

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Universidad
Politécnica
de Cartagena**

Trabajo Fin de Grado

**Utilización de OpenStreetMap en el cálculo de coberturas mediante un
trazador de rayos**



Autor: Adrián Gilabert Pérez
Director: Juan Pascual García

Agradecimientos

A mis padres y a mi pareja Patricia, por darme su apoyo durante estos años y estar ahí siempre que los he necesitado.

A mi director de proyecto, D.Juan Pascual García, por ayudarme a través de numerosas tutorías, haciendo que este proyecto sea posible.

A mis compañeros de carrera por los buenos momentos que hemos vivido juntos y hacer que estos años sean inolvidables.

Muchas gracias.

Índice general

1. Introducción	3
1.1. Motivación	3
1.2. Objetivos del proyecto	4
2. Modelos de propagación en entornos urbanos	7
2.1. Características y clasificación de los modelos de propagación	7
2.2. Clasificación de modelos de propagación en entornos urbanos	8
2.3. Definición del trazado de rayos	10
2.4. Trazado de rayos	12
3. Aplicación a partir de los mapas de OSM	17
3.1. El proyecto OSM	17
3.2. Elementos que se encuentran representados en OSM	18
3.3. Importación de los mapas OSM en MATLAB	22
3.3.1. Extracción de los datos a partir del archivo OSM	22
3.3.2. Representación del entorno original	23
3.4. Adaptación de los mapas OSM al trazador de rayos	25
3.4.1. Introducción a la herramienta de trazado de rayos	25
3.4.2. Representación del entorno en la herramienta de trazado de rayos	27
3.4.3. Selección de elementos de interés en el entorno urbano	33
3.4.4. Conversión de coordenadas geográficas a coordenadas UTM	35

3.4.5. Implementación de las superficies verticales	41
3.4.6. Implementación de las superficies horizontales	43
3.4.7. Propiedades eléctricas de los materiales que forman las superficies	47
4. Resultados obtenidos a partir de medidas reales	53
4.1. Presentación del entorno a simular	53
4.2. Aplicación de la herramienta de trazado de rayos	55
4.3. Comparación de los resultados obtenidos con las medidas reales	61
5. Conclusiones y futuras líneas de desarrollo	73

Índice de figuras

2.1. Tipos de rayos en un trazador de rayos[3]	11
2.2. Diferentes radios de impacto en el receptor[3]	12
2.3. Método de las imágenes utilizado para encontrar la diferencia de caminos entre el rayo directo y el rayo reflejado[1]	14
2.4. Imágenes de primer y segundo orden producidas por una fuente (S) situada entre dos espejos (A y B) [5]	15
3.1. Cabeceras XML en un nodo	19
3.2. Ejemplos de nodos en un mapa OSM. Se muestran los nos que definen el Cuartel de Antigones	20
3.3. Cabeceras XML en una vía	20
3.4. Ejemplos de vías en un mapa OSM. La vía mostrada es el Paseo de Alfonso XIII en Cartagena	21
3.5. Estructura del archivo OSM en MATLAB[10]	23
3.6. Ejemplo de representación haciendo uso de la función <i>plot_way</i>	24
3.7. Formato de las paredes rectangulares de la estructura[11]	29
3.8. Ejemplo de pared rectangular definida en MATLAB[11]	30
3.9. Efecto de cruce de rectas en un error de definición de paredes[11]	31
3.10. Formato de las paredes rectangulares de la estructura[11]	32
3.11. Ejemplo de pared triangular definida en MATLAB	32
3.12. Representación de la pared triangular definida en MATLAB	33

3.13. Representación de los conjuntos de nodos con la etiqueta <i>building</i> de la zona de Cartagena mostrada en la Figura 3.6	34
3.14. Sistemas de coordenadas geográficas	36
3.15. Sistemas de coordenadas UTM	37
3.16. Representación del entono en metros	40
3.17. Representación de la parte inferior y superior de los edificios del entorno	42
3.18. Representación del entono 2D a la altura deseada	43
3.19. Intersecciones creadas a partir de las proyecciones de rectas	45
3.20. Superficies horizontales a partir de triangulación	47
3.21. Comparación de entornos	48
4.1. Coordenadas de la estación base en la Calle Juan Fernández	53
4.2. Coordenadas de la estación base en la Calle Juan Fernández	54
4.3. Elementos del entorno urbano a simular. El circulo verde indica la estación base (transmisor)	55
4.4. Entorno urbano con las posiciones del receptor[17]	56
4.5. Entorno urbano con el recorrido donde se tomaron las diferentes medidas	57
4.6. Potencia recibida en la simulación en función de la distancia	61
4.7. Comparación de las medidas reales con las medidas ofrecidas por la simulación	62
4.8. Comparación de la potencia recibida (dBm) en cada punto del recorrido de la zona cercana frente a la distancia al transmisor(dB)	63
4.9. Diferencia entre la potencia recibida en cada punto y el valor de la recta de regresión en la zona cercana	64
4.10. Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las medidas reales	66

4.11. Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las soluciones otorgadas por la simulación 67

4.12. Comparación de la potencia recibida (dBm) en cada punto del recorrido de la zona lejana frente a la distancia al transmisor(dB) 68

4.13. Diferencia entre la potencia recibida en cada punto y el valor de la recta de regresión en la zona lejana 69

4.14. Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las medidas reales 70

4.15. Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las soluciones otorgadas por la simulación 71

Índice de tablas

3.1. Propiedades eléctricas de los materiales [16]	50
3.2. Atenuaciones de diferentes materiales [4]	51
4.1. Parámetros de entrada de la función <i>genera_rayos_paralelo</i>	59
4.2. Parámetros de entrada de la función <i>genera_rayos_difracción</i>	60
4.3. Tiempo aproximado de computo por posición de recepción en minutos para diferentes casos	61
4.4. Valores de L_0 y n obtenidos	63
4.5. Varianza y desviación típica de la función de error en ambos casos . . .	66
4.6. Valores de L_0 y n obtenidos	67
4.7. Varianza y desviación típica de la función de error en ambos casos . . .	69

Capítulo 1

Introducción

1.1. Motivación

En la actualidad las comunicaciones móviles viven un gran momento que se mantiene en constante evolución, debido principalmente a la creciente penetración de la telefonía móvil y la consolidación y transformación de la banda ancha móvil, sumada a la cada vez mayor incorporación al mercado de nuevos equipos. Esta evolución es debida a las ventajas que nos proporcionan dicha tecnología frente a la red cableada, dando lugar a gran cantidad de nuevas aplicaciones y mercados en el sector de las telecomunicaciones.

La existencia de una herramienta capaz de llevar a cabo cálculos de coberturas para sistemas de comunicaciones de forma precisa se presenta indispensable para facilitar el análisis y resolución de problemas de radiación de campos electromagnéticos en todo tipo de entornos. Una de las herramientas que cuenta con mayor precisión para el cálculo de dichas coberturas es el trazador de rayos. Esta herramienta puede llevar a cabo un cálculo de la cobertura radioeléctrica en un entorno con una precisión muy elevada si disponemos de un modelo digital del terreno que represente fielmente el entorno real.

Para lograr este modelo digital se hará uso de OpenStreetMap (OSM), un proyecto colaborativo para crear mapas libres y editables. Los mapas generados por los volun-

tarios de OSM han alcanzado gran precisión por lo que son idóneos para implementar en el trazador de rayos. Es este trabajo se llevará a cabo una adaptación de los mapas OSM con el objetivo de utilizar estos mapas con un trazador de rayos programado en MATLAB.

1.2. Objetivos del proyecto

El objetivo principal de este Trabajo Final de Grado es el de desarrollar una aplicación que permita la utilización de cualquier tipo de entorno 2D exportado de OpenStreetMap, llevando a cabo un modelo simplificado de dichos mapas y estableciendo un modelo en 3D aproximado con la altura media de los elementos de la zona que permita su compatibilidad con el trazador de rayos previamente programado en MATLAB por el grupo de investigación de Sistemas de Comunicaciones Móviles de la UPCT. En los siguientes puntos se exponen las tareas a realizar:

- Obtención de modelos simplificados de entornos urbanos a partir de mapas de OpenStreetMap.
- Aplicación de los modelos simplificados en el trazador de rayos disponible en el grupo de investigación.
- Comparación de las coberturas obtenidas con el trazador de rayos con medidas experimentales y con otras simulaciones del canal radio.

La memoria de este Trabajo Final de Carrera se encontrará dividida en cinco capítulos donde se describirá la teoría en la que se basa el trazador de rayos, la posibilidad de adaptar los mapas de OpenStreetMap para obtener modelos que pueden ser utilizados por dicha herramienta, una simulación para comparar los resultados obtenidos a partir de estos modelos y las conclusiones que podemos deducir a partir de estos resultados.

El primer capítulo es en el que nos encontramos. Su utilidad es presentar el proyecto con sus objetivos que se pretendían lograr al comienzo del mismo.

El objetivo del segundo capítulo es exponer la base teórica sobre la que reposan los diferentes modelos de propagación más utilizados en entornos urbanos. Se describirán los modelos más importantes tanto para entornos urbanos macrocelulares como para entornos urbanos microcelulares profundizando en mayor medida en el modelo del trazado de rayos.

El tercer capítulo describe como se gestionan los datos desde la plataforma de OpenStreetMap y como a partir de dichos datos se puede llevar a cabo el modelado 3D de entornos urbanos, seleccionando únicamente aquellos elementos del terreno más influyentes en el cálculo de coberturas. Posteriormente se establecerán los pasos seguidos para la elaboración de la aplicación.

El cuarto capítulo de esta memoria mostrará los resultados obtenidos con la aplicación desarrollada al introducir los modelos 3D obtenidos en el trazador de rayos desarrollado en MATLAB por el grupo de investigación. Los resultados de coberturas obtenidos por el trazador de rayos se compararán con medidas experimentales u otras simulaciones realizadas.

Finalmente, el quinto capítulo contiene las conclusiones obtenidas al comparar los resultados obtenidos con la herramienta y las medidas reales tomadas en dicho entorno. Además se propondrán unas futuras líneas de desarrollo para mejorar la aplicación.

Capítulo 2

Modelos de propagación en entornos urbanos

2.1. Características y clasificación de los modelos de propagación

En cualquier transmisión, la señal que es emitida por el emisor sufrirá diferentes transformaciones debido al camino que recorre hasta llegar al receptor, es decir, al receptor le llegará una versión transformada de la señal original. Además, al existir diferentes elementos en el entorno, al receptor le llegarán diversas réplicas de la señal transmitida. A este efecto se le llama efecto multicamino.

Los modelos de propagación surgen debido a la dificultad de predecir estas alteraciones que sufrirá la señal transmitida durante su propagación. Los modelos de propagación se centran en predecir la potencia de señal recibida en el receptor y el perfil de potencia (Power Delay Profile, PDP) haciendo uso de un conjunto de expresiones matemáticas, diagramas y algoritmos que son capaces de representar las características de un canal radio.

Es posible clasificar estos modelos en cuatro tipos, según el método empleado en la obtención de las pérdidas de propagación:

- **Modelos empíricos.** Este tipo de modelos se basan en campañas de medidas. Antiguamente estas medidas quedaban recogidas en tablas o curvas que relaciona-

ban las pérdidas con los parámetros libres del modelo, es decir, las características descriptivas propias del entorno.

- **Modelos semi-empíricos.** Se tratan de modelos empíricos a los que se les ha añadido soluciones exactas de alguno de los modelos de propagación.
- **Modelos deterministas.** Estos modelos se basan en la resolución de las ecuaciones de los mecanismos de propagación que cuantifican las pérdidas: reflexión, difusión y difracción básicamente. A las frecuencias de trabajo de comunicaciones móviles no existe ningún modelo puramente determinista, debido a que los elementos que encontramos en el canal no se pueden representar con el detalle requerido, que es del orden de la longitud de onda λ .
- **Modelos semi-deterministas.** Este tipo de modelos superan la restricción que poseían los modelos deterministas. De esta forma, es posible utilizar técnicas estadísticas para modelar algunos de los elementos físicos propios del canal en el que se produce la propagación.

2.2. Clasificación de modelos de propagación en entornos urbanos

El modelado del entorno urbano puede llegar a ser complejo debido a la cantidad de elementos y obstáculos que pueden existir entre el emisor y el receptor. Si nos encontramos en un entorno macrocelular, donde el número de edificios puede ser muy elevado, la señal transmitida sufrirá diversos efectos hasta su llegada al receptor. El efecto multicamino en este tipo de entornos cobra gran importancia ya que la cantidad de rayos difractados y refractados se verá incrementada considerablemente.

Existen algunos modelos empíricos que estiman las pérdidas básicas de propagación y la intensidad de campo recibida en este tipo de entornos. Uno de estos modelos es

2.2. CLASIFICACIÓN DE MODELOS DE PROPAGACIÓN EN ENTORNOS URBANOS 9

el modelo de Okumura-Hata, que únicamente tiene en cuenta parcialmente el grado de urbanización del entorno y no el entorno real del receptor. También existen modelos semi-empíricos como el modelo de Walfisch-Bertoni y el COST-231 o Walfisch-Ikegami. El primer modelo añade las pérdidas por la influencia de los edificios a los cálculos, especificando previamente la altura y separación de estos edificios. El segundo modelo está basado en los modelos de Walfisch-Bertoni y de Ikegami. Este modelo buscaba mejorar la predicción del modelo de Walfisch-Bertoni incorporando algunas correcciones empíricas como son la anchura de las calles y la orientación de estas con respecto a la dirección de la línea que une transmisor y receptor o la transmisión por debajo de los edificios.

El uso de células más pequeñas es cada vez más común en el desarrollo de los sistemas de comunicaciones móviles modernos, dando lugar a entornos urbanos microcelulares. Este tipo de células presenta una cantidad de edificios menor que en los entornos macrocelulares, pero aun así la predicción de pérdidas en este tipo de células es compleja debido a que se busca una resolución mayor en la predicción de la cobertura radioeléctrica [1].

Para este tipo de entornos encontramos modelos empíricos, semi-empíricos y semi-deterministas. Algunos de estos modelos son:

- **Modelo de Kaji-Akeyama.** El modelo Kaji-Akeyama es un modelo empírico que se caracteriza por calcular el valor del potencial en el receptor [2].
- **Modelos Har-Xia-Bertoni.** El modelo Har-Xia-Bertoni es un modelo semi-empírico utilizado en entornos urbanos y suburbanos. Para establecer las ecuaciones de este modelo se llevó a cabo una campaña de medidas en San Francisco, utilizando las frecuencias de 900 y 1900 MHz y variando la altura del transmisor entre 3.2 y 13.4 metros.
- **Lanzado de rayos.** El método de lanzado de rayos es un modelo semi-determinista.

Esta técnica consiste en asumir las ondas electromagnéticas como si de rayos se tratasen. De esta forma no es necesario calcular las ecuaciones de Maxwell en los obstáculos del canal del tamaño aproximado a la longitud de onda. Este método también es empleado en entornos macrocelulares y en entornos de interiores.

2.3. Definición del trazado de rayos

Como se vio en el punto anterior, el trazado de rayos es un modelo semi-determinista empleado para predicción de la respuesta del canal de radiocomunicaciones. La base fundamental de este modelo es la aplicación de la Óptica Geométrica (GO) y la Teoría Uniforme de la Difracción (UTD). A medida que la frecuencia aumenta, la primera zona de *Fresnel*, que es la que contiene la mayor parte de la energía, tiende a estrecharse y es posible aproximarla a un rayo. Al propagarse las ondas electromagnéticas de esta forma es posible indicar que su comportamiento será similar a la trayectoria que recorrería un rayo óptico, simplificando en gran medida el análisis.

Los mecanismos fundamentales de propagación son la reflexión, la difracción y la difusión. La reflexión tiene lugar cuando una onda electromagnética incide sobre un obstáculo con dimensiones mayores a su longitud de onda. GO contempla la reflexión y la trata haciendo uso de la ley de Snell. El fenómeno de la difracción surge cuando un rayo es obstruido por una superficie con irregularidades abruptas, como pueden ser las esquinas formadas por dos paredes. En este caso se generan ondas secundarias desde la arista sobre la que impacta la onda original debido al principio de Huygens, siendo la UTD la teoría electromagnética que permite estimar el campo eléctrico difractado. Finalmente, las componentes difusas presentan un valor de potencia menor que el de las componentes reflejadas o difractadas en comunicaciones móviles, por lo que no se suele tener en cuenta en el trazado de rayos. La Figura 2.1 muestra la contribución de los diferentes tipos de rayos tanto en un modelo 2D como en un modelo 3D.

Existen dos formas de trazar rayos:

- La primera se denomina *Ray-Launching* o método de la “fuerza bruta” o de lanzamiento de rayos, que consiste en el lanzamiento de rayos desde la posición del transmisor con una separación angular constante y en su posterior seguimiento.
- La segunda de ellas es mediante el método de las imágenes. Este método se basa en generar imágenes a partir de las posibles reflexiones que pueden producirse en un entorno.

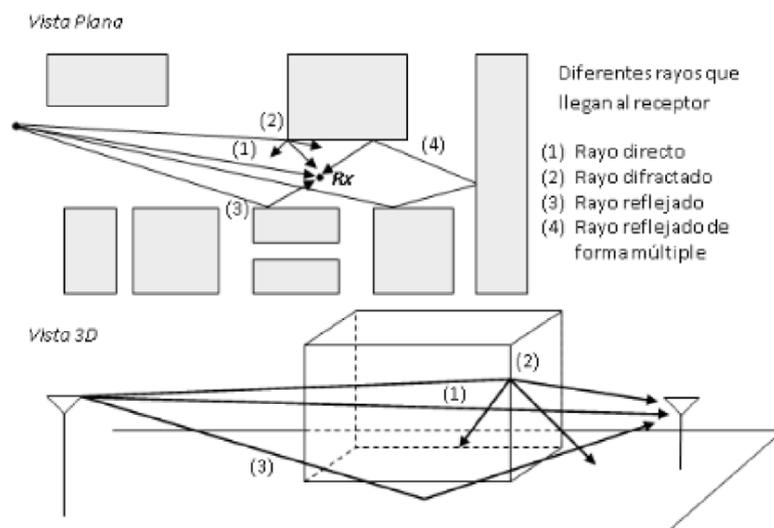


Figura 2.1: Tipos de rayos en un trazador de rayos[3]

La técnica del lanzamiento de rayos consiste en lanzar desde el transmisor un número finito de rayos y dejar que estos rayos se reflejen hasta llegar al receptor. En el receptor se considerará una esfera de incertidumbre que determinará si un rayo alcanza o no dicho receptor. Estas esferas receptoras describen las diferentes zonas que reciben únicamente un rayo. Un rayo se tomará como recibido si su trayectoria intercepta el círculo de recepción. Para la construcción del círculo de recepción es necesaria la definición del denominado radio de impacto. Si el radio de impacto es demasiado grande se tomarán en consideración muchos rayos en el cálculo del campo en el receptor,

mientras que si es muy pequeño se tomarán pocos rayos llegando incluso al extremo de que ningún rayo alcance el receptor. En la Figura 2.2 podemos observar el comportamiento seleccionando varios radios como esfera receptora. De esta forma es posible seleccionar un radio adecuado de impacto a partir de la separación angular de los rayos transmitidos (α) y de la distancia recorrida (d) por el rayo en su propagación[3].

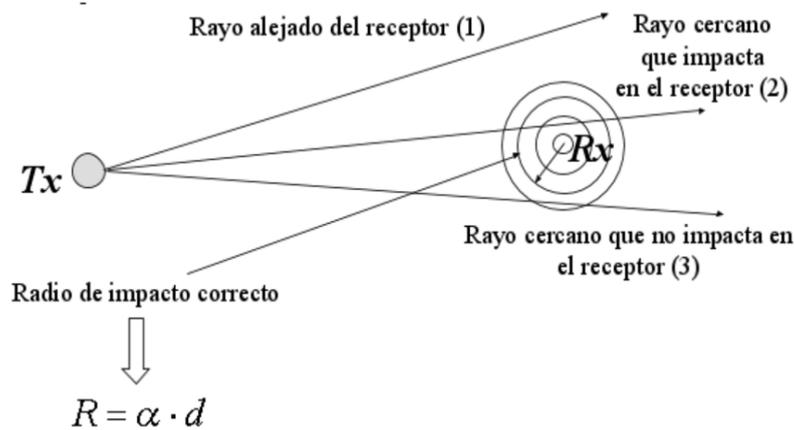


Figura 2.2: Diferentes radios de impacto en el receptor[3]

El trazado de rayos que se ha utilizado en este Proyecto hará uso del método de las imágenes para el cálculo de coberturas, debido a que este método se encuentra bien adaptado al análisis de la propagación de radio en entornos cuya complejidad sea baja y no tengamos un número excesivo de reflexiones. Cabe destacar que este método se podría aplicar perfectamente a entornos donde el número de reflexiones sea más elevado, pero el coste computacional requerido sería muy alto.

2.4. Técnica del trazado de rayos aplicando el método de las imágenes

El método de las imágenes es de vital importancia a la hora de analizar el trazador de rayos. Los rayos que se envían desde el transmisor pueden ser alterados considera-

blemente si existen obstáculos en el camino que estos recorren, especialmente si dicho obstáculo se encuentra en las inmediaciones del sistema radiante. Por lo general, el suelo es un medio con pérdidas cuya conductividad efectiva crece con la frecuencia, llegando a ser un buen conductor al superar una cierta frecuencia[1]. Con el objetivo de facilitar el análisis, se puede asumir que el suelo es un conductor perfecto, orientado horizontalmente y de infinita extensión.

El método de las imágenes introduce fuentes virtuales, denominadas imágenes, para analizar el comportamiento de un elemento radiante que se sitúa próximo a un conductor perfecto, en nuestro caso el suelo, permitiendo evaluar los rayos reflejados. Estas fuentes virtuales son fuentes ficticias y por lo tanto no son reales, pero al combinarlas dichas fuentes con las reales nos permitirían obtener un sistema equivalente, que reemplazaría al original únicamente con el objetivo de simplificar el análisis.

Asumiendo que un dipolo vertical (elemento radiante) está situado a una distancia h sobre un conductor perfecto, plano e infinito, y que dicho dipolo radia la energía en todas las direcciones, el observador situado en P_1 percibirá las contribuciones procedentes del rayo directo y del rayo reflejado en Q_{R1} . En la Figura 2.3 se observa como la energía en medios homogéneos describe trayectorias rectas. Por este motivo, la onda que llega al punto P_1 a partir del rayo reflejado parece ser originada por una fuente virtual situada a una distancia h por debajo del conductor. Si ahora el observado es P_2 , el punto de reflexión ha cambiado, pero la imagen no varía su posición siendo la misma de antes.

A modo de resumen, se puede decir que el método de las imágenes se basa en generar imágenes de todas las fuentes en todos los planos que pudiera tener el sistema. De esta forma si el sistema está compuesto por N planos a considerar obtendríamos N imágenes de primer orden ya que únicamente se tiene en cuenta la fuente radiante y el plano de interés sobre el que se produce la reflexión del rayo. Sin embargo, es posible

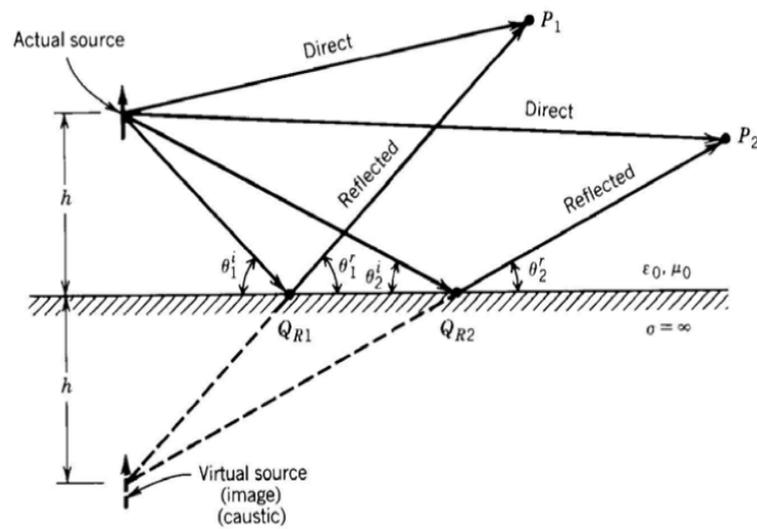


Figura 2.3: Método de las imágenes utilizado para encontrar la diferencia de caminos entre el rayo directo y el rayo reflejado[1]

que el rayo reflejado sufra un mayor número de reflexión en su trayectoria, dando lugar a imágenes de orden superior. Para calcular estas imágenes de orden superior ya no se utiliza el transmisor, sino que se trabaja con una imagen de orden inmediatamente inferior y con el nuevo plano de interés. La Figura 2.4. muestra un ejemplo donde se obtienen imágenes de primer y segundo orden.

El rayo describe una trayectoria desde la fuente radiante S hasta llegar al receptor en d . En esta trayectoria se produce una primera reflexión dando lugar a dos imágenes de primer orden S_A y S_B formadas a partir de los planos A y B respectivamente. Posteriormente se produce una segunda reflexión dando como resultado las imágenes de segundo orden S_{AB} y S_{BA} , donde ahora la fuente que produce la imagen no es la fuente S original, sino que se tratan de imágenes creadas a partir de las imágenes S_A y S_B . Es posible deducir a partir de este ejemplo que las imágenes de orden superior son imágenes que surgen a partir de otras imágenes, que a su vez pueden ser imágenes de otras imágenes dependiendo del orden de la imagen.

Es posible establecer el número de imágenes que existirán para cada orden si dis-

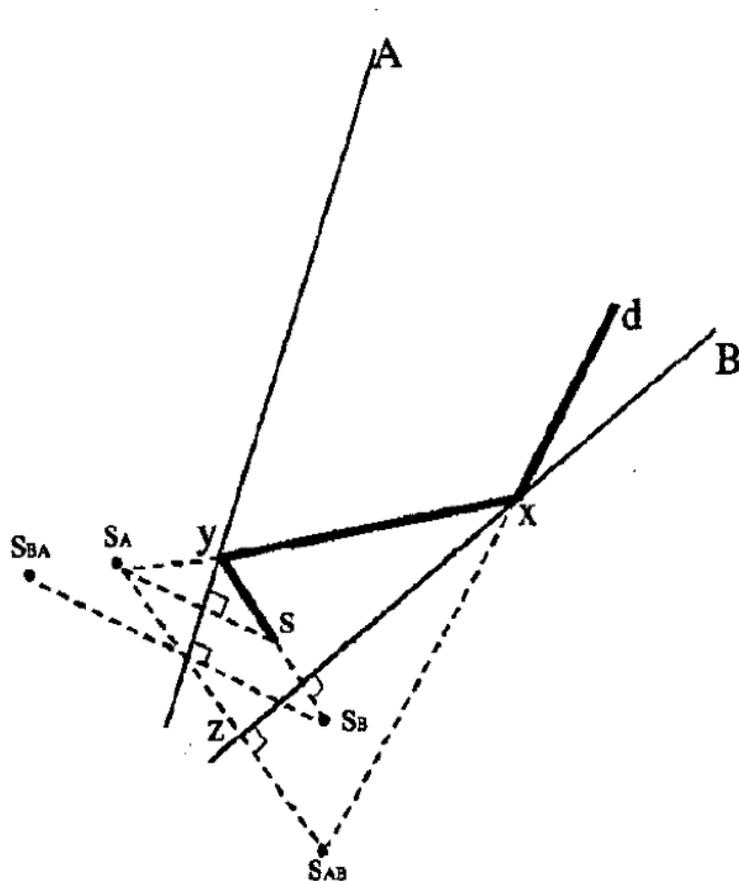


Figura 2.4: Imágenes de primer y segundo orden producidas por una fuente (S) situada entre dos espejos (A y B) [5]

ponemos de un sistema de N planos y una única fuente transmisora. Como se expuso anteriormente el número de imágenes de primer orden será de N , que tendrán como resultado $N(N - 1)$ imágenes de segundo orden, que a su vez serán fuentes de $N(N - 1)(N - 1)$ imágenes de tercer orden y así sucesivamente.

El método de las imágenes es un método determinista que ofrece una gran precisión en el cálculo de coberturas con una excelente eficiencia siempre que el entorno no sea excesivamente complejo. Si el entorno a simular es complejo existirán un gran número de elementos y obstáculos que producirán reflexión creando grandes cantidades de imágenes de orden superior a simular, elevando enormemente el coste computacional al aplicar el algoritmo.

Capítulo 3

Desarrollo de la aplicación a partir de los mapas de OpenStreetMap

3.1. El proyecto OSM

El proyecto OpenStreetMap nace en el mes de agosto del año 2004 a raíz de un experimento realizado con un receptor GPS por el programador británico Steve Coast. El modelo de datos inicial era muy rudimentario. Consistía en un conjunto de líneas dibujadas sobre imágenes de satélite Landsat. Con el paso del tiempo, la creación de datos evolucionó a un modelo más accesible, al mismo tiempo que se impusieron una serie de mejoras: el software inicial fue escrito en Java, posteriormente en Ruby, y finalmente en RoR (Ruby on Rails) que todavía es la plataforma empleada en la actualidad.

En 2006 nace la Fundación OpenStreetMap (OSMF) para promover la infraestructura necesaria que permitiese hacer funcionar el proyecto y conformar la estructura legal para poder aceptar cualquier tipo de donación económica por parte de los usuarios.

En octubre de 2014 en el proyecto se encontraban registrados en torno a 1.840.000 usuarios de los cuales alrededor de 22.600 habían realizado alguna edición en el último mes[6]. El número de usuarios no para de crecer continuamente fijándose en aproximadamente un 10 % de usuarios nuevos al mes. Por países el mayor número de ediciones

proviene de Alemania, Estados Unidos, Rusia e Italia. Cualquier usuario registrado puede subir sus trazas desde el GPS y crear y corregir datos vectoriales mediante herramientas de edición creadas por la comunidad de OpenStreetMap. Se estima que cada semana se añaden 90.000 km de nuevas carreteras con un total de casi 24.000.000 km de viales, sin contar otros tipos de datos (pistas, caminos, puntos de interés, etc.)[7]. El tamaño de la base de datos se situaba en julio de 2017 por encima de los 800 gigabytes.

3.2. Elementos que se encuentran representados en OSM

Cualquier elemento físico que se pueda imaginar y que se encuentre situado en el territorio puede ser digitalizado o registrado en la base de datos de OSM. Esto se puede llevar a cabo gracias a que existe un modelo de datos y una API tan sencilla que permite editar datos fácilmente, aunque varios usuarios se encuentren editando el terreno al mismo tiempo, y consigue mantener las propiedades topológicas entre los objetos [8].

Los elementos y principios básicos sobre los que se encuentra establecido el modelo de datos OSM son tres: nodos, vías y relaciones. Además, cualquiera de estos elementos geográficos puede ser descrito etiquetándolo con un par de valores clave. Es decir, “desde un punto de vista matemático, el modelo de datos de OSM es una especie de grafo consistente en un conjunto de vértices y bordes; diferentes partes del grafo pueden estar conectados, o pueden estar aislados, dependiendo del elemento del mundo real que estamos modelizando [9].

El formato que utiliza OSM para describir dichas relaciones o elementos es el lenguaje XML (*eXtensible Markup Language*, o Lenguaje de Marcas eXtensibles) que será el único formato en el que se podrán exportar los datos OSM.

```

<node id="982362611" visible="true" version="3" changeset="50500983" timestamp="2017-07-
23T12:46:13Z" user="Aroche" uid="50179" lat="37.6006540" lon="-0.9801556">
  <tag k="historic" v="archaeological_site"/>
  <tag k="name" v="Anfiteatro Romano"/>
  <tag k="wikidata" v="Q651415"/>
</node>

```

Figura 3.1: Cabeceras XML en un nodo

El ejemplo de nodo digitalizado anterior muestra como se le ha asignado un determinado identificador (*id*), así como sus coordenadas de latitud y longitud con siete decimales de precisión. A continuación se observa el usuario que ha editado el nodo y el momento en que se ha llevado a cabo dicha edición. Finalmente mediante etiquetas (*tags*) se establece la categoría a la que se ha añadido el nodo (*historic*, *archaeological_site*), el nombre que recibe (*Anfiteatro Romano*) y el código a wikidata para obtener mayor información acerca de este sitio.

La figura 3.2 muestra un mapa de OSM en el que se pueden observar diferentes nodos pertenecientes a todos los elementos que forman dicho mapa. Se distinguen nodos utilizados para establecer vías, paredes de edificios, distribución de parques, indicaciones de puntos de interés, etc.

Las vías se establecen mediante una secuencia ordenada de nodos que describen elementos de forma lineal como carreteras, caminos, ríos, etc. Éstas pueden formar elementos cerrados creando áreas o superficies. Esto sucede cuando el primer y el último nodo son el mismo, otorgando la posibilidad de etiquetar dicha vía como un área, siendo posible que su representación se establezca de forma diferente a una vía sin cerrar (por ejemplo, la muralla que rodea la UPCT). Un nodo puede formar parte de más de una vía, siempre que ambas vías compartan al menos un nodo. Además, si todos los nodos que establecen una línea son desplazados, pero mantienen su identificador no creará

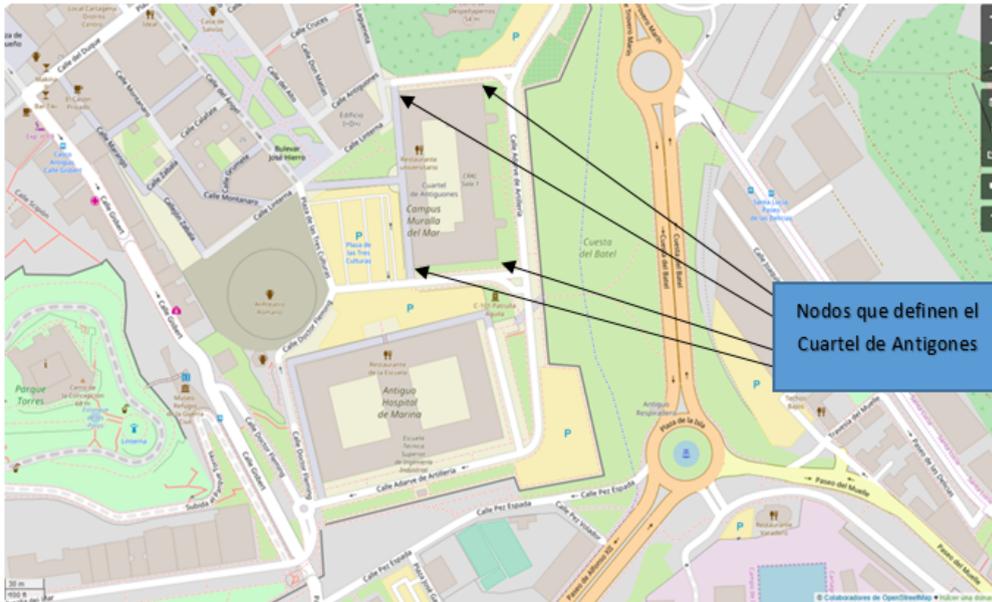


Figura 3.2: Ejemplos de nodos en un mapa OSM. Se muestran los nos que definen el Cuartel de Antigones

una nueva versión de la vía, sino que ésta se verá modificada. El aspecto de una vía en lenguaje XML se ve representado en la figura 3.3.

```
<way id="106560156" visible="true" version="9" changeset="29978057" timestamp="2015-04-04T18:21:00Z" user="KikeTM" uid="1703782">
  <nd ref="187177945"/>
  <nd ref="3437824186"/>
  <nd ref="2369061933"/>
  <nd ref="250317221"/>
  <nd ref="3437824185"/>
  <nd ref="1225793711"/>
  <tag k="highway" v="primary"/>
  <tag k="lanes" v="2"/>
  <tag k="maxspeed" v="50"/>
  <tag k="name" v="Paseo de Alfonso XIII"/>
  <tag k="oneway" v="yes"/>
</way>
```

Figura 3.3: Cabeceras XML en una vía

El ejemplo de código XML mostrado en la Figura 3.3 muestra información acerca del identificador del elemento (`way_id`), el usuario editor y la fecha de la edición. Así

mismo, se describe el identificador de cada uno de los nodos que forman la vía o la superficie cerrada. En este caso la vía del ejemplo es el paseo de Alfonso XIII y viene especificado en sus etiquetas, junto al resto de características existentes en esta vía.



Figura 3.4: Ejemplos de vías en un mapa OSM. La vía mostrada es el Paseo de Alfonso XIII en Cartagena

El último elemento sobre el que se basa OSM es el uso de relaciones. Una relación agrupa a varios elementos, como pueden ser nodos y/o vías. *“Los elementos son ‘miembros’ de una relación y cada uno de los miembros tiene un ‘rol’ ”*[9].

Una relación puede tener un determinado número de etiquetas. Normalmente una relación posee una etiqueta *type* (tipo) que define el tipo de elementos. Podemos encontrar relaciones de todo tipo, desde *tags* que indican los nodos que pertenecen a un mismo edificio, hasta representaciones de los límites de una isla o la representación de rutas con carril bici.

Citada anteriormente, *“una etiqueta es una propiedad que se asocia a un elemento para definir una característica de éste”* [9]. Por ejemplo la etiqueta *high-way=residential*

establece que una vía es una calle urbana. OSM aconseja aplicar determinadas etiquetas en función del tipo de elemento representado con el fin de establecer de forma precisa las características de dicho elemento.

3.3. Importación de los mapas OSM en MATLAB

Para llevar a cabo la importación de los archivos en formato XML propia de los mapas de MATLAB, se va a hacer uso del conjunto de funciones desarrolladas por Ioannis F. Filippidis [©]. El archivo en formato XML es cargado en MATLAB y posteriormente a través de un script se extrae la información que contiene el archivo. Esta información se guarda como una estructura de Matlab. A continuación, esta información puede ser utilizada para representar cada uno de los nodos y su relación entre ellos llegando a obtener un modelo 2D del terreno con un buen nivel de detalle.

3.3.1. Extracción de los datos a partir del archivo OSM

Para extraer los datos de archivo XML de OSM, desde ahora XML OSM, deberemos hacer uso de la función *parse_openstreetmap*, con el comando *parsed_osm = parse_openstreetmap(openstreetmap_filename)*, introduciendo el archivo XML OSM descargado previamente desde la página web de OpenStreetMap. El archivo XML OSM es el argumento de entrada de la función en formato *string*. Esta función nos proporcionará la estructura de MATLAB *parsed_osm* que contendrá el subconjunto de datos necesarios para la representación. La estructura de los campos se muestra en el diagrama de la Figura 3.5.

La estructura de los campos se muestra en el diagrama de la Figura 3.5, donde cada uno de los campos realiza la siguiente función:

- El primer elemento es el campo *bounds* cuya única finalidad es establecer los límites de la representación a realizar.

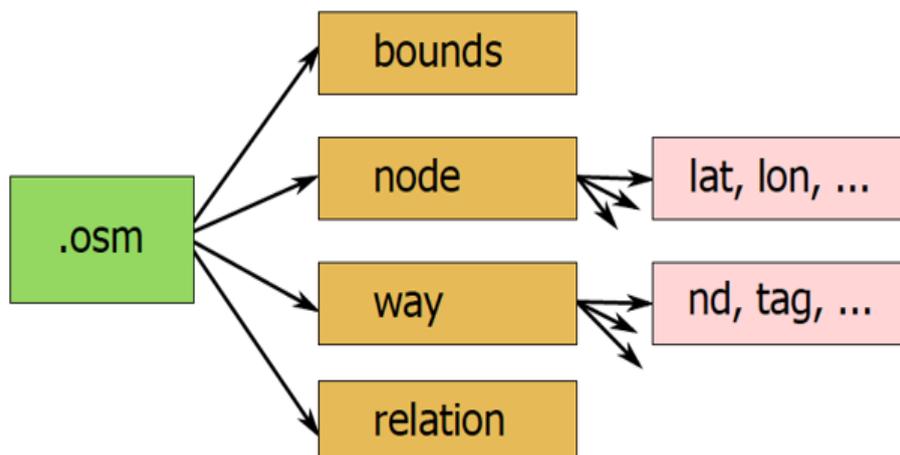


Figura 3.5: Estructura del archivo OSM en MATLAB[10]

- El segundo elemento es el campo *node* que contiene tanto el identificador de cada uno de los nodos (*id*), como la posición en la que se sitúan en coordenadas geográficas (*xy*), es decir, en términos de latitud/longitud.
- El tercer elemento es el campo *way*, que a su vez se encuentra dividido en otros tres subcampos. El subcampo *id* contienen el identificador de cada una de las relaciones entre nodos para identificarlas de forma unívoca. El subcampo *nd* muestra el conjunto de nodos que forman cada una de las relaciones. Finalmente, el subcampo *tag* muestra las etiquetas que caracterizan cada una de las relaciones de nodos.
- Por último, el campo *relation* solamente contiene atributos relacionado con la versión y direcciones de almacenamiento que utiliza OpenStreetMap.

3.3.2. Representación del entorno original

En este apartado se llevará a cabo la representación del archivo OSM importado anteriormente haciendo uso de la estructura obtenida en la sección anterior en MATLAB.

La función encargada de realizar la representación es `plot_way(ax, parsed_osm, map_img_filename)`.

El argumento de entrada `ax` es un objeto que contiene las propiedades de los ejes sobre los que se llevará a cabo la representación. El argumento `parsed_osm` es el objeto con estructura MATLAB que nos devolvió la función `parse_openstreetmap` mediante la extracción de los datos. Finalmente, el argumento `map_img_filename` es opcional y carga una imagen en el fondo del entorno a representar. La Figura 3.6. muestra la representación de un archivo OSM de una parte de la ciudad de Cartagena, donde se muestran todos los elementos que posee OSM del terreno utilizando unicamente nodos y uniones entre ellos a partir de las características propias de cada nodo.

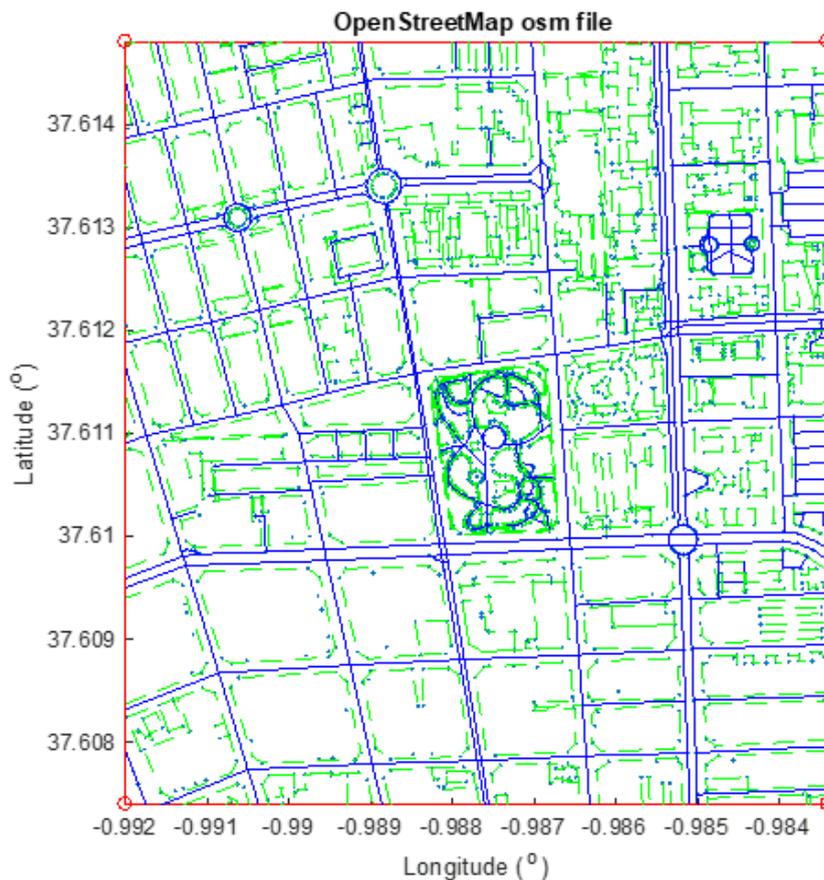


Figura 3.6: Ejemplo de representación haciendo uso de la función `plot_way`

A partir de estos datos es posible obtener un modelo 2D de cualquier zona urbana y posteriormente realizar una estimación de cómo sería dicho modelo en 3D. Hay que destacar que a partir de la Figura 3.6. se observa como el número de nodos y elementos es muy alto, por lo que será necesario llevar a cabo una simplificación previa del modelo.

3.4. Adaptación de los mapas OSM al trazador de rayos

3.4.1. Introducción a la herramienta de trazado de rayos

El trazador de rayos hace uso de la teoría de imágenes lo que presenta unas limitaciones asociadas, aunque no sean totalmente críticas, que debemos tener en cuenta a la hora de desarrollar nuestro entorno a simular para llevar a cabo el cálculo de coberturas. Al tratarse de un modelo determinista se nos presenta la posibilidad de obtener resultados con una gran precisión si nos encontramos con unas condiciones de trabajo adecuadas para el uso de esta técnica.

El coste computacional de realizar una simulación con un entorno urbano de dimensiones muy grandes con gran cantidad de elementos puede ser desorbitado. Por esto, el escenario donde mejor se puede desenvolver la herramienta de trazado de rayos es un entorno de geometría no excesivamente compleja, donde tanto el número de reflexiones como el de difracciones no sea muy elevado. Esta simplificación del entorno no tiene porque ser un problema ya que podemos seleccionar los elementos que tendrán una mayor contribución en nuestro caso de estudio. Asimismo, hay que matizar que estas limitaciones son de nivel computacional ya que en principio el trazador de rayos debería funcionar correctamente el cualquier entorno debido a que únicamente presenta las limitaciones propias del método ya que al fin y al cabo, realiza aproximaciones sobre la propagación de las ondas electromagnéticas, pero el número de operaciones requerido puede ser extremo.

A la hora utilizar el trazado de rayos existe una serie de matices que debemos tener en cuenta para implementar su desarrollo:

- La descripción más óptima del entorno a simular, lo que conlleva introducir las características eléctricas del material que forman las superficies que establecen nuestro entorno a simular.
- La correcta interpretación del entorno que deseamos simular. Esto es debido a que los parámetros del canal radio tiene una alta dependencia de la localización y orientación de los obstáculos.
- Establecer el número de mecanismos de propagación. En el caso de nuestra herramienta de trazado de rayos estableceremos el número de reflexiones máximas que sufrirá cada rayo.
- La posición en la que se encuentra el transmisor y el receptor, pudiendo ser esta fija o móvil. En caso de considerar una posición móvil es necesario indicar las diferentes posiciones en las que se encuentra el transmisor/receptor pudiendo describir la ruta arbitraria que se desee.
- La correcta implementación de un algoritmo de recepción, con el objetivo de evitar la repetición de alguna contribución, es decir, la aparición de rayos redundantes.
- El cálculo de las reflexiones se realiza empleando las leyes de Snell y la Teoría de Imágenes u otro modelo.
- Un algoritmo y una formulación apropiada para los casos en los que existen rayos difractados en las esquinas del entorno.
- Establecer una condición de finalización de la propagación para un rayo determinado, que puede ser debida a la llegada del número máximo de reflexiones, a

una potencia mínima discernible en el receptor o porque el rayo no intersecta con ninguna superficie.

- La frecuencia de la simulación, si ésta es en banda estrecha, o la frecuencia portadora y el ancho de banda de la señal si la simulación es de banda ancha.

Los puntos que se mencionaron anteriormente son la base sobre las que se centra la herramienta de simulación del trazado de rayos. En los puntos siguientes se mostrará como se han modificado los datos de los entornos OSM, con el fin de adaptarlos correctamente a las características del trazador. De esta forma podremos obtener simulaciones de entornos lo más fieles posibles a la realidad.

3.4.2. Representación del entorno en la herramienta de trazado de rayos

El entorno que se podría utilizar para realizar una simulación haciendo uso del trazado de rayos podría ser de cualquier tipo, es decir, teóricamente no existen más limitaciones que las restricciones computacionales o las que presente el lenguaje de desarrollo. Centrándonos en la primera restricción, si el entorno es de una complejidad muy elevada, el número de operaciones será también muy elevado. La segunda restricción está ligada al lenguaje de desarrollo que se empleo para implementar la herramienta del trazado de rayos. Es posible que este no nos permita definir estructuras con cierta geometría, como pueden ser superficies esféricas o redondeadas. La definición de ciertas superficies puede ser muy laboriosa o incluso no estar implementada en la herramienta de trazado que vamos a utilizar para simular nuestro entorno.

Por lo expuesto anteriormente, es necesario conocer correctamente como trata las superficies de los entornos la herramienta que vamos a utilizar. Así podremos representar el entorno de forma fiel a la realidad. En caso contrario, podríamos estar esta-

bleciendo superficies no validas para el trazador o añadiendo superficies definidas pero no correctamente llevándonos a contribuciones de rayos erróneas.

La herramienta de trazado que empleamos es capaz de representar cualquier entorno que se desee, siempre y cuando se empleen superficies rectangulares definidas por sus cuatro vértices, o bien, superficies triangulares definidas por tres vértices. Cada uno de estos vértices estarán compuestos por sus tres coordenadas XYZ agrupadas en una matriz de coordenadas.

El disponer unicamente de superficies triangulares y rectangulares hace muy difícil que en la simulación de la propagación radioeléctrica empleemos una descripción de entorno totalmente fiel, ya que la representación exacta sería muy compleja. En la herramienta que vamos a desarrollar intentaremos conseguir cualquier tipo de entorno, sea cual sea la geometría de su superficie, que se encuentre disponible en OSM a partir de rectángulos y triángulos.

El primer paso será comprender como quedan definidas cada una de las superficies rectangulares en MATLAB. Como se expuso anteriormente, cada rectángulo queda definido por cuatro vértices, estableciendo sus tres coordenadas para cada uno. A continuación será necesario asignar las características radioeléctricas propias del material que forma la superficie. Todos estos datos para una superficie quedan recogidos en una matriz. Por lo tanto el entorno quedará recogido en una matriz multidimensional donde el número de elementos de la última dimensión indica el número de superficies.

La Figura 3.7 muestra el formato que establece el trazador de rayos para poder definir superficies rectangulares. Por lo tanto, las matrices estarán formadas por tres filas y seis columnas que contendrán tanto la información espacial (coordenadas) como la correspondiente a la naturaleza de la superficie. Cada campo de la matriz queda definido de la siguiente forma[11] :

	Vértice 1	Vértice 2	Vértice 3	Vértice 4	5	6
1	Coord. X	Coord. X	Coord. X	Coord. X	Hor/Vert	At
2	Coord. Y	Coord. Y	Coord. Y	Coord. Y	ϵ_r	Perf_cond
3	Coord. Z	Coord. Z	Coord. Z	Coord. Z	σ	-

Figura 3.7: Formato de las paredes rectangulares de la estructura[11]

- Las cuatro primeras columnas se reservan para almacenar las coordenadas XYZ de los cuatro vértices que forman la superficie. Todos los valores de posición que utiliza el trazador deben estar incluidos en metros (m).
- La primera fila de la quinta columna establece la orientación de la pared rectangular. Este campo tomará un valor igual a 0 si la pared es horizontal, o bien, tomará el valor igual a 1 si se trata de una pared vertical.
- La segunda fila de la quinta columna almacena el valor de la permitividad relativa de dicha pared.
- La tercera fila de la quinta columna contiene la conductividad de la pared en Siemens por metro (S/m).
- La primera fila de la sexta columna almacena la atenuación que sufrirá un rayo al atravesar la superficie en transmisión. El valor se almacena en dBs.
- La segunda fila de la sexta columna de la matriz indica si el material se trata de un conductor perfecto o no. Si se rellena con un 1 el material será un conductor perfecto, mientras que si es 0 no lo será.
- Como podemos observar el campo de la tercera fila de la sexta columna no se utiliza y queda a disposición del usuario para utilizarlo como se desee. En nues-

tro caso haremos uso de él para indicar el número de edificio al que pertenece la superficie. De esta forma conoceremos todas las superficies que forman cada edificio.

	1	2	3	4	5	6
1	-5	-5	-5	-5	1	2
2	-5	5	5	-5	4	0
3	0	0	5	5	0.01	0

Figura 3.8: Ejemplo de pared rectangular definida en MATLAB[11]

La Figura 3.8 muestra un ejemplo de pared definida que cumple con todos los campos expuestos anteriormente. En este caso pared se encontraría orientada verticalmente ya que el valor de la primera fila quinta columna es igual a uno. Además, la permitividad relativa es 4, la conductividad es igual a $0.01 S/m$ y que los rayos que sufran en ella el fenómeno de transmisión se atenúan 2 dBs por unidad de longitud (por metro) de superficie atravesada.

Hasta ahora simplemente se han indicado las coordenadas de los cuatro vértices que forman nuestra superficie rectangular, pero no se ha mencionado nada del orden determinado en el que se deben introducir dichos vértices. Para que nuestra herramienta de trazado comprenda de forma correcta la superficie rectangular debemos introducir los vértices en un orden determinado. Cada uno de los vértices debe ser introducido de forma consecutiva al anterior siguiendo el perímetro de la superficie. De esta forma nos aseguramos que no existen cruces entre rectas.

La Figura 3.9 muestra el efecto de cruces de rectas producido por la introducción de manera errónea de los vértices. En este caso se han intercambiado las columnas 2 y 3 de

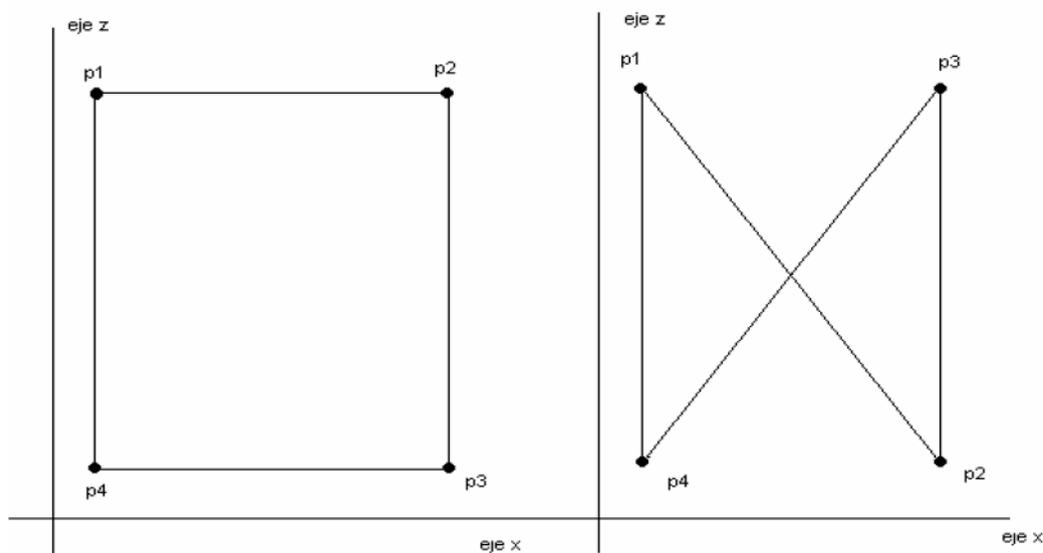


Figura 3.9: Efecto de cruce de rectas en un error de definición de paredes[11]

la matriz correspondiente a la pared. De esta forma no se respeta la introducción de puntos de forma consecutiva al perímetro de la superficie, provocando la interpretación incorrecta por parte del trazador, obteniendo errores al realizar la simulación deseada.

El segundo tipo de superficies con las que trabaja el trazador de rayos son aquellas que tienen una forma geométrica triangular. Lógicamente, será necesario representar este tipo de paredes de forma diferente a las superficies rectangulares. En este caso solamente contamos con tres vértices que definen la pared de forma unívoca. Además, en este caso no existirá el problema de cruces de rectas, debido a que es indiferente como se definan los vértices ya que estos siempre serán consecutivos.

Si nos centramos en la estructura de la matriz que define este tipo de paredes en MATLAB será similar a la que tenían las superficies rectangulares, es decir, se mantiene el formato de 3 filas y 6 columnas para definir cada pared. El único cambio que implementa este tipo de paredes es establecer todos los campos de la cuarta columna, correspondiente al vértice sobrante, como Not a Number (NaN). La Figura 3.10 muestra como quedaría definida la estructura para este tipo de paredes.

	Vértice 1	Vértice 2	Vértice 3	Vértice 4	5	6
1	Coord. X	Coord. X	Coord. X	NaN	Hor/Vert	At
2	Coord. Y	Coord. Y	Coord. Y	NaN	ϵ_r	Perf_cond
3	Coord. Z	Coord. Z	Coord. Z	NaN	σ	-

Figura 3.10: Formato de las paredes rectangulares de la estructura[11]

Tal como se realizó con las superficiales rectangulares, la Figura 3.11 trata de mostrar un ejemplo de matriz que define una pared de este tipo de forma correcta. A partir de esta matriz, la Figura 3.12 representa de forma gráfica la interpretación que realiza el trazador de rayos de dicha superficie.

	1	2	3	4	5	6
1	5	10	7.5000	NaN	1	10
2	5	10	7.5000	NaN	4	10
3	0	0	5	NaN	0.0100	0

Figura 3.11: Ejemplo de pared triangular definida en MATLAB

Una vez conocido como trata cada una de las superficies por separado la herramienta de trazado de rayos solamente debemos generar tantas matrices como paredes formen nuestro entorno. En MATLAB implementaremos un script que almacena estas matrices en "*planes*". Luego el formato que tendrá será "planes(:, :, [número de paredes])."

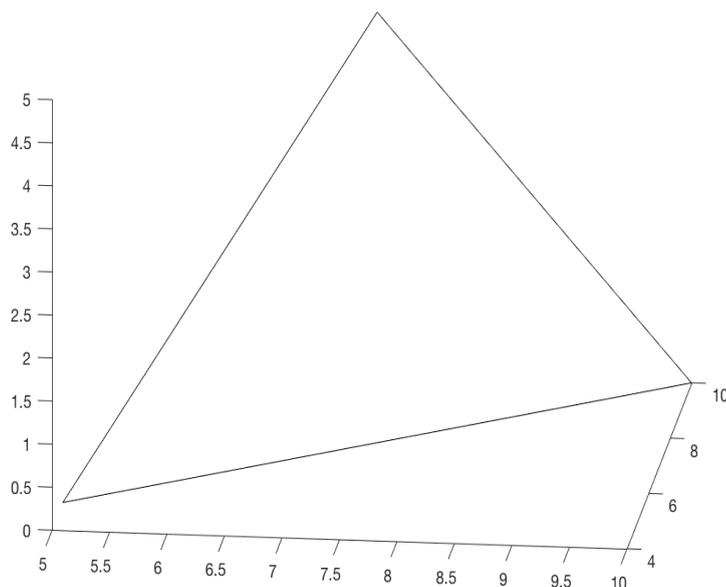


Figura 3.12: Representación de la pared triangular definida en MATLAB

3.4.3. Selección de elementos de interés en el entorno urbano

Como pudimos observar en 3.3.2 al llevar a cabo la representación en MATLAB de un entorno urbano a partir del mapa obtenido de OpenStreetMap para el ejemplo de la Figura 3.6, el número de elementos que definen el entorno es muy elevado. Esto hace inviable intentar realizar una simulación con nuestra herramienta de trazado de rayos directamente sin una simplificación previa.

En este tipo de entornos se caracteriza por contar con una gran densidad de edificios, siendo estas estructuras los elementos que más influirán al realizar la simulación. Por tanto sería interesante como primer paso para la simplificación de nuestro modelo poder establecer un primer entorno en 2D únicamente con los edificios de la zona, eliminando el resto de elementos cuya contribución será de menor importancia.

Según se vio en 3.3.1, los datos que conseguimos exportar de OSM se establecían

en varios campos. El campo *way* era el encargado de almacenar las relaciones entre los nodos existentes. Sería interesante seleccionar desde este campo aquellos conjuntos de nodos cuyas relaciones establezcan superficies que sean de nuestro interés, en este caso, los edificios de la zona.

Si observamos el subcampo *tag* contenido en *way*, encontramos gran cantidad de etiquetas para clasificar los conjuntos de nodos, siendo una de ellas la etiqueta, "*building*" que únicamente se encuentra en los conjuntos de nodos que forman un edificio. A partir de este momento el problema de simplificado se limita a eliminar todos aquellos conjuntos que no contengan esta etiqueta, almacenando los que si la tienen.



Figura 3.13: Representación de los conjuntos de nodos con la etiqueta *building* de la zona de Cartagena mostrada en la Figura 3.6

La Figura 3.13 muestra el mismo escenario que se mostró en 3.3.2 una vez se ha modificado la función *plot_way* para que unicamente seleccione los conjuntos de nodos con la etiqueta *building*. Como se puede observar, ahora disponemos de un entorno mucho más factible para trabajar con nuestra herramienta de trazado de rayos al seleccionar aquellos elementos más importantes.

3.4.4. Conversión de coordenadas geográficas a coordenadas UTM

OpenStreetMap utiliza las coordenadas geográficas basadas en latitud y longitud como sistema de coordenadas que permita establecer la ubicación de cada nodo en cualquier punto de la Tierra. Este tipo de sistemas de coordenadas se centra en ángulos esféricos o esferóides cuyo centro es el centro de la Tierra y se expresan en grados sexagesimales, donde:

- La latitud de cualquier punto en la superficie de la Tierra es el ángulo entre el plano ecuatorial y la línea que pasa por ese punto y el centro de la Tierra. El ecuador es el paralelo 0° y divide el globo en Norte y Sur, de forma que el polo norte es 90° N y el polo sur es 90° S.
- La longitud de un punto en la superficie de la Tierra es el ángulo entre el meridiano de referencia y el que pasa por este punto. El meridiano que se toma como referencia es el de Greenwich. Este meridiano divide el globo terrestre en los hemisferios este y oeste.

A partir de la Figura 3.14 es fácil deducir que un grado no se corresponde con la misma distancia en kilómetros en la zona del ecuador ya que será mayor que en uno de los polos. Luego la distancia en este sistema de coordenadas depende de la latitud, es decir, a medida que la latitud aumenta, hacia Norte o Sur, disminuyen los kilómetros

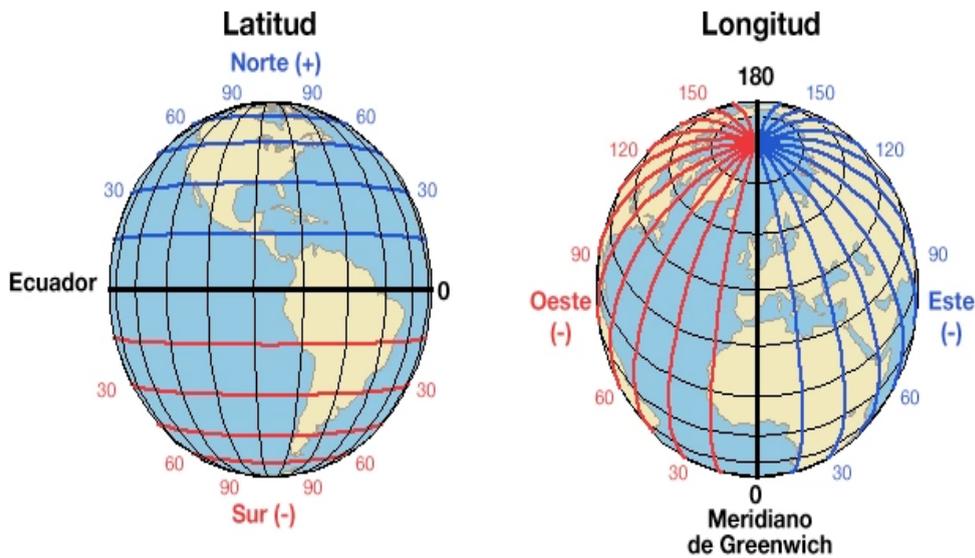


Figura 3.14: Sistemas de coordenadas geográficas

por grado. Para el paralelo del Ecuador, sabiendo que la circunferencia que corresponde a este paralelo es de 40 075,017 km, un grado equivale a 111,319 km. [15]

La herramienta de trazado de rayos hace uso de entornos cuyas distancias se establecen en metros. Por lo visto anteriormente, el sistema de coordenadas geográficas basado en latitud/longitud no es adecuado para realizar una conversión correctamente a metros. Como alternativa se hará uso de un sistema de coordenadas UTM (*Universal Transverse Mercator*).

El sistema de coordenadas UTM, a diferencia del sistema de coordenadas geográficas, se expresa en metros. Este sistema divide la Tierra en 60 husos de 6° de longitud. Cada huso tiene asignado un meridiano central, que es donde se sitúa el origen de coordenadas, junto con el ecuador.

A parte de esta división en husos, la Tierra también se divide en 20 bandas de 8° de latitud, que se denominan con letras desde la *C* hasta la *X* excluyendo la *I*, la *O* y la \tilde{N} .

Cada cuadrícula UTM se define mediante el número de huso y la letra de la zona. La división de la Tierra en las diferentes zonas UTM se muestra en la Figura 3.15.

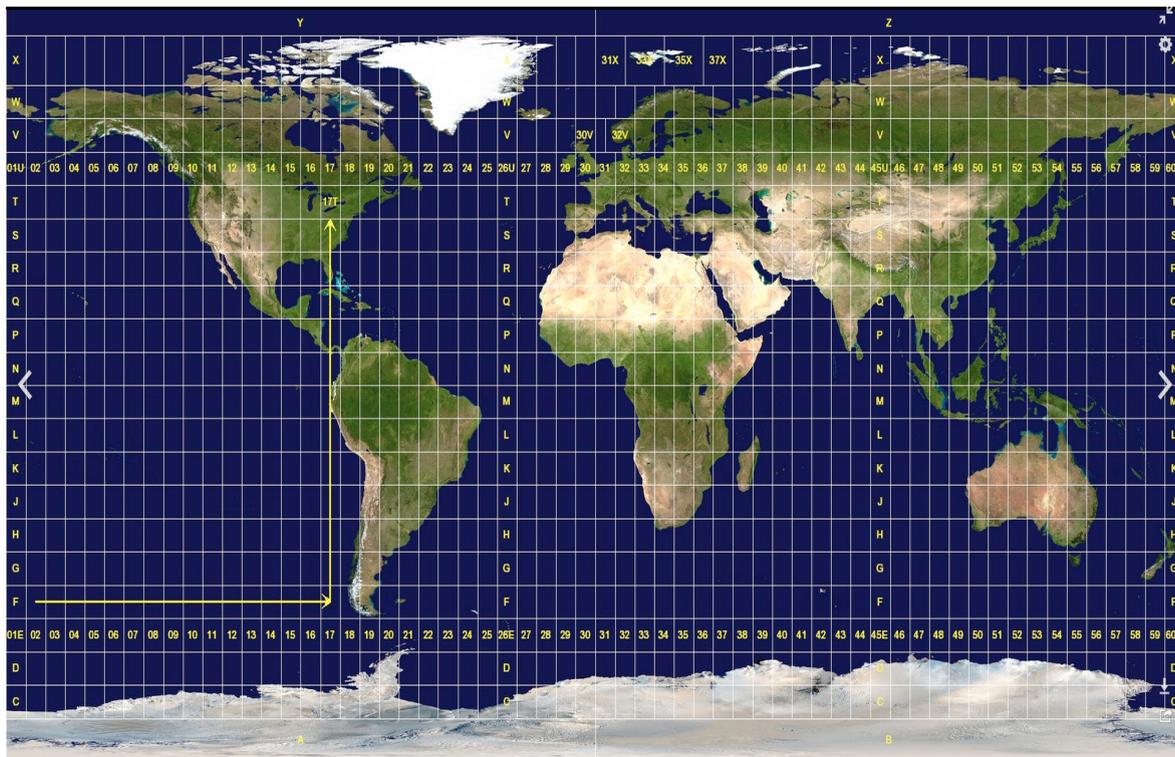


Figura 3.15: Sistemas de coordenadas UTM

Para el correcto funcionamiento del trazador, será necesaria la conversión de coordenadas geográficas extraídas de los nodos de OSM cuyos parámetros son latitud y longitud, a los coordenadas del sistema UTM en metros. Para llevar a cabo este proceso se emplea la función *deg2utm* que se muestra a continuación:

Función de conversión de latitud/longitud a UTM

```

1 function [x,y,utmzone] = deg2utm(Lat, Lon)
2
3
4 error(nargchk(2, 2, nargin));
5 n1=length(Lat);
6 n2=length(Lon);
7 if (n1~=n2)

```

```

8     error('Lat and Lon vectors should have the same length');
9 end
10
11
12
13
14 x=zeros(n1,1);
15 y=zeros(n1,1);
16 utmzone(n1,:)='60 X';
17
18
19 for i=1:n1
20     la=Lat(i);
21     lo=Lon(i);
22
23     sa = 6378137.000000 ; sb = 6356752.314245;
24
25
26     e2 = ( ( ( sa ^ 2 ) - ( sb ^ 2 ) ) ^ 0.5 ) / sb;
27     e2cuadrada = e2 ^ 2;
28     c = ( sa ^ 2 ) / sb;
29
30
31     lat = la * ( pi / 180 );
32     lon = lo * ( pi / 180 );
33
34     Huso = fix( ( lo / 6 ) + 31);
35     S = ( ( Huso * 6 ) - 183 );
36     deltaS = lon - ( S * ( pi / 180 ) );
37
38     if (la<-72), Letra='C';
39     elseif (la<-64), Letra='D';
40     elseif (la<-56), Letra='E';
41     elseif (la<-48), Letra='F';
42     elseif (la<-40), Letra='G';
43     elseif (la<-32), Letra='H';
44     elseif (la<-24), Letra='J';
45     elseif (la<-16), Letra='K';
46     elseif (la<-8), Letra='L';
47     elseif (la<0), Letra='M';
48     elseif (la<8), Letra='N';
49     elseif (la<16), Letra='P';
50     elseif (la<24), Letra='Q';
51     elseif (la<32), Letra='R';
52     elseif (la<40), Letra='S';
53     elseif (la<48), Letra='T';

```

```

54     elseif (la < 56), Letra='U';
55     elseif (la < 64), Letra='V';
56     elseif (la < 72), Letra='W';
57     else Letra='X';
58     end
59
60     a = cos(lat) * sin(deltaS);
61     epsilon = 0.5 * log( ( 1 + a ) / ( 1 - a ) );
62     nu = atan( tan(lat) / cos(deltaS) ) - lat;
63     v = ( c / ( ( 1 + ( e2cuadrada * ( cos(lat) ) ^ 2 ) ) ) ^
64           0.5 ) * 0.9996;
65     ta = ( e2cuadrada / 2 ) * epsilon ^ 2 * ( cos(lat) ) ^ 2;
66     a1 = sin( 2 * lat );
67     a2 = a1 * ( cos(lat) ) ^ 2;
68     j2 = lat + ( a1 / 2 );
69     j4 = ( ( 3 * j2 ) + a2 ) / 4;
70     j6 = ( ( 5 * j4 ) + ( a2 * ( cos(lat) ) ^ 2 ) ) / 3;
71     alfa = ( 3 / 4 ) * e2cuadrada;
72     beta = ( 5 / 3 ) * alfa ^ 2;
73     gama = ( 35 / 27 ) * alfa ^ 3;
74     Bm = 0.9996 * c * ( lat - alfa * j2 + beta * j4 - gama * j6
75           );
76     xx = epsilon * v * ( 1 + ( ta / 3 ) ) + 500000;
77     yy = nu * v * ( 1 + ta ) + Bm;
78
79     if (yy < 0)
80         yy = 9999999 + yy;
81     end
82
83     x(i) = xx;
84     y(i) = yy;
85     utmzone(i, :) = sprintf( '%02d %c', Huso, Letra );
86 end

```

Esta función toma como parámetros de entrada un array de latitudes y otro de longitudes correspondientes a la posición geográfica de cada uno de los nodos. Como salida nos devolverá de nuevo dos arrays, con las mismas posiciones geográficas pero en este caso en el sistema de coordenadas UTM.

Al conocer los límites de la representación recogidos en el campo *bounds* de los datos exportados resulta interesante llevar a cabo la conversión al sistema UTM de estos puntos. Si hacemos esto, podemos normalizar el resto de nodos con respecto a

dichos límites, obteniendo ahora una representación que muestran la distancia real en metros. De esta forma, el usuario tendrá una visión más amigable de la distancia real que existen entre los diferentes elementos.

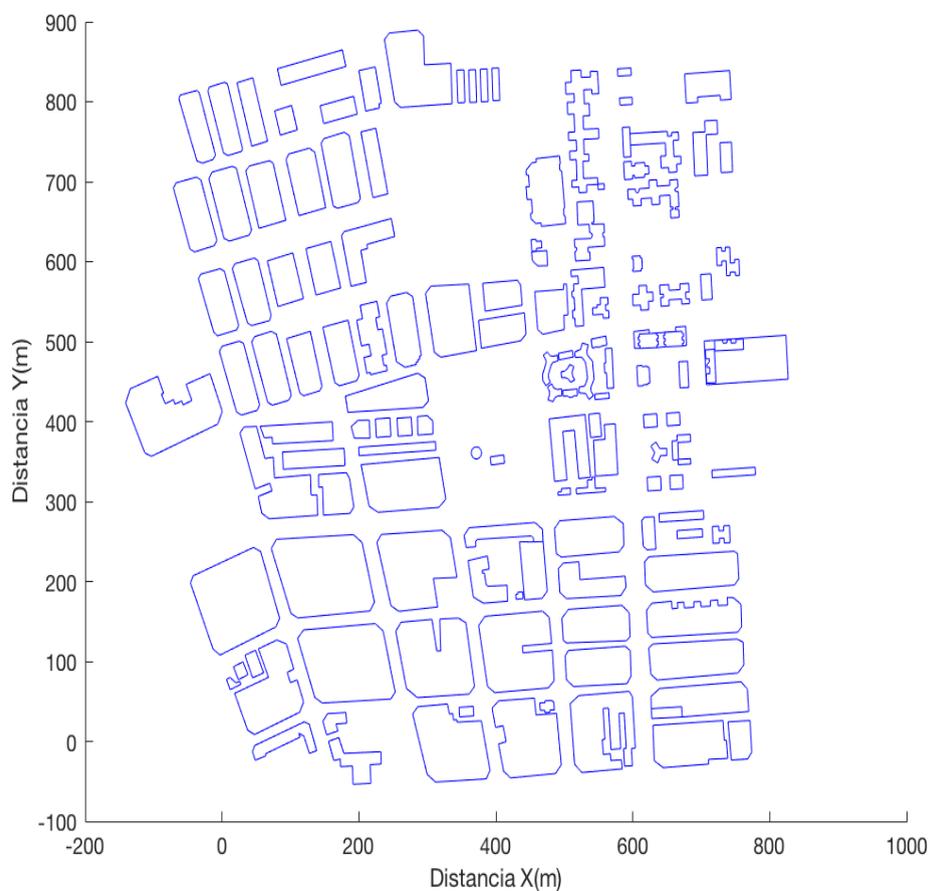


Figura 3.16: Representación del entorno en metros

La Figura 3.16 muestra una representación del entorno simulado en apartados anteriores al que se le aplicó esta conversión de coordenadas y posteriormente se han normalizado haciendo uso de sus límites de representación.

3.4.5. Implementación de las superficies verticales

La implementación de las paredes verticales será realizado modificando la función original *plot_way*. El primer paso será replicar el entorno 2D que tenemos a la altura de los edificios deseada por el usuario. OpenStreetMap no proporciona ningún dato acerca de la altura de los edificios en los entornos urbanos que dispone. El usuario cuenta con tres opciones a la hora de establecer la altura de los edificios:

- Introducir una altura global para todos los edificios, a partir de información externa que proporcione la altura media de los edificios de la zona.
- Introducir la altura de los edificios manualmente, es decir, uno a uno. Si disponemos de un entorno muy pequeño, esta sería la mejor opción, aunque cuando el tamaño del entorno crece es laborioso introducir todas las alturas.
- Introducir una altura media global para todos los edificios y, posteriormente modificar la altura de aquellos edificios más influyentes. En general, esta sería la mejor opción para entornos complejos.

La Figura 3.17 muestra, por así decirlo lo que sería el *techo* y el *suelo* de nuestra selección de edificios. Para establecer las paredes verticales debemos unir cada punto del *techo* con su correspondiente en la coordenada XY en el *suelo*.

Este proceso lo realiza la función *generaplanes* llamada directamente al utilizar *plot_way*. La función *generaplanes* se encarga de establecer la primera versión de *planes* que solo contiene, por el momento, las superficies verticales del entorno.

La herramienta de trazado de rayos hacía uso de matrices con 3 filas y 6 columnas, incorporando tantas matrices como superficies se desearan representar. La función *generaplanes* modifica las primeras 4 columnas de cada matriz con la posición de cada

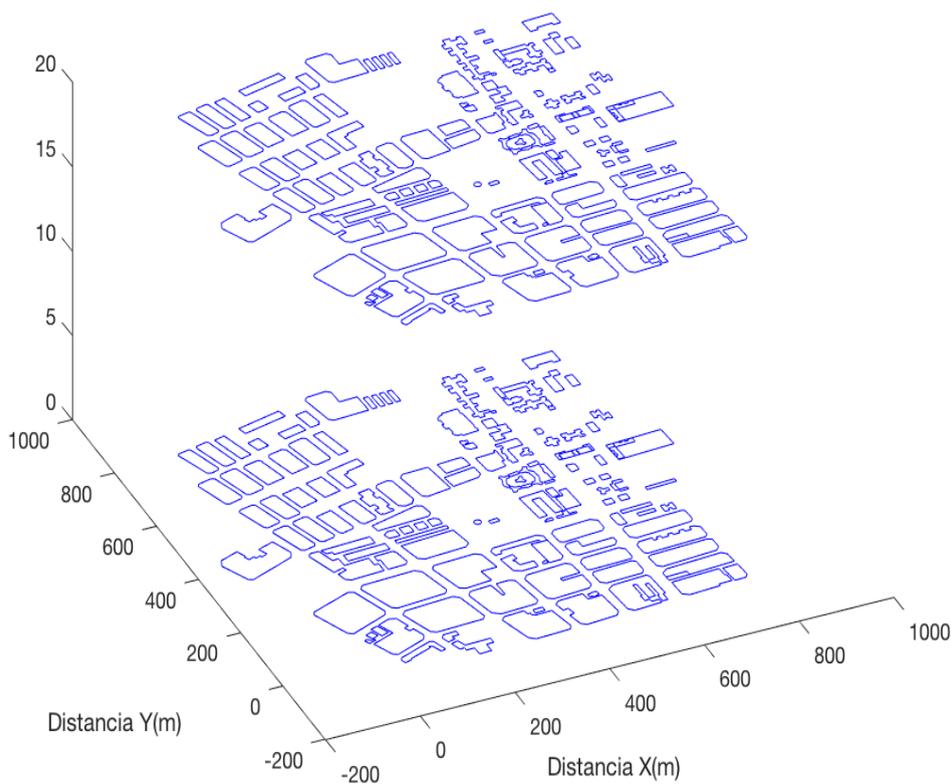


Figura 3.17: Representación de la parte inferior y superior de los edificios del entorno

vértice, así como, la primera fila de la quinta columna que tomará el valor 1, correspondiente a pared vertical. Finalmente, se modifica también la tercera fila y sexta columna con el número del edificio al que pertenece la superficie.

La Figura 3.18 muestra como quedaría representado el entorno unicamente con paredes verticales. El entorno del ejemplo cuenta con 133 edificios formados por 1275 paredes verticales.

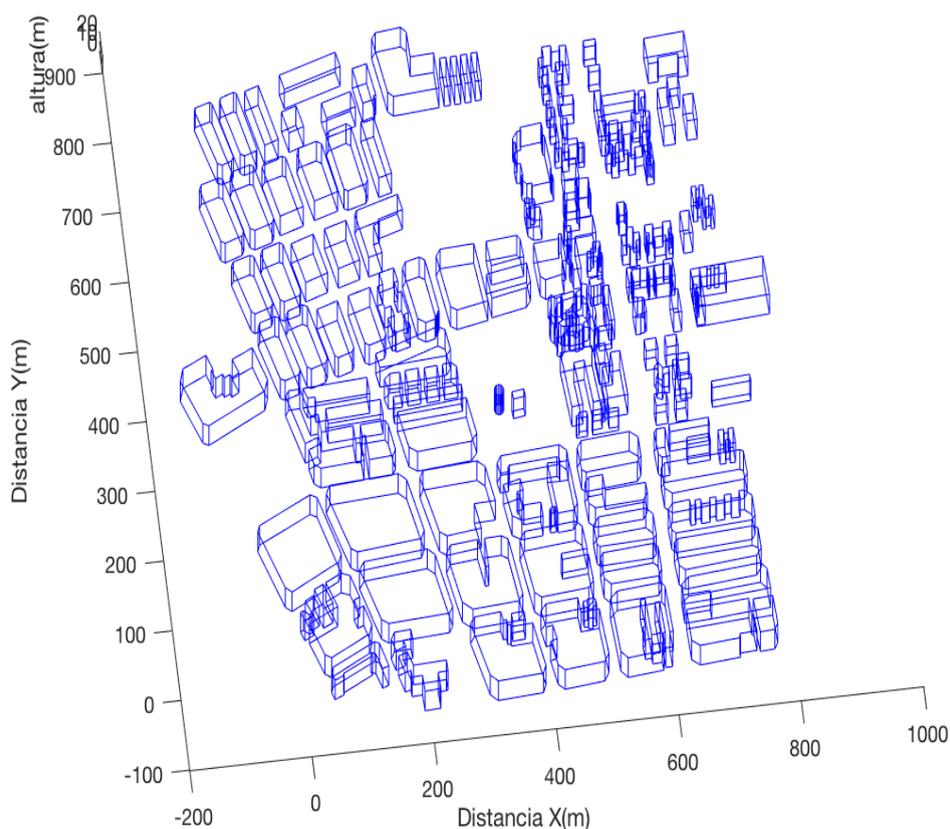


Figura 3.18: Representación del entono 2D a la altura deseada

3.4.6. Implementación de las superficies horizontales

La implementación de superficies horizontales no es tan evidente como en el caso anterior, debido a la complejidad existente de simplificar cualquier forma geométrica mediante triángulos y cuadriláteros de manera automática.

Para abordar este problema, la aplicación hará uso de los vértices que forman el techo del edificio. Además, tenemos la posibilidad de evaluar cada edificio de forma aislada al resto, ya que tenemos en *planes* los vértices que pertenecen a cada edificio.

El archivo OSM ya nos devolvía las uniones de los vértices que forman la superficie

de manera ordenada. Por ejemplo, si la superficie esta formada por cinco vértices, cuyas uniones deben ser vértice 1 con vértice 2, vértice 2 con vértice 3, vértice 3 con vértice 4, etc., ya tenemos disponible este orden, gracias a la disposición de los datos del archivo OSM.

Teniendo esta disposición de los vértices, es posible generar rectas formadas por dos vértices consecutivos. Si realizamos estas rectas con todas las parejas de vértices y, posteriormente, nos quedamos con las intersecciones que se encuentran dentro del polígono, podremos utilizar estos puntos intermedios para llevar a cabo la simplificación de la forma geométrica aleatoria.

Para obtener la intersección entre las rectas se hace uso de la función *lines_exp_int_2D* perteneciente a un paquete de funciones de MATLAB que permiten obtener figuras y datos genéricos. Esta función toma como entrada la posición de los dos vértices que forman la primera recta y la posición de los dos vértices que forman la segunda recta. Como salida, obtenemos una variable que nos indica si las rectas no se intersectan, si tienen un punto de intersección o si existen infinitas intersecciones (líneas idénticas). En caso de que exista un único punto, también devolverá dicho punto.

Estas rectas formarán intersecciones tanto en el interior de la superficie, como fuera de ella. Para nuestra implementación, solamente se hará uso de las intersecciones que estén contenidas en la superficie. La función *inpolygon* permite seleccionar las intersecciones interiores, de tal forma que descartamos el resto de intersecciones que no son de interés. *inpolygon* tomará como entrada las coordenadas de la intersección que deseamos evaluar, así como, un array con las posiciones de los vértices que forman el polígono. La salida será un 1 si la intersección se encuentra contenida en el polígono o en el borde de este. En caso contrario, la salida será un 0 y se descartará esta intersección.

Existe la posibilidad de que existan intersecciones que se repitan, luego se debe comprobar que el vector que puntos que contiene las intersecciones interiores no contiene

elementos repetidos.

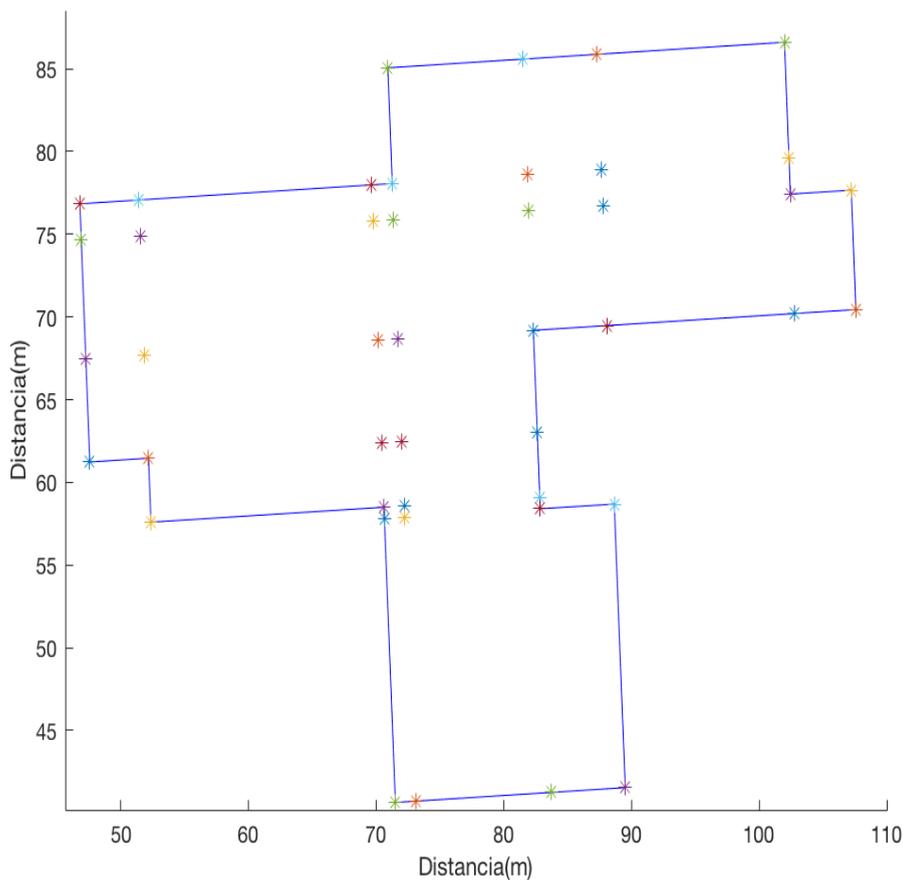


Figura 3.19: Intersecciones creadas a partir de las proyecciones de rectas

La Figura 3.19 muestra un ejemplo de superficie compleja con una forma aleatoria pero perteneciente a un edificio real, a la que se le han aplicado las funciones expuestas anteriormente.

A partir de las intersecciones obtenidas y de los vértices que delimitan la superficie, podemos aplicar triangulación para obtener una simplificación mediante triángulos lo más óptima posible. Para ello, utilizaremos la función *delaunayTriangulation*. Esta función utiliza como entrada el conjunto de puntos interiores y las coordenadas de los

vértices que delimitan la superficie horizontal. La salida será un objeto que contiene tres atributos:

- El atributo ***Constraints*** se encuentra representado por una matriz de 2 columnas con los identificadores de los vértices. Cada fila corresponde a la unión de dos vértices, que establecen las rectas limitantes de la superficie.
- El atributo ***Points*** contiene las coordenadas de los puntos. Cada fila de *Points* contiene las coordenadas de un punto. Además, el número de la fila hará de identificador de ese punto.
- El atributo ***ConnectivityList*** esta formado por una matriz de tres columnas con los identificadores asignados a los diferentes puntos. Cada fila de esta matriz representa un triángulo en la triangulación. De la misma forma que en el atributo anterior, cada número de fila actúa como identificador de dicho triángulo.

La Figura 3.20 muestra el resultado de aplicar la triangulación a la representación del entorno de la Figura 3.16. Es interesante superponer ambos entornos en la representación para observar cual ha sido la tasa de éxito al aplicar esta técnica.

A partir de la Figura 3.21 es fácil observar como se ha conseguido simplificar, haciendo uso de triángulos, todas las formas geométricas correspondientes a los diferentes edificios, excepto la de uno. A pesar de este caso aislado, consideramos que es una simplificación óptima y suficiente para añadir a nuestra herramienta de trazado de rayos debido a la gran tasa de éxito.

Para el usuario final no será necesario seguir todos los apartados de esta sección. Se ha implementado una función llamada *genera_techo_edificios* a la que introduciendole el *planes* generado al añadir las paredes verticales ya devolverá un nuevo *planes* con las superficies verticales y horizontales en el orden correcto. Con el objetivo de minimizar el número de estructuras, esta función también representará los edificios formador por

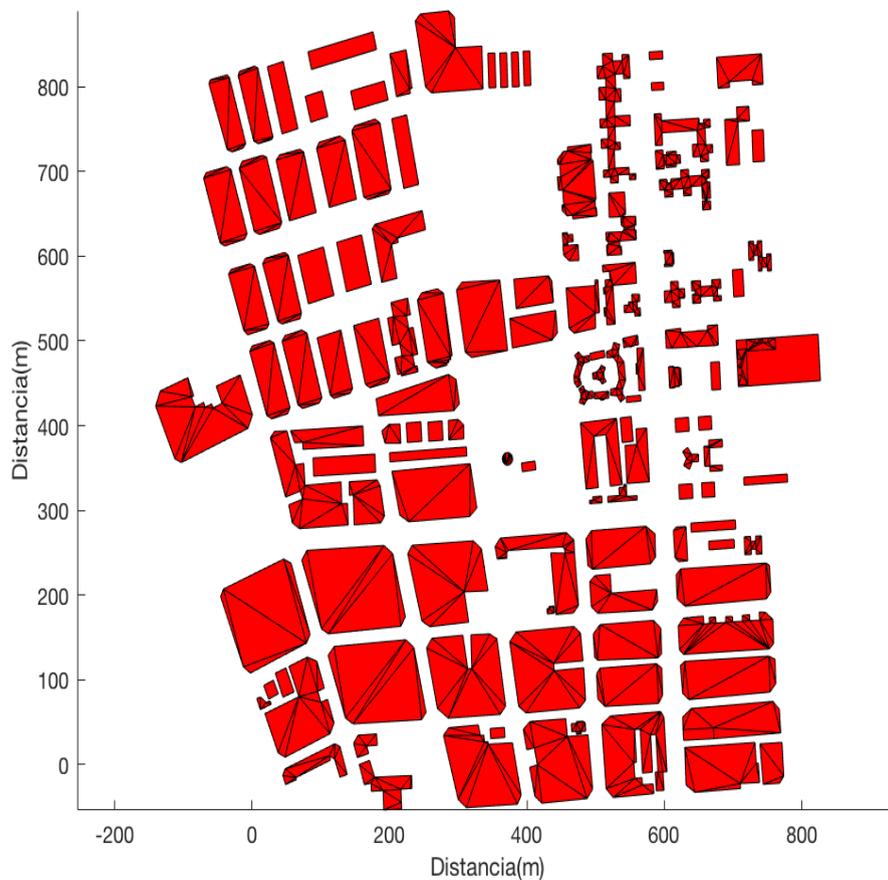


Figura 3.20: Superficies horizontales a partir de triangulación

una superficie horizontal definida por cuatro vértices como un cuadrilátero, en lugar de aplicar triangulación, devolviendo varios triángulos.

3.4.7. Propiedades eléctricas de los materiales que forman las superficies

Como se mostró en la Figura 3.7, es necesario introducir en *planes* las propiedades eléctricas del material que forma la superficie establecidas por la permitividad relativa, la conductividad de la pared en Siemens por metro (S/m), la atenuación que sufre un rayo en transmisión al atravesar la pared y si el material es un conductor perfecto o

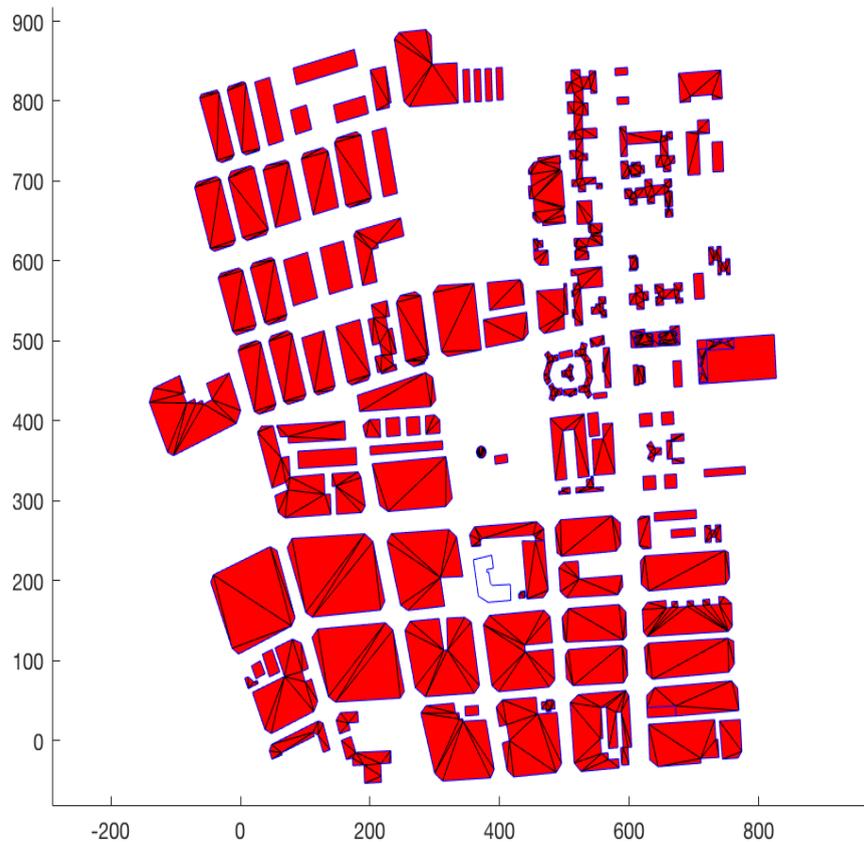


Figura 3.21: Comparación de entornos

no.

Según la ITU en [16], los datos sobre las propiedades del material eléctrico pueden ser difíciles de determinar, puesto que las características se expresan haciendo uso de diferentes combinaciones de parámetros y puede que los valores indicados de la permitividad relativa sean para frecuencias alejadas de las frecuencias de interés. Por tanto se ha elaborado un cuadro de las propiedades de los materiales recogidas en la Figura 3.22.

Para llevar a cabo este formato normalizado se utilizaron los datos procedentes de ocho conjuntos de propiedades eléctricas del material que figuran en los texto técnicos.

Para cada grupo, se obtuvieron expresiones sencillas de los valores dependiendo de la frecuencia de la parte real de la permitividad relativa, η' , y la conductividad, σ , donde:

$$\eta' = af^b \quad (3.1)$$

y:

$$\sigma = cf^d \quad (3.2)$$

siendo f la frecuencia en GHz, σ expresada en $\frac{S}{m}$ y η' adimensional. Los valores de a , b , c y d se encuentran disponibles en la Figura 3.22. Cuando el valor de b o d es cero, el correspondiente valor de η' o σ es a o c respectivamente, es decir, es independiente de la frecuencia.

Para el valor de la parte imaginaria de la permitividad relativa η'' puede calcularse a partir de la conductividad y la frecuencia:

$$\eta'' = 17,98 \frac{\sigma}{f} \quad (3.3)$$

La Tabla 3.1 muestra unos límites de frecuencia que no son estrictos, sino más bien indicativos de los datos medidos que se utilizaron para obtener los modelos. Aún así, el rango de frecuencias se encuentra dentro del rango de frecuencias utilizadas en comunicaciones móviles, por lo que estos modelos son perfectos para utilizarlos en nuestra herramienta.

Para asignar la atenuación de cada material a la frecuencia deseada se hará uso de los datos recogidos en [4]. La Tabla 3.2 muestra la pérdida de señal media captada por varios investigadores al obstruir la señal por cada uno de los materiales indicados.

Propiedades del material

Tipo de material	Parte real de la permitividad relativa		Conductividad S/m		Gama de frecuencias
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	
					GHz
Vacío (\approx aire)	1	0	0	0	0,001-100
Hormigón	5,31	0	0,0326	0,8095	1-100
Ladrillo	3,75	0	0,038	0	1-10
Placa de yeso	2,94	0	0,0116	0,7076	1-100
Madera	1,99	0	0,0047	1,0718	0,001-100
Vidrio	6,27	0	0,0043	1,1925	0,1-100
Revestimiento de techos	1,50	0	0,0005	1,1634	1-100
Aglomerado	2,58	0	0,0217	0,7800	1-100
Revestimiento de suelos	3,66	0	0,0044	1,3515	50-100
Metal	1	0	10^7	0	1-100
Suelo muy seco	3	0	0,00015	2,52	1-10 sólo
Suelo medianamente seco	15	-0,1	0,035	1,63	1-10 sólo
Suelo húmedo	30	-0,4	0,15	1,30	1-10 sólo

Tabla 3.1: Propiedades eléctricas de los materiales [16]

Para el usuario final, se ha implementado una función denominada *material_properties* que recoge las propiedades correspondientes a las superficies formadas por los materiales: hormigón, ladrillo, yeso, madera, metal y cristal. Si fuese necesario obtener las propiedades eléctricas de cualquier otro material, el usuario solamente debería introducir los datos de las Tablas 3.1 y 3.2.

La función *material_properties* toma como entradas el nombre del material como un *string* y la frecuencia a la que se va llevar la simulación. Como salida devuelve la permitividad del material, tanto su parte real como su parte imaginaria, la conductividad del material en $\frac{S}{m}$, la atenuación que se produce en transmisión al atravesar un

Material Type	Loss (dB)	Frequency	Reference
All metal	26	815 MHz	[Cox83b]
Aluminium siding	20.4	815 MHz	[Cox83b]
Foil insulation	3.9	815 MHz	[Cox83b]
Concrete block wall	13	1300 MHz	[Rap91c]
Loss from one floor	20-30	1300 MHz	[Rap91c]
Loss from one floor and one wall	40-50	1300 MHz	[Rap91c]
Fade observed when transmitter turned a right angle corner in a corridor	10-15	1300 MHz	[Rap91c]
Light textile inventory	3-5	1300 MHz	[Rap91c]
Chain-like fenced in area 20 ft high containing tools, inventory, and people	5-12	1300 MHz	[Rap91c]
Metal blanket — 12 sq ft	4-7	1300 MHz	[Rap91c]
Metallic hoppers which hold scrap metal for recycling - 10 sq ft	3-6	1300 MHz	[Rap91c]
Small metal pole — 6" diameter	3	1300 MHz	[Rap91c]
Metal pulley system used to hoist metal inventory — 4 sq ft	6	1300 MHz	[Rap91c]
Light machinery < 10 sq ft	1-4	1300 MHz	[Rap91c]
General machinery - 10 - 20 sq ft	5-10	1300 MHz	[Rap91c]
Heavy machinery > 20 sq ft	10-12	1300 MHz	[Rap91c]
Metal catwalk/stairs	5	1300 MHz	[Rap91c]
Light textile	3-5	1300 MHz	[Rap91c]
Heavy textile inventory	8-11	1300 MHz	[Rap91c]
Area where workers inspect metal finished products for defects	3-12	1300 MHz	[Rap91c]
Metallic inventory	4-7	1300 MHz	[Rap91c]
Large I-beam — 16 - 20"	8-10	1300 MHz	[Rap91c]
Metallic inventory racks — 8 sq ft	4-9	1300 MHz	[Rap91c]
Empty cardboard inventory boxes	3-6	1300 MHz	[Rap91c]
Concrete block wall	13-20	1300 MHz	[Rap91c]
Ceiling duct	1-8	1300 MHz	[Rap91c]
2.5 m storage rack with small metal parts (loosely packed)	4-6	1300 MHz	[Rap91c]
4 m metal box storage	10-12	1300 MHz	[Rap91c]
5 m storage rack with paper products (loosely packed)	2-4	1300 MHz	[Rap91c]

Material Type	Loss (dB)	Frequency	Reference
5 m storage rack with large paper products (tightly packed)	6	1300 MHz	[Rap91c]
5 m storage rack with large metal parts (tightly packed)	20	1300 MHz	[Rap91c]
Typical N/C machine	8-10	1300 MHz	[Rap91c]
Semi-automated assembly line	5-7	1300 MHz	[Rap91c]
0.6 m square reinforced concrete pillar	12-14	1300 MHz	[Rap91c]
Stainless steel piping for cook-cool process	15	1300 MHz	[Rap91c]
Concrete wall	8-15	1300 MHz	[Rap91c]
Concrete floor	10	1300 MHz	[Rap91c]
Commercial absorber	38	9.6 GHz	[Vio88]
Commercial absorber	51	28.8 GHz	[Vio88]
Commercial absorber	59	57.6 GHz	[Vio88]
Sheetrock (3/8 in) — 2 sheets	2	9.6 GHz	[Vio88]
Sheetrock (3/8 in) — 2 sheets	2	28.8 GHz	[Vio88]
Sheetrock (3/8 in) — 2 sheets	5	57.6 GHz	[Vio88]
Dry plywood (3/4 in) — 1 sheet	1	9.6 GHz	[Vio88]
Dry plywood (3/4 in) — 1 sheet	4	28.8 GHz	[Vio88]
Dry plywood (3/4 in) — 1 sheet	8	57.6 GHz	[Vio88]
Dry plywood (3/4 in) — 2 sheets	4	9.6 GHz	[Vio88]
Dry plywood (3/4 in) — 2 sheets	6	28.8 GHz	[Vio88]
Dry plywood (3/4 in) — 2 sheets	14	57.6 GHz	[Vio88]
Wet plywood (3/4 in) — 1 sheet	19	9.6 GHz	[Vio88]
Wet plywood (3/4 in) — 1 sheet	32	28.8 GHz	[Vio88]
Wet plywood (3/4 in) — 1 sheet	59	57.6 GHz	[Vio88]
Wet plywood (3/4 in) — 2 sheets	39	9.6 GHz	[Vio88]
Wet plywood (3/4 in) — 2 sheets	46	28.8 GHz	[Vio88]
Wet plywood (3/4 in) — 2 sheets	57	57.6 GHz	[Vio88]
Aluminium (1/8 in) — 1 sheet	47	9.6 GHz	[Vio88]
Aluminium (1/8 in) — 1 sheet	46	28.8 GHz	[Vio88]
Aluminium (1/8 in) — 1 sheet	53	57.6 GHz	[Vio88]

Tabla 3.2: Atenuaciones de diferentes materiales [4]

rayo la superficie y si el material es o no un conductor perfecto.

Capítulo 4

Resultados obtenidos a partir de medidas reales

4.1. Presentación del entorno a simular

Para comprobar el funcionamiento de nuestra herramienta se va a llevar cabo la caracterización del canal radio para GSM 1800 en un entorno urbano real. Para ello, haremos uso de una serie de medidas que se realizaron en Cartagena [17] y posteriormente serán comparadas con las obtenidas por el trazador de rayos cuando utiliza el entorno generado por nuestra herramienta.

Estas medidas fueron tomadas en el año 2006 alrededor de la estación base de Movistar situada en la Calle Juan Fernández. La posición concreta de esta estación base en coordenadas UTM encuentra en la Figura 4.1.

Emplazamiento	Coordenada x (m)	Coordenada y (m)
Juan Fernández	677606	4165202

Figura 4.1: Coordenadas de la estación base en la Calle Juan Fernández

Una vez conocida la posición del transmisor, debemos localizar su posición en OpenStreetMap y exportar el entorno a analizar. Los pasos a seguir para generar el

entorno urbano se muestran en el diagrama de la Figura 4.2 y estará contenido en la variable *planes*.

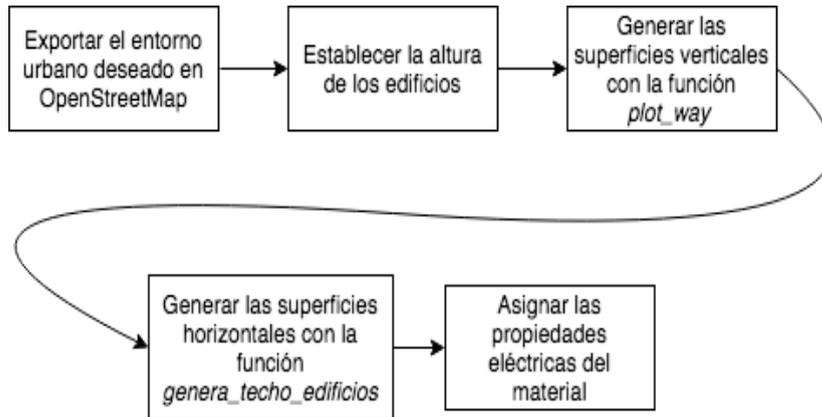


Figura 4.2: Coordenadas de la estación base en la Calle Juan Fernández

La Figura 4.3 muestra el entorno generado por la aplicación y almacenado en *planes*. En esta representación también se muestra la posición de la estación base en la Calle Juan Fernández que actuará como transmisor. En cuanto a la altura de los edificios, se ha seleccionado 20 metros para todos los edificios, al tratarse de la altura aproximada de la mayoría de los edificios de la zona de medida[17], encontrándose la estación base a 25 metros sobre el suelo.

Las medidas reales se llevaron a cabo en los alrededores de esta estación base, donde el recorrido queda recogido en la Figura 4.4. A partir de esta figura y los datos ofrecidos en [17] observamos que se realizaron una gran cantidad de medidas, en torno a 900 posiciones de recepción. Debido a que una simulación con tantos puntos de recepción y un entorno urbano con tantos elementos tiene un coste computacional muy elevado, en nuestro caso seleccionaremos una cuarta parte de los puntos. De esta forma, obtendremos un recorrido similar, pero con menos puntos a analizar, simplificando la simulación. La figura 4.5 muestra el entorno urbano con la posición del transmisor y los diferentes puntos de recepción seleccionados.

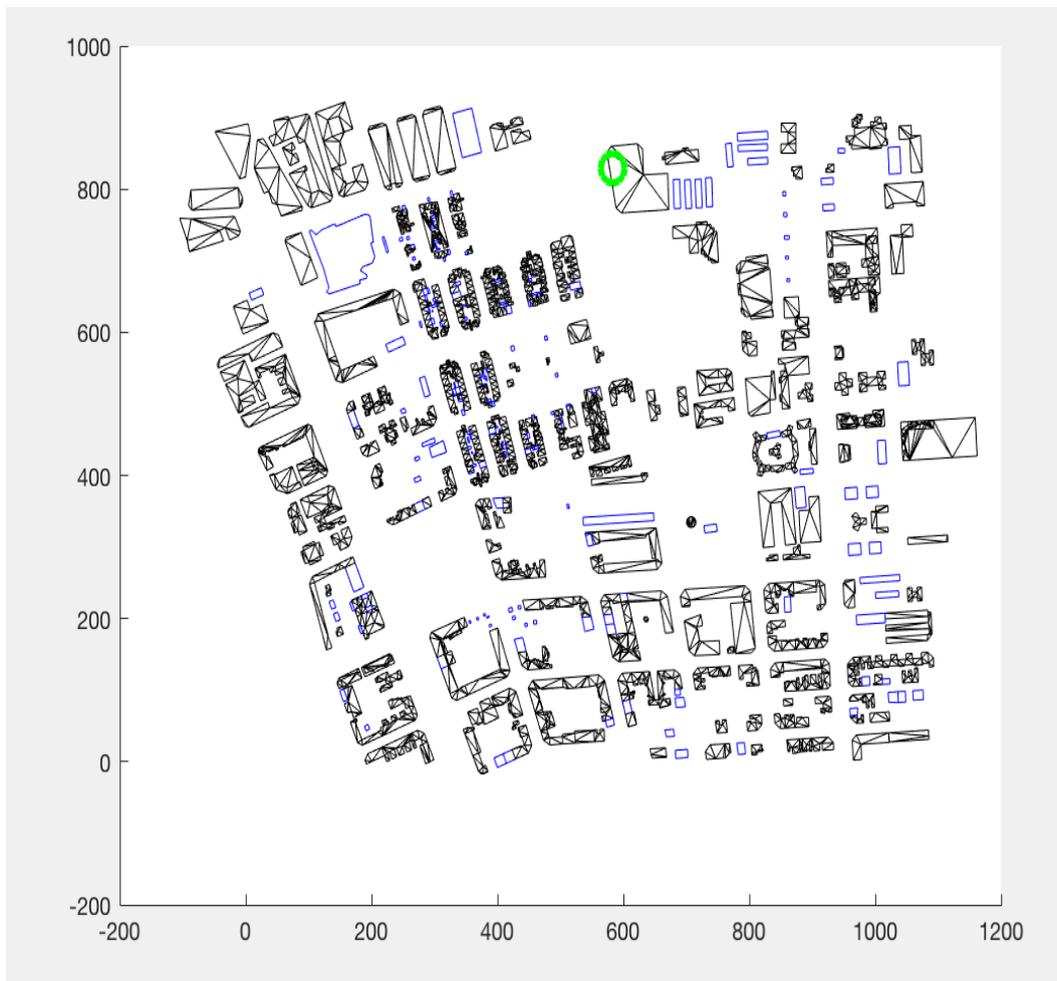


Figura 4.3: Elementos del entorno urbano a simular. El círculo verde indica la estación base (transmisor)

Con estos datos ya es posible llevar a cabo la simulación haciendo uso del trazador de rayos, comparando la potencia que esta herramienta nos devuelve con las medidas reales.

4.2. Aplicación de la herramienta de trazado de rayos al entorno generado

La herramienta de trazado de rayos es capaz de calcular la potencia en cada uno de los puntos de recepción, a partir de las contribuciones del rayo directo, los rayos

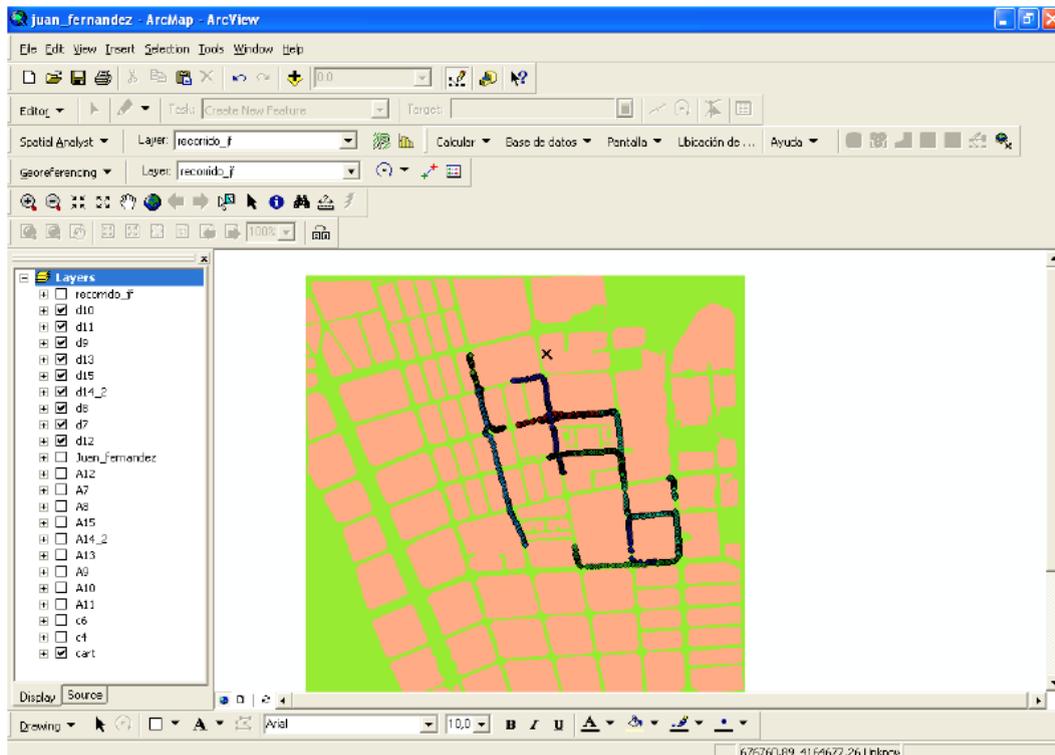


Figura 4.4: Entorno urbano con las posiciones del receptor[17]

reflejados y los rayos difractados. La frecuencia a la que se llevará a cabo la simulación es de 1800 MHz y únicamente se tendrá en cuenta una reflexión de cada rayo. Sería posible establecer un número mayor de reflexiones, pero de nuevo aumentaría tanto el coste computacional como el coste temporal.

El primer paso consiste en establecer el rayo directo a partir de la posición del transmisor, las posiciones del receptor y *planes*. La función encargada de esto es *genera_rayo_directo*. A partir del rayo generado, podemos extraer todos los datos de dicho rayo haciendo uso de la función *extrae_datos_rayos* como son: el campo eléctrico, la distancia que recorre el rayo, el valor del ángulo *theta* y *phi* tanto en transmisión como en recepción, etc.

El segundo paso será generar los rayos reflejados. La Tabla 4.1 contiene los parámetros de entrada que utiliza la función *genera_rayos_paralelo*, encargada de generar los

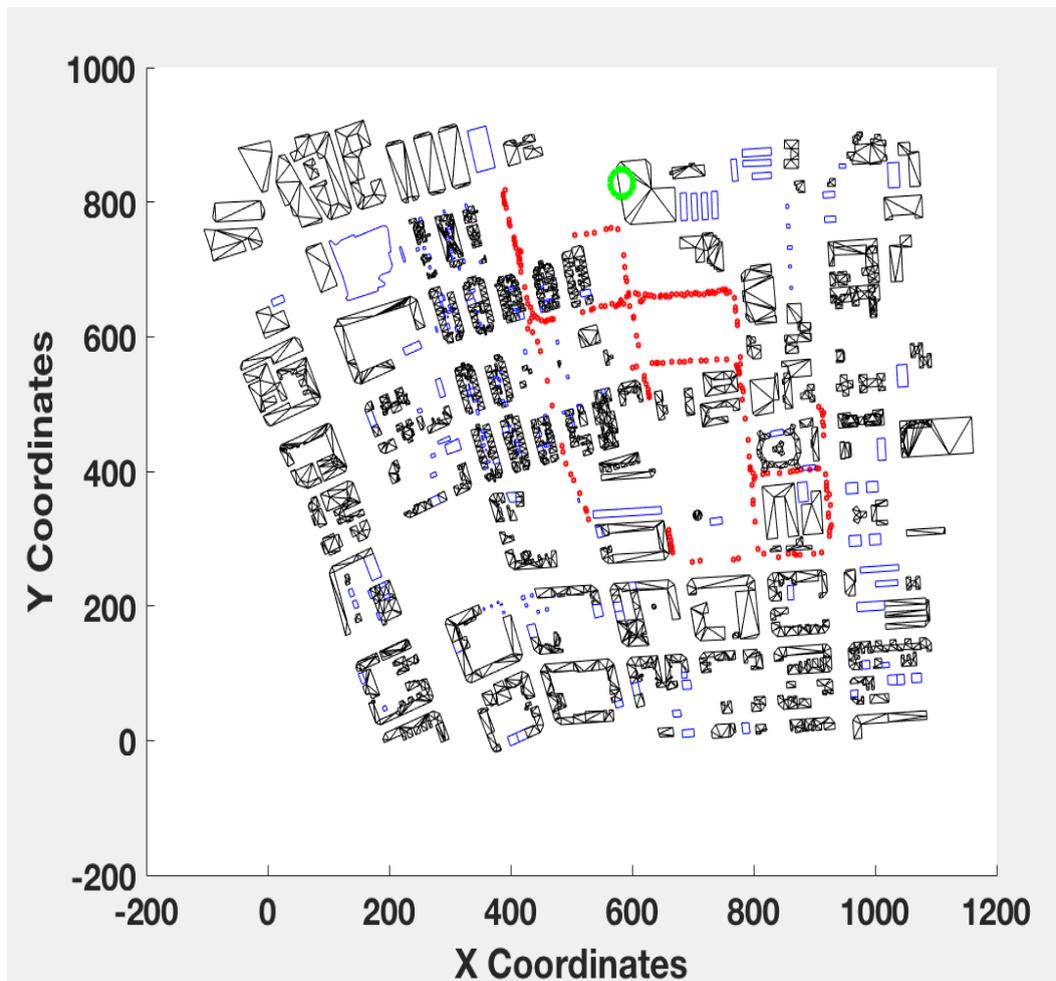


Figura 4.5: Entorno urbano con el recorrido donde se tomaron las diferentes medidas

rayos reflejados.

- Para obtener las *imagenes* haremos uso de la función *genera_imagenes*, a partir de *tx*, *rx*, *planes* y el número de reflexiones.
- El número de reflexiones almacenado en *num_reflex* será igual a uno, es decir, únicamente se tendrán en cuenta los rayos que llegan al receptor sufriendo una única reflexión. No se han incluido dos reflexiones debido a que el tiempo de computación era demasiado elevado.
- Las variables *planes*, *tx* y *rx* son las expuestas anteriormente.

- El valor de *frec_media* es de 1800 MHz, correspondiente a la frecuencia utilizada en GSM 1800.
- Se ha considerado que la polarización de la onda es vertical tanto en transmisión como en recepción.
- El número de transmisiones que puede realizar el rayo se ha establecido en 5.
- El número de datos para cada transmisión se ha establecido en 4.
- La atenuación límite a partir de la cual no se considerará la contribución de ese rayo será de 200 dB.
- *lineas_refl* sirve para evitar considerar los impactos en las aristas como reflexiones validas.

Una vez tenemos los diferentes rayos reflejados podemos obtener su información, al igual que se hacía con el rayo directo, con la función *extrae_datos_rayos*.

La última contribución será la que proviene de los rayos difractados generados por la función de *genera_rayos_difracción* cuya descripción de los parámetros que toma como entrada se describen en la Tabla 4.2. Una vez generados los rayos difractados se extrae la información de cada rayo con la función *extrae_datos_rayos*.

Ahora que contamos con las contribuciones de los diferentes rayos que llegan al receptor el último paso consiste en sumar todas estas contribuciones y obtendremos el campo eléctrico total en ese punto.

Para realizar el cálculo de potencia se ha asumido que la antena transmisora es omnidireccional. Este supuesto es posible debido a que como los puntos de recepción se encuentran alejados de esta, la variación del ángulo será mínima otorgándonos una ganancia muy similar. La ganancia de dicha antena es de 14 dBi según su certificación. La ganancia de la antena receptora es de 2 dBi [17]. El valor de la PIRE es desconocido

<i>genera_rayos_paralelo</i>	
Parámetros de entrada	Descripción
<i>imagenes</i>	Imágenes generadas a partir de planes, posición del transmisor y receptor, y el número de reflexiones
<i>num_reflex</i>	Establece el número de reflexiones que sufrirá el rayo al sufrir el fenómeno de la reflexión
<i>planes</i>	Contiene el entorno urbano a simular
<i>tx</i>	Posición en la que se encuentra el transmisor
<i>rx</i>	Posiciones de los diferentes receptores
<i>frec_media</i>	Frecuencia media a la que se realiza la simulación
<i>polarizacion</i>	Polarización de la onda en la antena transmisora y receptora
<i>num_transmisiones</i>	Número de transmisiones que puede atravesar el rayo reflejado
<i>num_datos_transmisiones</i>	Indica el número de datos de cada rayo reflejado
<i>atenuacion_limite</i>	Establece la atenuación máxima que puede sufrir un rayo reflejado antes de ser descartado
<i>lineas_refl</i>	Generado previamente mediante la función <i>calcula_lineas</i> , contiene las líneas que forman las aristas entre los elementos rectangulares y triangulares

Tabla 4.1: Parámetros de entrada de la función *genera_rayos_paralelo*

por lo que tomaremos un valor de 10 dBm para ajustar las simulaciones con las medidas ya que la diferencia era constante a lo largo de todo el recorrido. La nube de puntos de la Figura 4.6 muestra la potencia en dBm que hemos obtenido en la simulación en función de la distancia en metros a la que se encuentra el receptor.

Para concluir este punto es de interés comentar algunos datos en cuanto al coste computacional de realizar esta simulación. El entorno en el que se ha llevado a cabo cuenta con 9789 superficies que modelan un total de 624 edificios. Se han tomado 225 puntos de recepción repartidos por todo el entorno. El tiempo total de la simulación ha sido de 56 horas, lo que supone una media de aproximadamente 15 minutos por punto de recepción. Este tiempo de media por punto se ha medido una vez se han generado

<i>genera_rayos_difraccion</i>	
Parámetros de entrada	Descripción
<i>planes</i>	Contiene el entorno urbano a simular
<i>punto_origen</i>	Posición en la que se encuentra el transmisor
<i>punto_destino</i>	Posiciones de los diferentes receptores
<i>frec_media</i>	Frecuencia media a la que se realiza la simulación
<i>polarizacion</i>	Polarización de la onda en la antena transmisora y receptora
<i>vector_pol_origen</i>	Establece el vector de polarización a partir de la polarización de la onda en la antena transmisora y receptora
<i>aristas</i>	Contiene las aristas generadas por las uniones entre las superficies rectangulares y las superficies triangulares

Tabla 4.2: Parámetros de entrada de la función *genera_rayos_difraccion*

las aristas del entorno, lo que ha supuesto unas 4 horas de forma aproximada. El ordenador utilizado contaba con un procesador Intel Core i5 4690k a una frecuencia de 3.5 GHz y una memoria RAM de 8 Gb a una frecuencia de 1600 MHz. La Tabla 4.3 recoge el tiempo medio aproximado de computo por punto de recepción en minutos para tres casos: únicamente teniendo en cuenta una reflexión y sin difracción, con una reflexión y con difracción y con dos reflexiones y sin difracción. Cabe destacar que el realizar la simulación únicamente con una reflexión y sin difracción nos proporciona unos resultados aceptables para un tiempo de cálculo relativamente bajo pero no llega a ser tan preciso como cuando consideramos difracción.

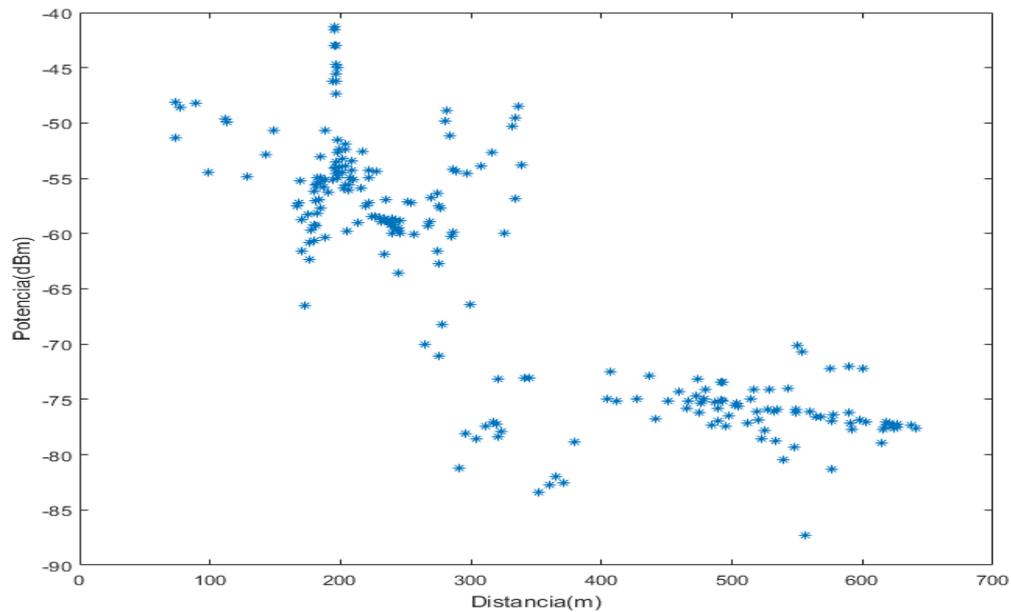


Figura 4.6: Potencia recibida en la simulación en función de la distancia

Una reflexión Sin difracción	Una reflexión Con difracción	Dos reflexiones Sin difracción
1	15	75

Tabla 4.3: Tiempo aproximado de computo por posición de recepción en minutos para diferentes casos

4.3. Comparación de los resultados obtenidos con las medidas reales

La nube de puntos de las medidas de potencia reales tomadas en los alrededores de la estación base en la Calle Juan Fernández se muestra en la Figura 4.6 en color rojo mientras que superpuestas con estas se muestran los resultados de la simulación en color azul.

Como se puede observar en la Figura 4.6, la nube de puntos presenta dos compor-

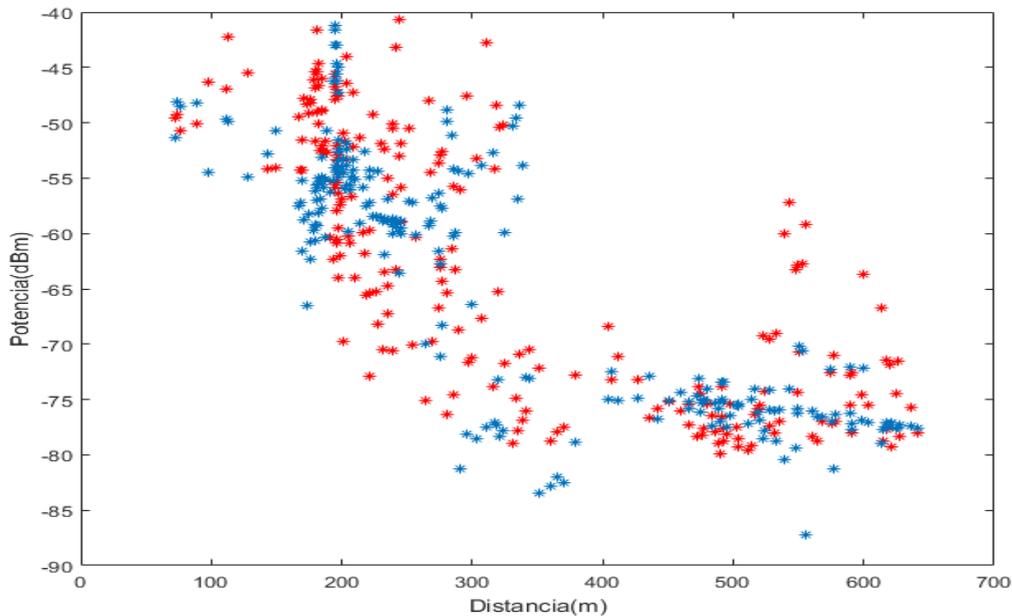


Figura 4.7: Comparación de las medidas reales con las medidas ofrecidas por la simulación

tamientos diferentes en función de la distancia entre el transmisor y el receptor. Para conseguir una mejor caracterización del canal radio, se ha realizado un estudio para las dos zonas por separado: la primera zona cercana a la estación base ($d < 300$ metros) y la segunda zona lejana respecto a esta ($d > 300$ metros). La representación de la potencia recibida (dBm) frente a la distancia (dB) y la recta de regresión correspondiente a las medidas más cercanas a la estación base se muestran en la Figura 4.8.

El resultado óptimo sería que ambas rectas de regresión fuesen paralelas entre sí. Como podemos observar no son exactamente paralelas pero si bastante cercanas a este hecho. Esta diferencia de inclinación puede ser debida a que en la realidad existen una gran cantidad de elementos que no podemos considerar en la simulación. La diferencia de altura entre rectas es debida a que posiblemente se ha tomado una PIRE diferente a la que proporciona la antena transmisora siendo mayor la PIRE del caso real al de la simulación. Los parámetros de los valores L_0 y n obtenidos tanto para la recta de la

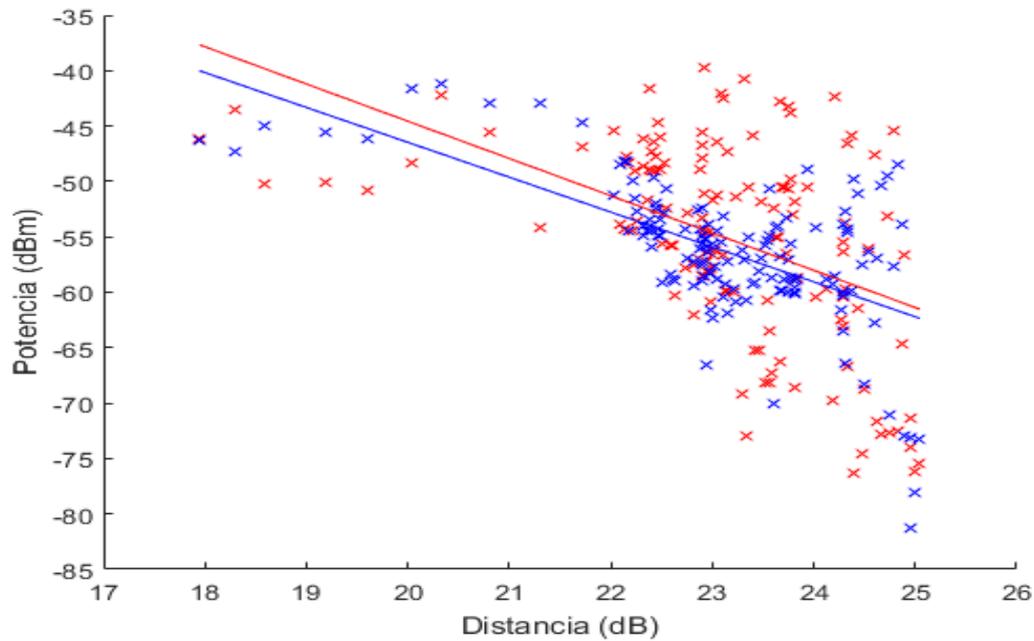


Figura 4.8: Comparación de la potencia recibida (dBm) en cada punto del recorrido de la zona cercana frente a la distancia al transmisor (dB)

simulación como la recta de las medidas reales se recogen en la Tabla 4.4.

L_0 medido	n medido	L_0 simulación	n simulación
22.6526	-3.3613	16.4678	-3.1472

Tabla 4.4: Valores de L_0 y n obtenidos

A partir de esta figura se puede observar como el valor de n correspondiente a la pendiente de la recta es muy similar para ambos casos, siendo negativo en ambos casos debido a que la potencia se reduce al aumentar la distancia. Este hecho es indicativo de que los resultados obtenidos en la simulación son bastante próximos a las medidas reales.

Por otro lado, el valor de L_0 correspondiente con el corte de la recta de regresión

y el eje de ordenadas de la Figura 4.8, no es el mismo para los dos casos. El principal motivo de esto se expuso anteriormente y podría ser debido a tomar una PIRE que no sería la de la estación base real.

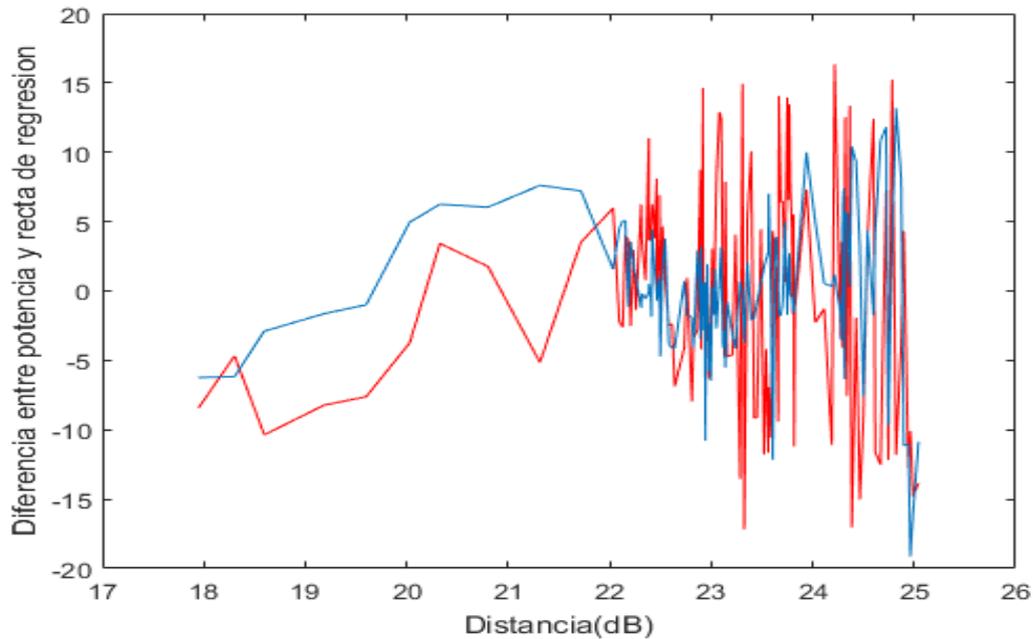


Figura 4.9: Diferencia entre la potencia recibida en cada punto y el valor de la recta de regresión en la zona cercana

En la Figura 4.9 se representa la diferencia entre el valor de la recta de regresión y la potencia captada en ese punto, tanto para las potencias reales (rojo) como para las potenciadas devueltas por la simulación (azul). A partir de esta figura se puede apreciar como a medida que la distancia aumenta entre el transmisor y el receptor, el error cometido al aproximar el valor de potencia por la recta de regresión aumenta para ambos casos.

La variabilidad de la propagación se traduce en variaciones con los emplazamientos y con el tiempo. La potencia recibida varía con la distancia según la ley exponencial [18]:

$$P_b = P_0 + 10n \log_{10} d + X_\sigma \quad (4.1)$$

donde:

- n depende del medio de propagación, de la frecuencia y de la altura de la antena que actúa como transmisor.
- d es la distancia entre el transmisor y el receptor.
- P_0 es la potencia a una distancia unitaria del transmisor.
- X_σ es una variable aleatoria de media cero y varianza σ que depende de la diversidad del entorno y varía en tiempo y en ubicación.

El histograma de X_σ queda recogido en las Figuras 4.10 (caso real) y 4.11 (simulación).

El test de Kolmogorov-Smirnov sirve para comparar si nuestro vector de muestras sigue una determinada distribución. La función con la que debemos comparar nuestras muestras es una función de distribución normal acumulativa de media cero y de desviación típica la obtenida para nuestras muestras. Este test devolverá un cero si como máximo un 5% de las muestras no cumple la hipótesis, o bien, un uno en caso contrario.

Al aplicar el test de Kolmogorov-Smirnov valido para verificar la normalidad de la distribución, en ambos casos se obtiene un cero. Este cero indica que la representación del histograma del error cometido al aproximar los valores por la recta de regresión corresponde a una distribución normal de media cero y varianza σ , lo que indica que en la mayoría de los puntos el error cometido es cercano a ceno. La Tabla 4.5 muestra los valores de varianza y desviación típica en ambos casos.

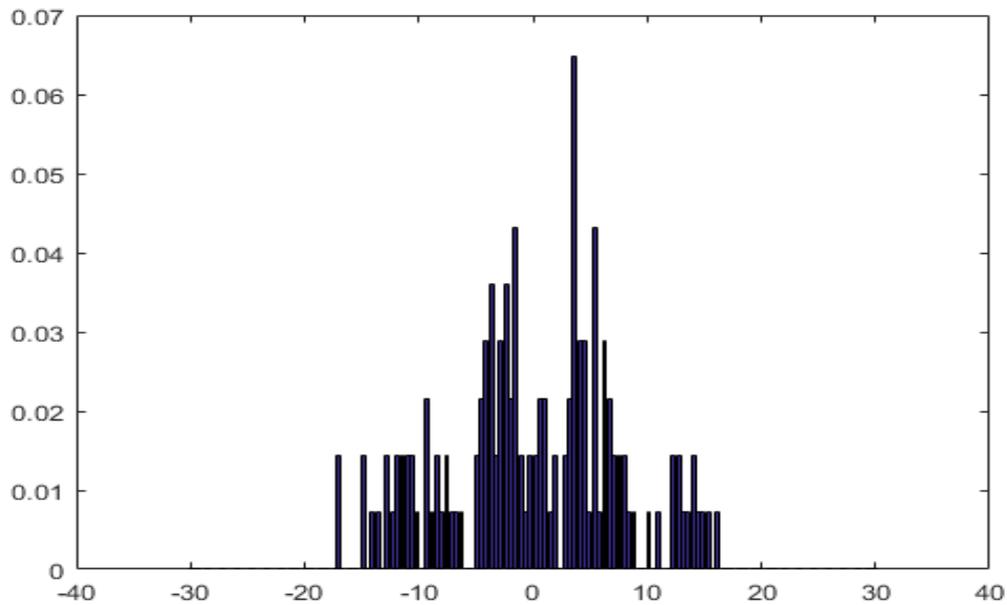


Figura 4.10: Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las medidas reales

Varianza c. medido	Desviación típica c. medido	Varianza c. simulación	Desviación típica c. simulación
58.14686	7.6268	24.9779	4.9978

Tabla 4.5: Varianza y desviación típica de la función de error en ambos casos

Para la zona del recorrido más alejado a la estación base de Juan Fernández se realizará el mismo análisis. Se considerarán los puntos de recepción cuya distancia al transmisor sea mayor de 300 metros. La representación de la potencia (dBm) frente a la distancia (dB) y la recta regresión tanto para las medidas tomadas en campo (rojo) como las medidas de potencia obtenidas de la simulación se muestran en la Figura 4.12.

En este caso los valores de potencia medidos son muy bajos, debido a la gran atenuación sufrida por el alto grado de edificación de la zona y el aumento de distancia entre transmisor y receptor. Como podemos observar el ajuste de las rectas no es tan

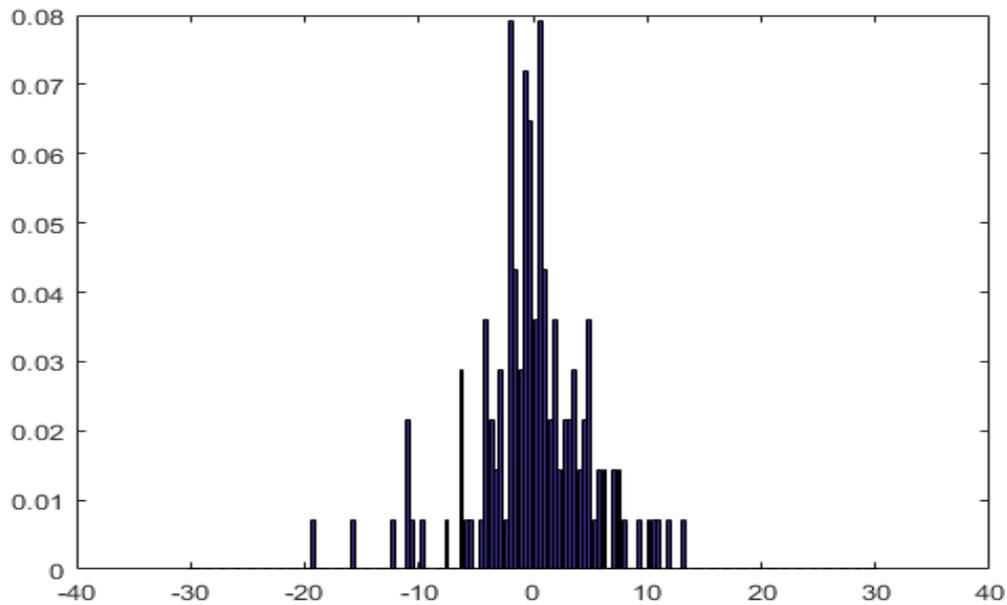


Figura 4.11: Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las soluciones otorgadas por la simulación

similar como lo era en el caso anterior. Esto puede ser debido a que al encontrarnos más alejados del transmisor, el número de elementos a tener en cuenta es mayor y no es posible considerarlos todos en la simulación a un coste computacional asequible. Los valores de L_0 y n se recogen en la Tabla 4.6.

L_0 medido	n medido	L_0 simulación	n simulación
-45.3021	-1.0826	-87.1388	0.3960

Tabla 4.6: Valores de L_0 y n obtenidos

El valor de n es el parámetro de interés en la zona lejana y, en ambos casos observamos como este se ha reducido, llegando incluso a ser positivo en el caso de los valores obtenidos en la simulación. Un valor de n positivo no tiene mucho sentido debido a

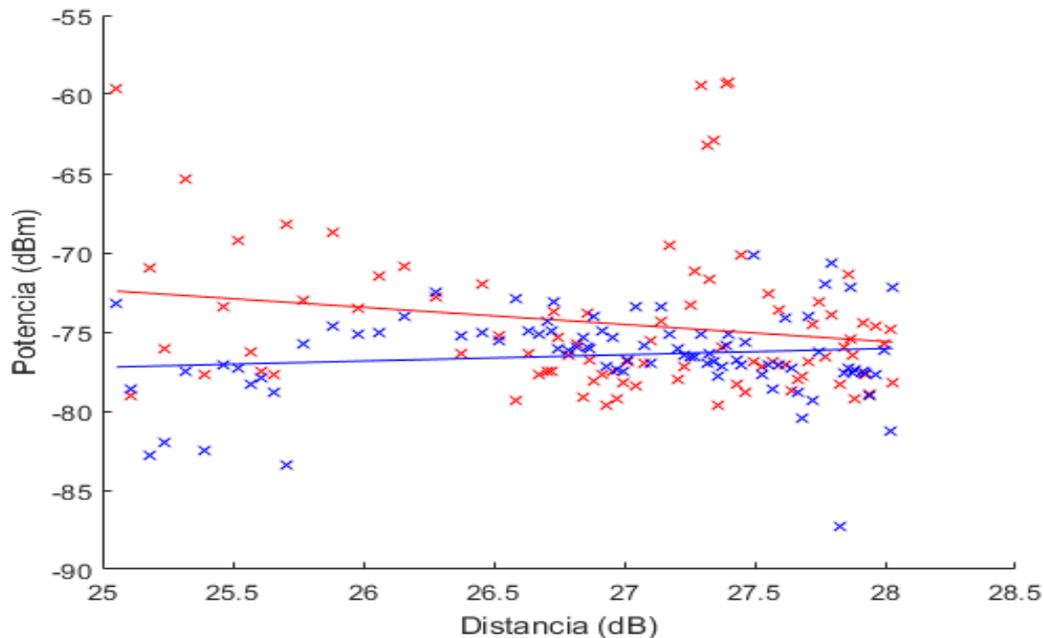


Figura 4.12: Comparación de la potencia recibida (dBm) en cada punto del recorrido de la zona lejana frente a la distancia al transmisor(dB)

que la potencia debería reducirse con la distancia. Este hecho puede justificarse debido a que ya nos encontramos en una potencia muy baja y a aleatoriedad del entorno. A parte de esto, en este rango de distancias unicamente se han analizado unas 90 posiciones de recepción, lo que no supone un elevado número de muestras para el rango de distancias tan grande que abarca.

La Figura 4.13 muestra la diferencia de potencia recibida en cada punto y el valor de la recta de regresión para ambos casos, en rojo para las medidas de potencia reales y en azul para la simulación. En esta gráfica se observan zonas con mayor variación del error en las medidas reales que en las simulaciones, probablemente causadas por las condiciones del entorno en el momento en el que se tomó la medida.

La Figura 4.14 y 4.15 muestran el histograma normalizado del error. En este caso al llevar a cabo el test de Kolmogorov-Smirnoc, el resultado que nos devuelve es 1, por lo que no se puede decir que la función de error corresponda a una distribución normal

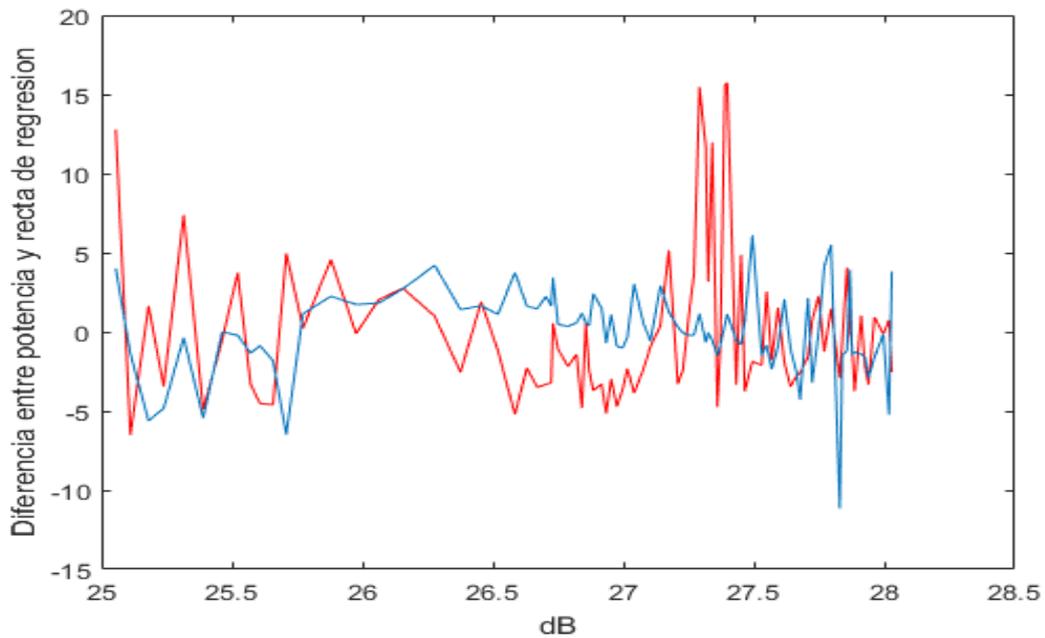


Figura 4.13: Diferencia entre la potencia recibida en cada punto y el valor de la recta de regresión en la zona lejana

de media 0 y desviación típica σ . No obstante, la forma del histograma tiene una gran semejanza con una densidad de probabilidad Gaussiana.

Varianza c.real	Desviación típica c.real	Varianza c.simulación	Desviación típica c.simulación
22.5347	4.7471	7.1918	2.6818

Tabla 4.7: Varianza y desviación típica de la función de error en ambos casos

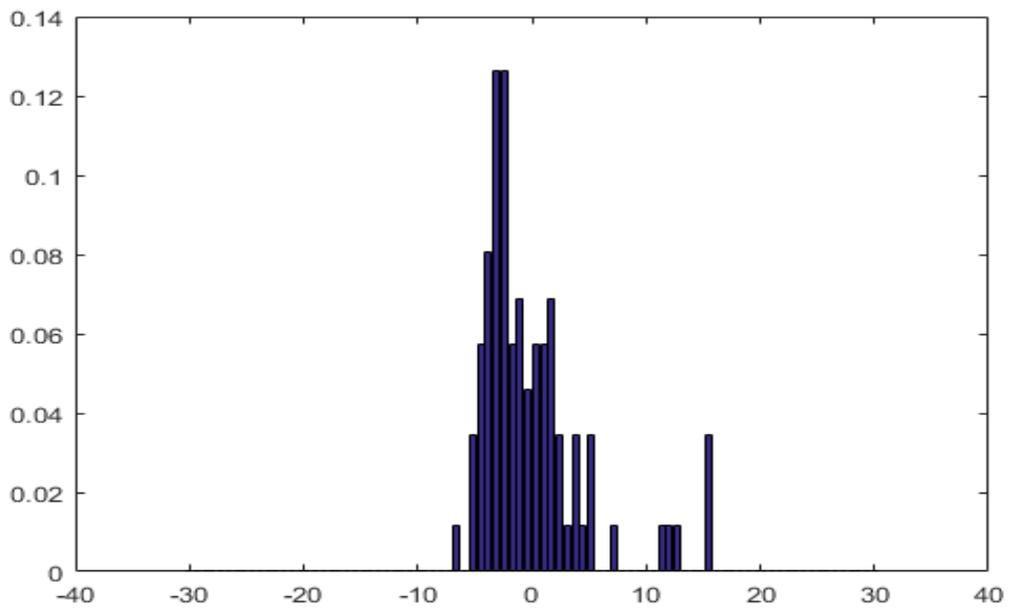


Figura 4.14: Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las medidas reales

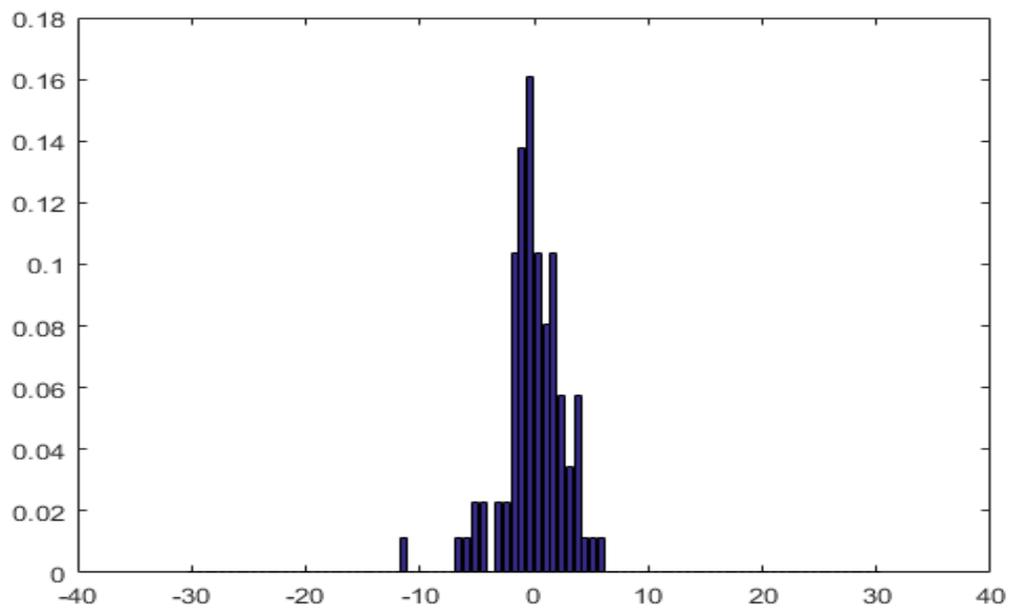


Figura 4.15: Histograma normalizado del error cometido al aproximar la potencia recibida en un punto por el valor de la recta de regresión en las soluciones otorgadas por la simulación

Capítulo 5

Conclusiones y futuras líneas de desarrollo

El objetivo principal de este Trabajo Final de Grado era el de desarrollar una aplicación que permitiese el modelado de cualquier tipo de entorno 2D exportado a través de la plataforma de OpenStreetMap, para posteriormente ser simplificado y conseguir generar un modelo 3D lo más fiel posible a la realidad. Este modelo debía adaptarse para permitir la compatibilidad con la herramienta de trazado de rayos desarrollada en MATLAB por el grupo de investigación de Sistemas de Comunicaciones Móviles de la UPCT.

En general, la herramienta ha conseguido modelar de forma correcta cualquier entorno que se le ha introducido, exceptuando el techo de algún edificio específico y de forma muy esporádica con forma muy compleja al que no se ha podido aplicar el método de la triangulación.

Al realizar la simulación con el entorno urbano generado hemos obtenido unos resultados bastante próximos a las medidas reales que se llevaron a cabo en 2006[17]. Este hecho nos permite validar el correcto funcionamiento de la herramienta.

Con la comparación de resultado llevada a cabo se puede afirmar que se han obtenido mejores resultado en la zona cercana donde teníamos a nuestra disposición un

mayor número de muestras. En campo lejano la precisión ha sido un poco menor pero para nada negativa.

A parte del correcto funcionamiento de la herramienta se pueden extraer una serie de conclusiones a partir del presente proyecto. Al encontrarnos con algunas superficies realmente complejas a la hora de definir el techo de un edificio, es necesario un gran número de triángulos para modelarlo de forma correcta. Este hecho hace que los entornos generados cuenten con una cantidad de superficies muy elevada, lo que aumenta el tiempo de simulación notablemente. Además, este factor también es limitante a la hora de establecer el número de reflexiones. En la simulación únicamente se consideró una reflexión por rayo para limitar el cálculo de imágenes y reducir el coste computacional. En principio la herramienta funciona correctamente sea cual sea el entorno seleccionado, el número de reflexiones que se desee o el número de transmisiones pero el aumentar estos valores se verá repercutido en un tiempo de simulación realmente alto.

Para futuras líneas de desarrollo de la herramienta sería interesante en un futuro poder importar datos de mapas en 3D. Existen algunas aplicaciones de mapas como puede ser Google Maps que ya modelan sus ciudades en 3D, luego es posible que esta implementación también se lleve a cabo en una plataforma de libre distribución. Si esto fuese posible se podría adaptar la herramienta a este tipo de mapas donde se tomaría de forma precisa la altura de los edificios y no sería necesario que el usuario final la introdujese manualmente, o bien, tome la altura media de los edificios de la zona.

Con el fin de mejorar la eficiencia de la herramienta, es posible que exista la posibilidad de combinar el algoritmo de triangulación utilizado con otros supuesto que permita seleccionar de forma más óptima las diferentes superficies que definen el techo.

Para terminar destacar que sería de gran utilidad que se modificara la herramienta de trazado de rayos para que se pudiese utilizar con cualquier forma geométrica definida

por líneas rectas. Posiblemente permitiendo que el número de columnas de planes fuese variable e indicando en algún campo el número de vértices que forma la superficie. La dificultad de esta modificación sería muy grande debido a la cantidad de funciones que se deberían modificar. Aún así, si esto fuese posible se optimizarían en gran medida las simulaciones de entornos urbanos, reduciendo notablemente el tiempo de computo.

Bibliografía

- [1] BALANIS, A.S. '*Geometrical Theory of Diffraction*' en *Advanced Engineering Electromagnetics*. Ed. John Wiley & Sons. New York. 1989. p.743-847.
- [2] Parsons J.D (2000). *The Mobile Radio Propagation Channel*, Second Edition, John Wiley & Sons, ISBN 0-471-98887-X.
- [3] Pascual García, Juan. Molina Garcia-Pardo, Jose María. Juan Llácer, Leandro. *Sistemas de comunicaciones móviles. Caracterización del canal móvil*. ISBN 978-84-96997-54-7, septiembre 2010.
- [4] Rappaport, Theodore S. (2002), *Wireless Communications principles and practice*, Second Edition, Prentice Hall, ISBN 0-13-042232-0.
- [5] Tapan K, Sarkar; Zhong, Ji; Kyungjung, Kim; Abdellatif, Medouri; Magdalena Salazar-Palma. *A Survey of Various Propagation Models for Mobile Communication*. IEEE Antenas and Propagation Magazine, Vol. 45, N^o3, Junio 2003.
- [6] OSMStats (7 de octubre de 2014). «Stats».
- [7] Beyonav (2011). *Geo-Analytics on OpenStreetMap Road Data*. Archivado desde el original el 26 de noviembre de 2015.
- [8] Vicente Cabanas, César, *Análisis espacial mediante OpenStreetMap. Edición y Difusión Web de Cartografía*. Proyecto Final de Máster. Universidad de Zaragoza, 2012.

- [9] Bennett, Jonathan (2010). *OpenStreetMap*. Birmingham: Packt Publishing Ltd., 2010. 252pp. ISBN: 978-1-847197-50-4.
- [10] Ioannis F. Filippidis, *OpenStreetMap Functions for MATLAB v.0.2*, Mayo 2012
- [11] Campillo Torrecillas, Valeriano, *Programación de una herramienta de trazado de rayos 3D basada en GO y UTD*, Proyecto Final de Carrera, ETSIT, UPCT, Junio 2008
- [12] Dongsoo Har, Howard H. Xia, Senior Member, IEEE, and Henry L. Bertoni, Fellow, IEEE, *Path-Loss Prediction Model for Microcells*, IEEE Transactions on vehicular technology, Vol. 48, NO. 5, Septiembre 1999
- [13] Autor desconocido *Modelos de propagación en interiores. Radiopropagación Indoor*. Universidad de Sevilla.
- [14] Autor desconocido, *Técnica del trazado de rayos. Trazado de rayos*. Universidad de Sevilla.
- [15] *Compendio de Geografía General* página 9. José Manuel Casas Torres y Antonio Higuera Arnal. Ediciones RIALP Madrid (1977)
- [16] Recomendación UIT-R P.2040-1 (07/2015), *Efectos de los materiales y estructuras en la propagación de las ondas radioeléctricas por encima de unos 100 MHz*
- [17] Muñoz Ayuso, Helena, *Caracterización experimental del canal radio para GSM 900 y GSM 1800 en un entorno urbano*, Proyecto Final de Carrera, ETSIT, UPCT, Octubre de 2006.
- [18] Hernando Rábanos, José María, *Comunicaciones Móviles*, Editorial Centro de Estudios Ramón Areces.