Int. Conf. 3D Web Technol.*, 2015, pp. 189–197, doi: 10.1145/2775292.2775303.
[55] B. Danchilla, "Three.Js framework," in *Beginning WebGL for HTML5*. Berkeley, CA, USA, Apress, 2012, doi: 10.1007/978-1-4302-3997-0_7.
[56] A. L. Possemiers and I. Lee, "Fast OBJ file importing and parsing in CUDA," *Comput. Vis. Media*, vol. 1, no. 3, pp. 229–238, Sep. 2015, doi: 10.1007/s41095-015-0021-5.
[57] J. Gómez, A. Tayebi, C. J. Hellín, A. Valledor, M. Barranquero, and J. J. Cuadrado-Gallego, "Accelerated ray launching method for efficient field coverage studies in wide urban areas," *Sensors*, vol. 23, no. 14, p. 6412, Jul. 2023, doi: 10.3390/s23146412.

JOSEFA GÓMEZ was born in 1984. She received the B.S. and M.S. degrees in telecommunications engineering from the Polytechnic University of Cartagena, Spain, in 2005 and 2007, respectively, and the Ph.D. degree in telecommunications engineering from the University of Alcalá, Spain, in 2011. She was a Faculty Researcher with The University of Hong Kong, in 2011, and Instituto de Telecomunicações, Lisbon, in 2014. Since 2012, she has been an Assistant Professor with the University of Alcalá. She has participated in 37 research projects with Spanish and European companies. She has published 29 articles in peer-reviewed journals, a book, two book chapters, and more than 45 conference contributions at national and international symposia. Her current research interests include optimization and analysis of antennas, design of graphical user interfaces, and the study of propagation for mobile communications or wireless networks in both outdoor and indoor environments.

CARLOS J. HELLÍN received the B.S. degree in computer science engineering and the M.S. degree in agile software development for the web from the University of Alcalá, Spain, in 2022 and 2023, respectively. He is currently pursuing the Ph.D. degree in information and knowledge engineering. He is a Researcher with the Department of Computer Science, University of Alcalá. His current research interests include the optimization of antennas, the analysis of radio propagation in rural and urban environments, gamification, data science, and artificial general intelligence.

ADRIÁN VALLEDOR received the B.S. degree in information systems engineering and the M.S. degrees in agile software development for the web from the University of Alcalá, Spain, in 2021 and 2023, respectively. He is currently pursuing the Ph.D. degree in information and knowledge engineering. He is a Researcher with the Department of Computer Science, University of Alcalá. His current research interests include the optimization of antennas, the analysis of radio propagation in rural and urban environments, artificial intelligence, neuronal networks, and voice processing.

MARCOS BARRANQUERO received the bachelor's degree in computer engineering and the master's degree in cybersecurity from the University of Alcalá, in 2020 and 2021, respectively. He is currently pursuing the Ph.D. degree in computer engineering, exploring new technologies in the field of web development and ray tracing simulation. In addition to his technical knowledge, he is passionate about teaching and mentoring. His current research interests include full-stack and web programming, backend development, cybersecurity, and 3D geometry construction.

JUAN J. CUADRADO-GALLEGO is currently an Associate Professor of computer science and artificial intelligence with the Department of Computer Science, University of Alcalá, and an Affiliate Associate Professor with the Department of Computer Science and Software Engineering, Faculty of Engineering and Computer Science, Concordia University, Montreal, Canada. He has made more than 200 scientific publications, many of which have been in journals indexed in the JRC Science Edition. He has participated as a principal investigator or a researcher of numerous research projects. He has also been an External Evaluator of projects in computer science by the Natural Sciences and Engineering Research Council of Canada, since 2014, and an Evaluator of the National Agency for Evaluation and Prospective, the General Directorate of Scientific and Technical Research of Spain, and the Ministry of Economy, Industry and Competitiveness, Spain. His current research interests include artificial intelligence and data science.

ABDELHAMID TAYEBI was born in 1983. He received the B.S. and M.S. degrees in telecommunications engineering from the University Polytechnic of Cartagena, Spain, in 2005 and 2007, respectively, and the Ph.D. degree in telecommunications engineering from the University of Alcalá, Spain, in 2011. In 2011, he was an Assistant Professor with the University of Alcalá, where he has been a Professor, since 2019. He was a Faculty Researcher with The University of Hong Kong, in 2011, and Instituto de Telecomunicações, Lisbon, in 2014. He has participated in 35 research projects with Spanish and European companies. He has published 27 articles in peer-reviewed journals, a book, a book chapter, and around 45 conference contributions at national and international symposia. His current research interests include the design and optimization of antennas, electromagnetic radiation and scattering, and the development of web tools for the analysis of radio propagation in rural and urban environments.