



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



FACULTAD DE
INGENIERIA

Simulación del sonido con trazado de rayos

Tesis de grado

Camilo Satut

Tesis de grado presentada en la Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de grado en Ingeniería en Computación.

Supervisores:

Eduardo Fernández

Benoit Beckers

RESUMEN

El trazado de rayos o *ray tracing* es una técnica muy utilizada en el área de la computación gráfica, debido a sus posibilidades para modelar de forma simple la interacción de la luz con los objetos.

En este trabajo se desea explorar su uso para modelar la propagación del sonido en lugar de la luz. Tanto la luz como el sonido son fenómenos ondulatorios que se propagan por el espacio, pero que pueden modelarse como rayos rectilíneos que interactúan con los objetos a través de aspectos esencialmente geométricos (reflexión, refracción, ley de Snell, etc.). A su vez, presentan diferencias que deben ser contempladas en las simulaciones. En ese contexto, se quiere realizar un motor “acústico” que permita simular algunos resultados prácticos de la propagación del sonido en un entorno arquitectónico.

Para el desarrollo de este trabajo en primera instancia se realizó una etapa de investigación, donde se relevaron aspectos físicos de la propagación del sonido y aspectos computacionales sobre el almacenamiento y reproducción del sonido en una computadora. También se analizaron trabajos previos y soluciones existentes para el problema en cuestión. Posteriormente, se realizó una etapa de diseño y desarrollo que culminó con la implementación de un motor acústico¹. Este programa permite, a partir de escenas y sonidos cargados, simular el sonido resultante de la interacción de ambos elementos. Este motor se limita a simular la reflexión del sonido y no contempla otros fenómenos acústicos como la difracción o la transmisión.

Palabras claves:

Trazado de rayos, Simulación del sonido, Auralización.

¹Disponible en GitHub: <https://github.com/cameelo/AudioRendering>

Tabla de contenidos

1	Introducción	1
1.1	Motivación y problema	1
1.2	Objetivos	3
1.3	Estructura del documento	3
2	Estado del arte	5
2.1	Fundamentos del sonido	5
2.1.1	El sonido como onda	5
2.1.2	Rayo como simplificación de una onda	8
2.1.3	Trazado de rayos y caminos	9
2.1.4	Fenómenos de propagación del sonido	11
2.1.5	Energía acústica	16
2.1.6	El oído humano y la percepción del sonido	18
2.1.7	Nivel Sonoro	19
2.1.8	Respuesta al impulso	21
2.1.9	Tiempo de reverberación y EDT	22
2.2	Representación digital del sonido	24
2.2.1	Frecuencia de muestreo	25
2.2.2	<i>Bit depth</i>	26
2.2.3	Canales de audio	26
2.2.4	Adimensionalidad	27
2.3	Transformada de Fourier	28
2.4	Simulación vs Auralización	30
2.5	Soluciones existentes a la simulación del sonido	31
2.5.1	VRWorks Audio	31
2.5.2	Wayverb	32
2.6	Bibliotecas útiles	32

2.6.1	Embree	33
2.6.2	OpenGL	33
2.6.3	RtAudio	34
2.7	Trabajos relacionados	34
3	Solución propuesta	37
3.1	Alcance y objetivos	37
3.2	Diseño	37
3.2.1	Módulo principal	38
3.2.2	Módulo de geometría	39
3.2.3	Módulo de render	40
3.2.4	Módulo de trazado de caminos	41
3.2.5	Módulo de audio	42
4	Implementación	45
4.1	Traza de rayos	45
4.1.1	Generación de los rayos	45
4.1.2	Generación de Caminos	47
4.1.3	Cálculo de la intensidad	48
4.2	Construcción de la IR	49
4.3	Implementación de la auralización	50
4.3.1	Convolución de las señales	50
4.3.2	Buffer circular	54
4.3.3	Procesamiento de audio desde micrófono	56
4.3.4	Procesamiento de audio desde archivo	57
5	Experimentos y resultados	59
5.1	Ambiente de prueba	59
5.2	Escenas estudiadas	59
5.3	Análisis de la simulación	60
5.3.1	Análisis cualitativo	61
5.3.2	Forma de los picos	67
5.3.3	Cantidad de rayos y distribución	68
5.3.4	Análisis cuantitativo	75
5.4	Análisis de la auralización	78
5.4.1	Cuellos de botella	78
5.5	Tiempos de ejecución	79

6 Conclusiones y trabajo futuro	81
6.1 Conclusiones	81
6.2 Trabajo futuro	83
6.2.1 Modelado de la escena	83
6.2.2 Difracción, transmisión y frecuencias	84
6.2.3 Ancho de los picos	84
6.2.4 Fourier como solución de la convolución	85
6.2.5 Soluciones a problemas alternativos	86
6.2.6 Extensión del programa como herramienta de trabajo . .	87
Lista de figuras	89
Lista de tablas	93
Referencias bibliográficas	95
A Archivo de configuración	101

Glosario

- Auralización** Refiere al proceso de hacer audible una señal de audio de forma de simular la escucha en una escena determinada. 30
- Decibel** Unidad del nivel sonoro. 20
- EDT** Early Decay Time. 24
- Escena** Ambiente o círculo en que se desarrolla una actividad. En el contexto de este proyecto, los términos escena y recinto son sinónimos. 1
- Factor remanente** Fracción de la energía inicial del rayo que aún se conserva al llegar al oyente 41
- HRTF** Función característica del receptor 18
- IR** Respuesta al impulso 21
- Nivel sonoro** Medida de cómo el ser humano percibe un sonido de cierta intensidad, potencia o presión. 19
- Path tracing** Técnica que permite generar rutas de transporte de luz entre las fuentes de luz y los puntos de la escena para los que queremos calcular los valores de radiancia. En el caso del sonido, los caminos van desde una fuente de sonido hasta un receptor, a través de una escena. 9
- Recinto** Espacio, generalmente cerrado, comprendido dentro de ciertos límites. En el contexto de este proyecto, los términos escena y recinto son sinónimos. 2
- Sample** Valor de la presión efectiva de una señal de audio en un tiempo determinado. Representa el desplazamiento de la onda de sonido en ese tiempo. 25
- Sample rate** Frecuencia con la que se miden los valores de la presión efectiva de una señal de audio. 25
- Simulación** Refiere a la simulación de la propagación del sonido en una escena. 30

Capítulo 1

Introducción

1.1. Motivación y problema

El trazado de rayos es una técnica comúnmente utilizada para la simulación de iluminación. Esto puede verse en distintos ambientes, como en videojuegos, películas, modelado de arquitectura, etc.

Debido a las propiedades físicas de la luz, es posible modelar su propagación como rayos que viajan desde una fuente de luz, a través de la escena, hasta los ojos del observador. Por ejemplo, uno de los aspectos particulares de la luz es su velocidad, que al ser tan grande se asume infinita al modelarla.¹ Resulta interesante la búsqueda de otros fenómenos físicos que puedan modelarse computacionalmente mediante el trazado de rayos, y su uso en aplicaciones variadas.

El sonido es otro fenómeno físico de interés para las personas. El buen diseño de la acústica de un espacio arquitectónico es esencial para la calidad de vida de las personas, por ejemplo, en el caso de un hogar o de un ambiente de trabajo. El ruido es uno de los agentes contaminantes más habituales en los puestos de trabajo, es peligroso para la audición y su exposición prolongada, incluso a niveles inferiores a los valores límite establecidos, puede producir efectos negativos en la salud [28]. Por este motivo, el acondicionamiento acústico de los espacios de trabajo es en ocasiones obligatorio.

Por otro lado, es esencial para poder disfrutar de actividades de recreación en un teatro, cine o arena de eventos y en lugares como salones de clase donde es

¹Si bien el debate entre luz instantánea o con velocidad finita estuvo instalado por siglos, no fue hasta 1676 en que el danés Ole Rømer (1644-1710) pudo dar un valor razonablemente aproximado a la velocidad de la luz, basado en datos experimentales [19].

necesario lograr cierto nivel de inteligibilidad del locutor. Otro tipo de medios, como los videojuegos, también transmiten información al espectador a través del sonido. Si bien, en este caso, el realismo no suele ser estrictamente necesario, la posibilidad de simular el sonido de un recinto podría ayudar al diseño del audio. La alternativa más utilizada, es construir el sonido de cada recinto de forma manual. Esto se logra creando filtros que modifican las características de un archivo de sonido, similar a como funciona un pedal de efectos de una guitarra eléctrica. Puede verse entonces que la simulación del audio puede ser de gran utilidad, por ejemplo, permitiendo evaluar un espacio acústico auditivamente antes que este sea construido o dando la posibilidad de recrear espacios acústicos que ya no existen como ruinas de catedrales.

Tanto la luz como el sonido son fenómenos físicos que comparten características similares: Ambos parten de una fuente, se propagan a través del espacio como ondas, e interactúan con los objetos de una escena de un modo u otro dependiendo de sus características. Sin embargo, también hay diferencias sustanciales entre ambos. La velocidad de propagación del sonido es unas 10^6 veces más lenta que la de la luz, y esa lentitud es claramente perceptible en fenómenos cotidianos como el eco, el tiempo que pasa entre un rayo y su trueno, y los cambios en el tono de la sirena de una ambulancia que se aleja a gran velocidad. Otra diferencia importante es que la longitud de onda de los sonidos perceptibles por el ser humano es también unas 10^6 veces menor a la de la luz visible.

Por lo antes mencionado es posible ver que la traza de rayos también es aplicable al modelado del sonido, aunque no sin realizar las adaptaciones correspondientes. Resulta entonces de interés estudiar estas posibilidades. Sin embargo, el trazado de rayos ve sus aplicaciones mayoritariamente en la simulación de la luz, por esta razón la cantidad de trabajos y bibliografía disponibles son escasos. Además, muchos de los trabajos disponibles se enfocan en la simulación del sonido en condiciones muy específicas [46] o en mejoras de otros modelos donde no se explican todos los desafíos del problema [47]. Por lo tanto, resulta interesante conocer cómo es el proceso de principio a fin, es decir, desde el trazado de rayos hasta la reproducción del sonido obtenido.

1.2. Objetivos

Los objetivos para este proyecto fueron definidos considerando los elementos mencionados anteriormente. En particular, el objetivo primario de este proyecto es la documentación de los principales fundamentos necesarios, tanto físicos como de programación, para la implementación de la simulación del sonido. Luego, como culminación de este proceso, realizar un programa que realice estas tareas.

Como objetivo secundario se exploraron alternativas para mejorar los resultados y optimizar los procesos. Se optó por estudiar, en mayor profundidad, estos pocos aspectos principales de la simulación del sonido, en lugar de cubrir todos los aspectos posibles.

El programa creado debe ser considerado un prototipo o una prueba de concepto de una herramienta de trabajo a ser utilizada por arquitectos, ingenieros, desarrolladores de videojuegos, etc y que debe presentar *features* que permitan realizar distintas pruebas para un análisis detallado de los resultados obtenidos.

1.3. Estructura del documento

El resto de este documento está dedicado a detallar todo lo realizado a lo largo del proyecto. El Capítulo 2 detalla el estado del arte, fundamentos de la física detrás del sonido, programas y bibliotecas existentes. El Capítulo 3 ilustra el diseño de la solución propuesta y cómo se resuelve cada problema. El Capítulo 4 menciona aspectos de la implementación del diseño. El Capítulo 5 da cuenta del análisis de los resultados obtenidos y comparaciones con mediciones reales. Por último, el Capítulo 6 presenta las conclusiones y posibles trabajos futuros que surgen de lo logrado en este proyecto.

Capítulo 2

Estado del arte

En este capítulo se detallan los conceptos esenciales para la comprensión del trabajo realizado en este proyecto. Se presentan definiciones y conceptos sobre el sonido, así como también trabajos previos, soluciones existentes al problema del trazado de rayos para el sonido y bibliotecas que pueden ser usadas en la implementación de una solución.

2.1. Fundamentos del sonido

2.1.1. El sonido como onda

Desde el punto de vista perceptual, el sonido es una sensación producto de una excitación del mecanismo auditivo del ser humano u otro ser vivo. Sin embargo, desde el punto de vista físico, el sonido puede verse como un fenómeno de onda en el aire u otro medio elástico. El problema a estudiar determina cuál de las 2 definiciones será utilizada. Si el problema concierne a la perturbación en el aire generada por un altoparlante, se utilizará la segunda definición. Por otro lado, si el problema concierne a cómo una persona percibe dicha perturbación, se utilizará la primera definición [25]. En este trabajo se explicará cómo ambas definiciones son importantes para el problema de la simulación del sonido, aunque la definición física requiere un estudio más profundo.

En física, una onda es descrita como una perturbación de algún tipo, es decir, un cambio de una condición de equilibrio [27]. En el sonido, el cambio se da en la posición de las partículas del medio. Normalmente cuando se piensa en el sonido, se piensa en el aire, sin embargo, el sonido también se propaga a través de otros medios elásticos como el agua, acero y concreto. Las partículas

de estos medios son desplazadas de su posición de equilibrio por la onda, la partícula luego oscila hasta volver a su posición de equilibrio como se muestra en la Figura 2.1.

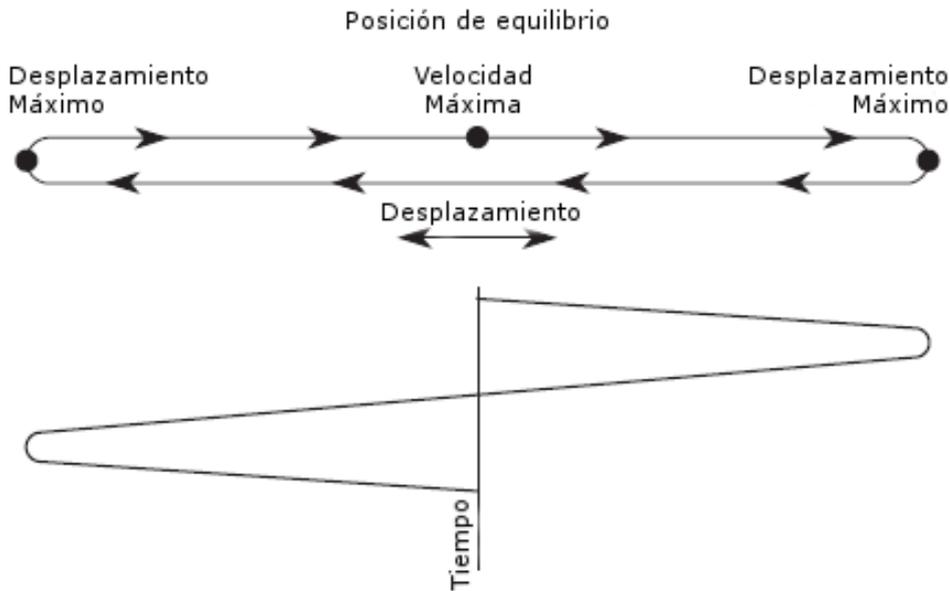


Figura 2.1: Desplazamiento de una partícula. Gráfico tomado de [25].

Este movimiento en las partículas del medio hace que estas se agrupen en alguna zonas, como puede verse en la Figura 2.2. Estas zonas marcan lugares donde la presión del medio es ligeramente mayor a la presión atmosférica ¹ y son la cresta de la onda, mientras que las zonas donde las partículas están más dispersas son los valles.

Estos aspectos de la onda de sonido puede expresarse como un conjunto de propiedades que determinan las características del sonido percibido por el receptor [27]. A continuación se detallan las propiedades más importantes de una onda para el trabajo realizado en este proyecto:

1. **Desplazamiento:** Refiere al desplazamiento con respecto al valor de equilibrio. Como fue mencionado antes, para una onda de sonido, este valor hace referencia a la variación de presión con respecto a la presión atmosférica media. Por este motivo el desplazamiento en una onda de

¹El sonido más débil que un ser humano puede oír corresponde a una variación de $20\mu Pa$ sobre la presión atmosférica que es unas 5000 millones de veces mayor en magnitud [25].

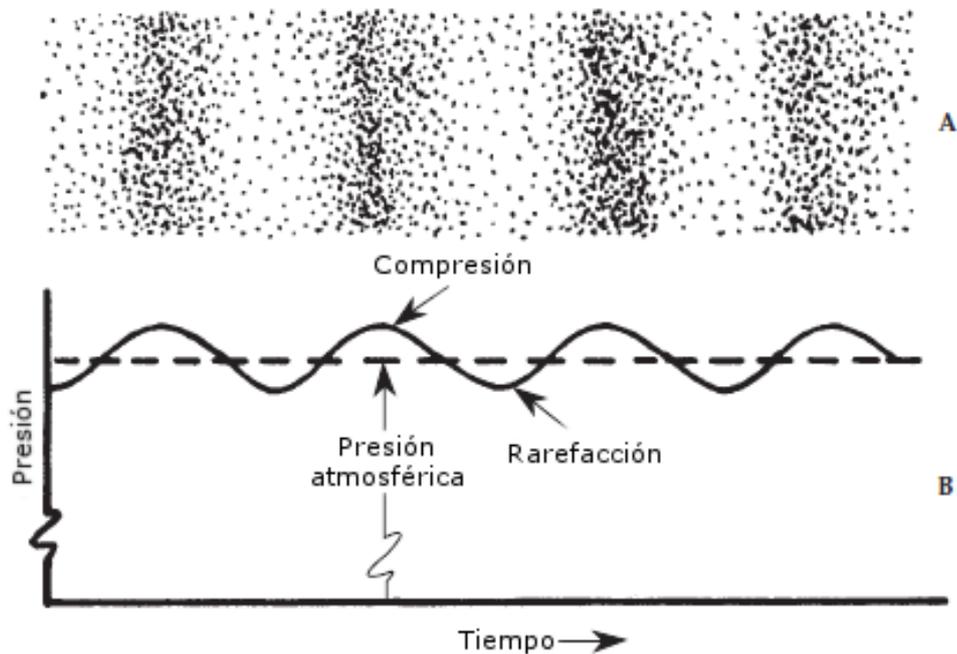


Figura 2.2: Las regiones del medio donde las partículas se acumulan corresponden con las crestas de la onda. Gráfico tomado de [25].

sonido se mide en **Pascales (Pa)**. Al máximo de estos valores en valor absoluto, es decir, los valores en las crestas y en los valles se le llama **amplitud**.

2. **Frecuencia:** La frecuencia cuenta el número de ciclos de la onda que ocurren cada segundo, y es medida en **Hertz (Hz)**. Un ciclo refiere a la porción más pequeña de la onda que se repite, por ejemplo, la porción que va desde una cresta a la siguiente. El largo de esta porción se llama **longitud de onda (λ)** y es inversamente proporcional a la frecuencia [25].

$$f = \frac{c}{\lambda}. \quad (2.1)$$

Donde c es la velocidad de la onda en el medio. Para el ser humano, la frecuencia de una onda de sonido determina el tono percibido. Por ejemplo, una frecuencia de 440 Hz corresponde a un La (A_4 en cifrado americano).

3. **Velocidad:** También llamada velocidad de fase. Se trata del tiempo que demora la cresta u otro punto cualquiera de la onda en recorrer una

distancia determinada. Para el sonido en el aire a $20\text{ }^{\circ}\text{C}$ la velocidad de la onda es de $343\frac{\text{m}}{\text{s}}$.

Estas propiedades se pueden apreciar de forma gráfica en la Figura 2.3.

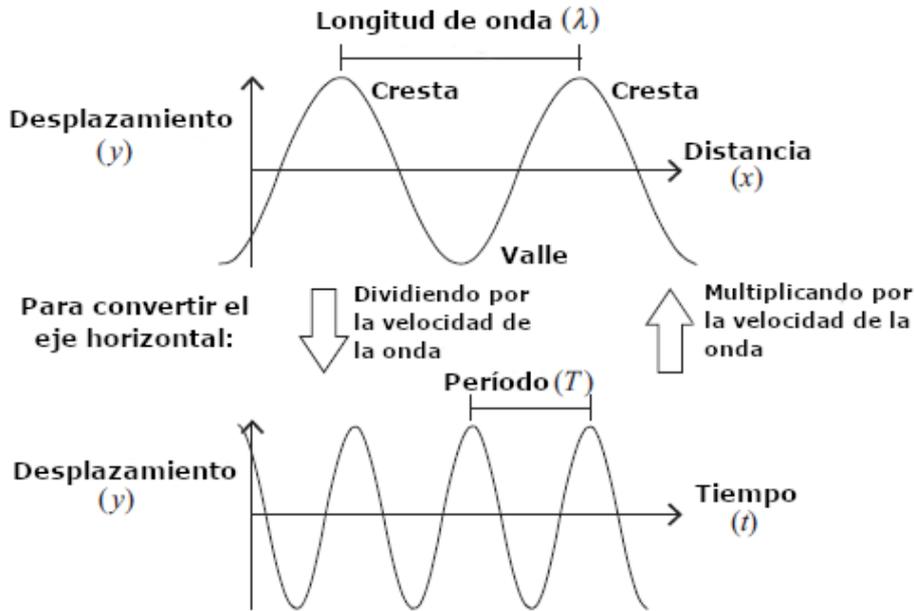


Figura 2.3: Ilustración de una onda y sus propiedades. Gráfico tomado de [27].

2.1.2. Rayo como simplificación de una onda

Se le llama frente de onda a todos los puntos del medio que son alcanzados en un mismo instante por una onda. Como puede verse en la Figura 2.4, la dirección de la normal del frente de onda define el camino por el cual la energía se propaga. A este camino se le llama rayo y para una fuente puntual esférica, los rayos irradian hacia afuera de la fuente.

A esta simplificación se le conoce como **teoría geométrica de rayos** o *geometric ray theory* [14]. Siempre que las variaciones en la velocidad de la onda no sean demasiado altas, estos rayos pueden comportarse como rayos de luz (ajustando sus índices de refracción).

Sin embargo, esta simplificación no es capaz de modelar todos los fenómenos de onda, como por ejemplo, la superposición de ondas. Sin información adicional no es posible conocer el valor de desplazamiento o amplitud de la

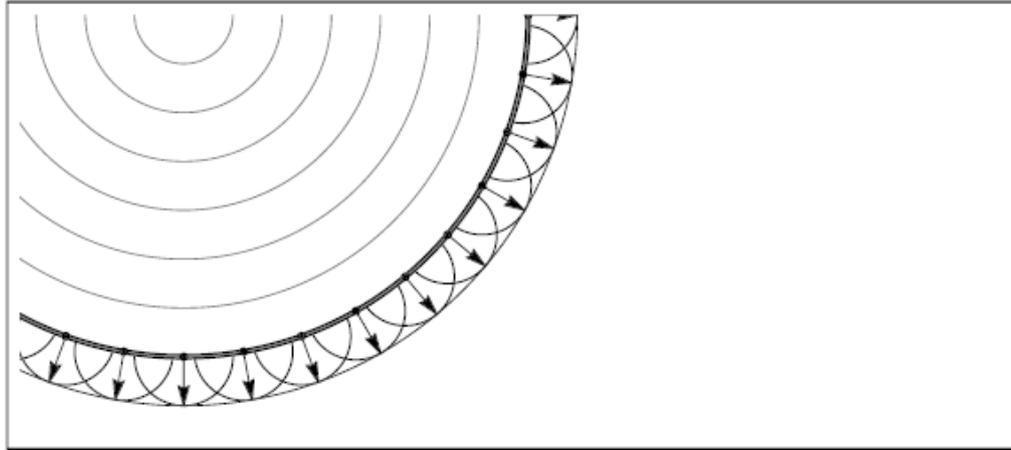


Figura 2.4: Las flechas indican la dirección de la propagación de la energía en una onda longitudinal esférica.

onda para un punto en el espacio en un instante de tiempo dado. Por lo tanto, esta simplificación no permite modelar la interferencia de ondas.

2.1.3. Trazado de rayos y caminos

Hasta ahora se ha hablado sobre cómo la luz y el sonido pueden ser modelados por rayos, pero, ¿Por qué es de utilidad esta simplificación? En iluminación, la **traza de rayos** o *ray tracing* es una técnica utilizada para determinar cómo la luz interactúa con los distintos elementos de una escena. Cada rayo se emite desde un punto y en una dirección determinada, y lo que se busca es determinar con qué objetos de la escena y en qué coordenadas interseca. [12].

Para los algoritmos de iluminación global, uno o más rayos son emitidos desde la cámara por cada píxel de la imagen final. Estos rayos se propagan a través de la escena hasta llegar a una fuente de luz o cumplir otra condición de parada, como puede verse en la Figura 2.5. En el caso de que el rayo haya llegado a una fuente de luz, el trayecto realizado por éste representa uno de los caminos que la luz sigue hasta llegar a la cámara. Esta técnica, que consiste en hallar los caminos que van desde la fuente de luz hasta un punto es conocida como **trazado de caminos** o *path tracing*. Cada uno de estos caminos hallados contiene la información de los objetos de la escena con los que la luz interactúa y es a través de estas interacciones que se calcula la contribución de este camino al color de la imagen final [12].

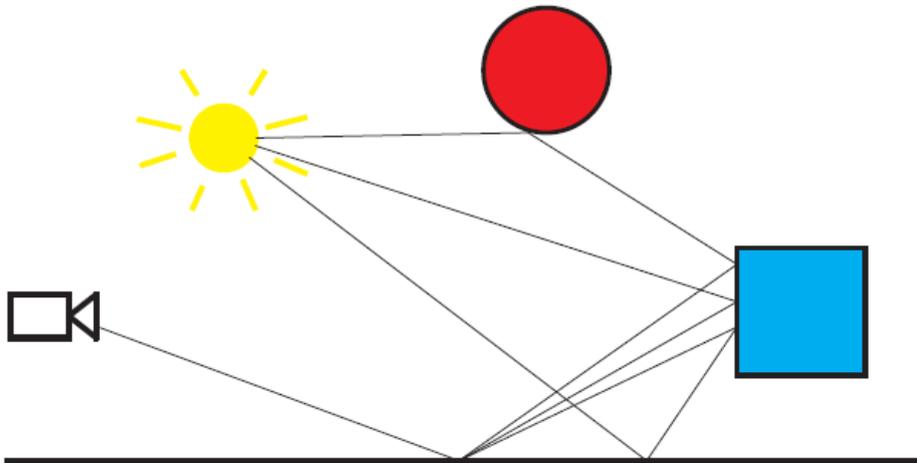


Figura 2.5: *Path tracing*. Los rayos se propagan desde la cámara hasta la fuente de luz. Gráfico tomado de [12].

Con lo expuesto en la Sección 2.1.2 puede verse que es posible construir un *path tracer* para simular la propagación del sonido. Como en el sonido no hay píxeles, los rayos de sonido pueden simplemente emitirse desde un punto correspondiente a la posición de una fuente de sonido en varias direcciones. Para generar las direcciones de los rayos de sonido, puede considerarse a la fuente como una esfera y tomar direcciones uniformemente distribuidas sobre su superficie. El receptor también es modelado como una esfera, como puede verse en la Figura 2.6.

Los rayos que llegan hasta el oyente describen los caminos por los que el sonido se propaga a través de la escena. Una de las principales diferencias entre el *path tracing* para luz y para sonido es el tiempo de propagación de los rayos. Como se explicó en el Capítulo 1 la diferencia entre las velocidades de propagación de la luz y el sonido genera diferencias sustanciales en cómo se perciben ambos fenómenos. Por este motivo, a diferencia de los caminos para la luz, la distancia de los caminos de sonido determinan cuándo llega este al receptor. Esto será explicado en mayor detalle en los Capítulos 3 y 4.

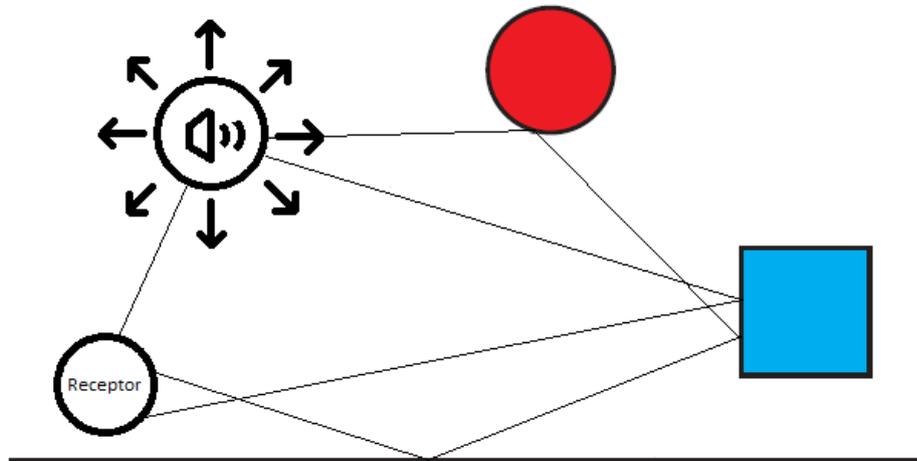


Figura 2.6: *Path tracing* para audio. Los rayos se propagan desde la fuente de sonido hasta el receptor.

2.1.4. Fenómenos de propagación del sonido

Reflexión

La reflexión ocurre cuando la libre propagación de una onda se ve interrumpida por algún objeto de la escena. En esta situación, parte de la onda incidente se verá reflejada en alguna dirección.

Según las características del objeto en cuestión, la reflexión puede ocurrir de varias formas posibles. En general se distinguen 2 formas, tal como se aprecia en la Figura 2.7.

1. **Reflexión especular:** Ocurre cuando la superficie reflejante es lisa. En este caso el ángulo de reflexión es igual al ángulo de incidencia.
2. **Reflexión difusa:** Ocurre cuando la superficie reflejante es rugosa. En este caso la onda es reflejada en direcciones aleatorias. La dirección de las reflexiones resultantes depende de la superficie. Puede considerarse que las reflexiones resultantes siguen la regla del coseno donde la intensidad del rayo depende de la dirección en la que es reflejada. La distribución también puede modelarse a partir de una **BRDF** (de sus siglas en inglés **Bidirectional Reflectance Distribution Function**) que determina cual será la intensidad del rayo reflejado dependiendo del punto en la superficie, dirección y ángulo de incidencia del rayo [36].

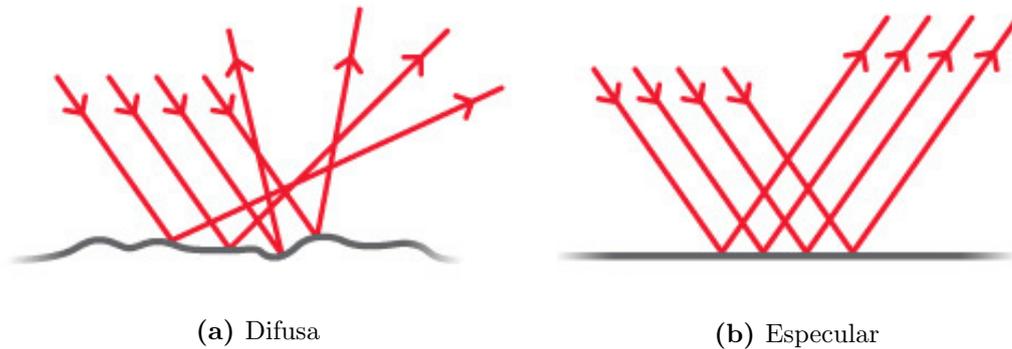


Figura 2.7: Posibles formas de reflexión de la luz con superficies.

Es evidente que ninguna superficie es completamente lisa, sin embargo, una superficie puede considerarse como lisa (y consecuentemente a su reflexión como especular) si la longitud de onda incidente es suficientemente pequeña frente a las dimensiones del objeto reflejante y suficientemente grande con respecto a las rugosidades de su superficie ([34], [41]). En la Figura 2.8 pueden verse 3 casos distintos. De izquierda a derecha:

1. La frecuencia es demasiado pequeña frente a las rugosidades de la superficie, es decir, la longitud de onda es mucho mayor a las rugosidades. En este caso, la superficie actúa como una superficie lisa, la reflexión es especular según la “macro-normal” [21].
2. La frecuencia es proporcional a la rugosidad de la superficie, y la reflexión resultante es difusa.
3. La frecuencia es mucho mayor que las rugosidades. La reflexión es especular pero sigue la distribución de las “micro-normales” o rugosidades.

Para el caso del sonido es posible modelar las reflexiones como puramente especulares. Si bien esta simplificación puede no ser completamente adecuada para todo el rango de frecuencias audibles, los resultados de modelos de este estilo han sido buenos [18].

Difracción

Los frentes de onda y por lo tanto, los rayos de sonido, se propagan de forma rectilínea. Incluso si se reflejan seguirán su camino de forma rectilínea en una nueva dirección. Sin embargo, los obstáculos de una escena pueden

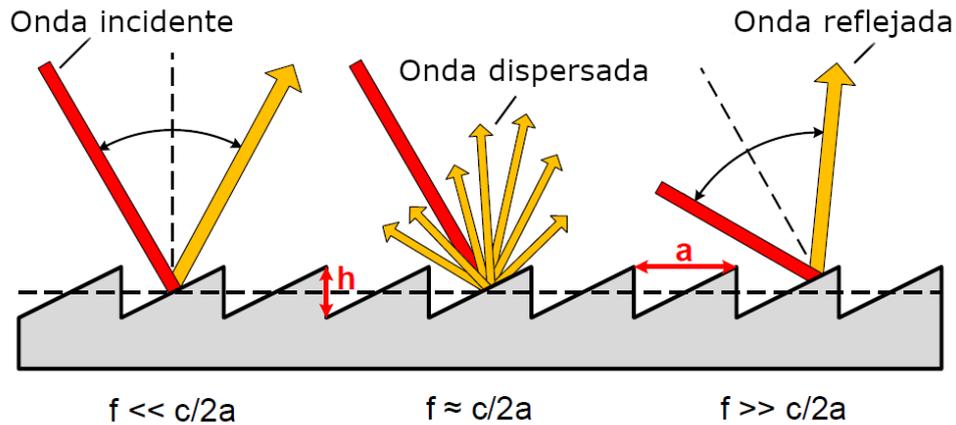


Figura 2.8: Tipos de reflexión dependiendo de la frecuencia de la onda. Donde: f es la frecuencia de la onda incidente, c la velocidad del sonido, h y a la altura y el ancho de las irregularidades de la superficie. Gráfico tomado de [41].

hacer que el sonido cambie su dirección de propagación como si doblara. A este fenómeno se le llama difracción.

La difracción depende del obstáculo y de la longitud de onda como puede verse en la Figura 2.9. De hecho, la difracción ocurre en todos los fenómenos de onda, incluida la luz, y es más fuerte cuanto mayor es la longitud de onda. La difracción en la luz suele descartarse en las simulaciones, pero no así la difracción del sonido [25].

Si un frente de onda se encuentra con una apertura en una pared como se muestra en la Figura 2.10, la difracción puede crear el efecto de un nuevo frente de onda.

La influencia de la difracción en espacios cerrados suele ser de un orden menor que la de las reflexiones [49][30], y es por este motivo principal que no fue incluido en este proyecto. Keller [31], ha desarrollado la teoría geométrica de la difracción, que extiende la teoría geométrica de rayos.

Refracción

La refracción es otro fenómeno que modifica la dirección de propagación del sonido. Esta ocurre cuando la onda atraviesa 2 medios con diferentes velocidades de propagación. La dirección de la onda cambia, siguiendo el principio de Fermat que estipula que la onda atravesará el camino con menor tiempo de

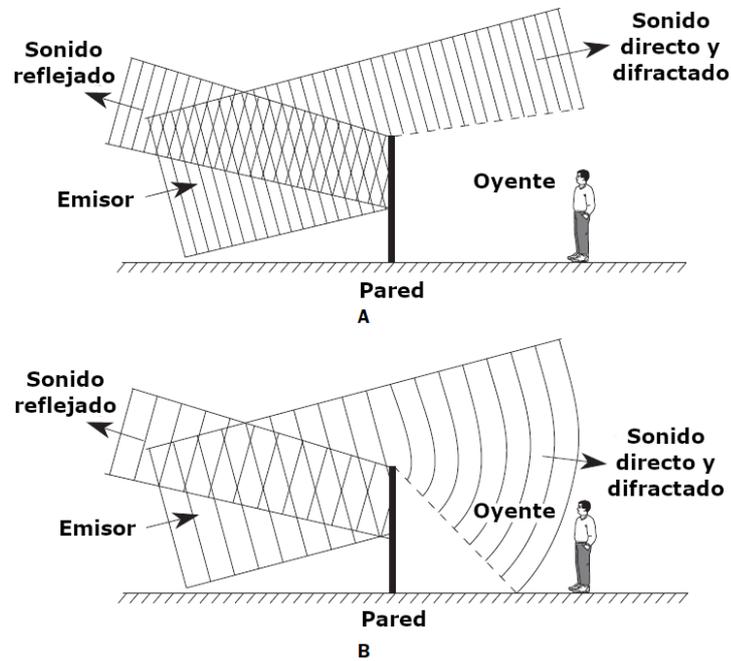


Figura 2.9: (A) Un frente de onda con una longitud de onda pequeña continuará su propagación con cambios leves en su dirección. (B) Un frente de onda con longitud de onda mayor ve su propagación modificada de forma más pronunciada. Gráfico tomado de [25].

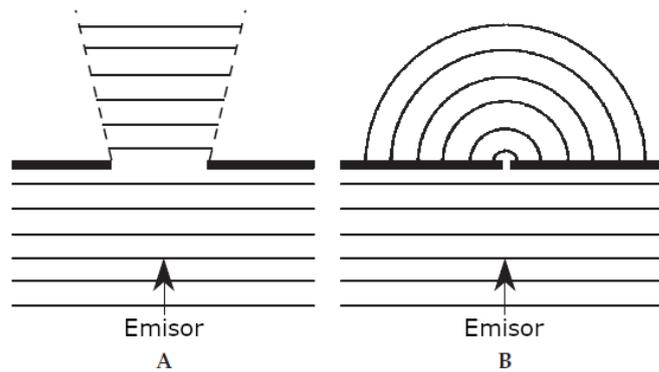


Figura 2.10: (A) Una apertura grande con respecto a la longitud de onda dejará pasar los frentes de onda con pocas modificaciones. (B) Si la apertura es pequeña con respecto a la longitud de onda, los frentes de onda que atraviesen la apertura se comportarán como nuevas fuentes. Gráfico tomado de [25].

propagación. En la Figura 2.11 puede verse como la trayectoria de los rayos se ve afectada al pasar de un medio al otro.

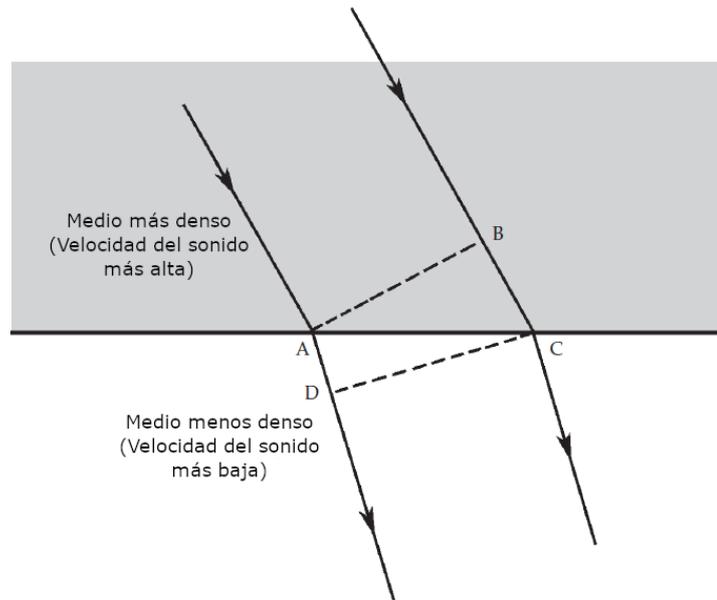


Figura 2.11: Los rayos de sonido son refractados cuando atraviesan a un medio con distinta velocidad de propagación. Notar que el frente de onda A-B no es paralelo con el frente de onda C-D. Gráfico tomado de [25].

Transmisión

Cuando una onda de sonido choca contra una superficie, parte de la energía se transmite a través de esta. Para el caso de 2 cuartos separados por una pared, la energía se transmitirá a través de la pared de un cuarto a otro. La cantidad de energía transmitida, reflejada y absorbida depende de las propiedades del material (ver Figura 2.12).

La cantidad de energía transmitida también depende de la frecuencia de la onda. Ondas de menor frecuencia se transmiten mejor que ondas de mayor frecuencia [41].

Difusión

Todos los fenómenos antes mencionados determinan cómo el sonido se distribuye en el ambiente. A esto se le llama difusión del sonido. La difusión

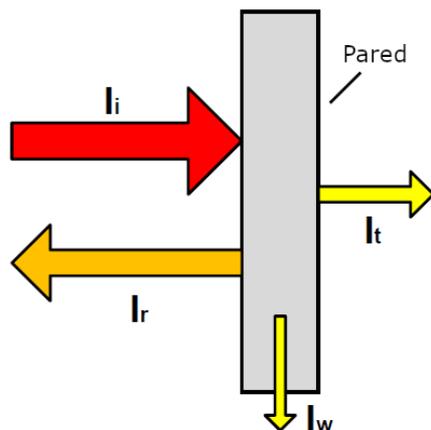


Figura 2.12: Transmisión del sonido. Donde I_i es la intensidad incidente, I_r la intensidad reflejada, I_t la intensidad transmitida e I_w la intensidad absorbida. Gráfico tomado de [41].

suele considerarse como homogénea, cuando el sonido llega con la mismas características de todas las direcciones, sin embargo, esto rara vez ocurre en la práctica [25].

2.1.5. Energía acústica

Se trata de otra forma de cuantificar el sonido transmitido. La energía acústica puede expresarse de 3 formas distintas [17]:

1. Potencia: Representa la energía por unidad de tiempo y es medida en watts (W).
2. Intensidad: Representa la potencia por unidad de superficie y es medida en W/m^2 .
3. Como el cuadrado de la presión eficaz. La presión eficaz es la variación de la presión producida por la onda con respecto a la presión atmosférica media en reposo.

La potencia se utiliza para describir la energía acústica de una fuente de sonido. Los equipos de sonido como parlantes y amplificadores siempre indican la potencia de los mismos.

Por otro lado, la intensidad y el cuadrado de la presión se utilizan como magnitudes de la energía percibida por el oyente y no la energía emitida por la fuente. La intensidad y el cuadrado de la presión se relacionan de la siguiente forma [17]:

$$I = \frac{p_{eff}^2}{400} \quad (2.2)$$

Donde p_{eff} es la presión eficaz, es decir, la presión con respecto a la presión atmosférica media.

Intensidad de un rayo

Para construir un *path tracer* de sonido como el descrito en la Sección 2.1.3 es necesario hallar la energía acústica percibida por el receptor, expresada por ejemplo, en intensidad. Yang y Shield [51] presentan un método para calcular la intensidad correspondiente a un rayo registrada por el receptor en su modelo *NORMAL*. Como puede verse, la Ecuación 2.3 tiene como unidades $\frac{m \cdot W}{m^3} = \frac{W}{m^2}$.

$$I = \frac{sE_0e^{-h \cdot d} \prod_i (1 - \alpha_i)}{V_{receptor}} \quad (2.3)$$

Donde:

- s es la distancia que el rayo recorre dentro del receptor expresada en metros.
- E_0 es la energía acústica inicial del rayo expresada en watts.
- d es la distancia recorrida hasta el receptor en metros.
- h el coeficiente de absorción del aire.
- i es la cantidad de reflexiones del rayo hasta llegar al receptor.
- α_i es el coeficiente de absorción del objeto que produce la i -ésima reflexión. La fórmula $(1 - \alpha_i)$ representa el factor reflejado y se le llama coeficiente de reflexión.
- $V_{receptor}$ es el volumen del receptor, dado que, por ejemplo, el receptor puede ser modelado como una esfera.

La energía acústica inicial E_0 puede calcularse en base a la potencia de la fuente y la cantidad de rayos emitidos. Dada la cantidad total de rayos N a emitir, la energía acústica inicial de cada rayo se calcula como la fracción correspondiente de la potencia de la fuente W .

$$E_0 = \frac{W}{N} \quad (2.4)$$

2.1.6. El oído humano y la percepción del sonido

Existe una distinción importante entre las cualidades físicas que se han descrito hasta ahora, y cómo el ser humano percibe al sonido. En una descripción puramente física bastaría enumerar las intensidades de cada frecuencia excitada por la emisión de un sonido y explicitar luego cómo éste se va modificando en su propagación hasta alcanzar un receptor determinado. Sin embargo, el oído humano percibirá este sonido de forma no lineal, con una sensibilidad a las intensidades que varían en función de la frecuencia [17]. Lo antes mencionado refiere a la **sonoridad** o *loudness*, que es el rasgo de la sensación auditiva en términos del cual los sonidos pueden ser ordenados en una escala que se extiende del silencio a lo muy sonoro [16].

Una forma de ejemplificar la percepción humana del sonido es con los valores de las frecuencias de las notas musicales. Al aumentar la frecuencia, detectar la variación de un tono requiere de una diferencia de frecuencias mayor. Volviendo al ejemplo del A_4 , para que ocurra una variación de un semi tono (el menor intervalo entre 2 notas para la escala común de 12 tonos occidental), la frecuencia debe variar de 440 Hz (A_4) a $466,16 \text{ Hz}$ ($A\#_4$), una diferencia de $26,16 \text{ Hz}$. Sin embargo, 4 octavas más arriba la frecuencia debe variar de 7040 Hz (A_8) a $7458,62 \text{ Hz}$ ($A\#_8$)², una diferencia de $418,62 \text{ Hz}$ [8]. Notar que aumenta la diferencia entre los pares de frecuencias, pero los cocientes se mantienen iguales:

$$\frac{466,16}{440} = \frac{7458,62}{7040} = 1.059 \quad (2.5)$$

A partir del ejemplo anterior puede verse cómo el oído no percibe al sonido de forma lineal, sino que lo hace de forma logarítmica en concordancia con la ley de Weber-Fechner [17]. Esta ley establece que los ratios de las medidas son mejores descriptores que las diferencias para evaluar la percepción de fenómenos como la visión, audición, vibración y hasta choques eléctricos [25].

Otro aspecto importante sobre la percepción del sonido es la dirección con la que el sonido llega al oído, esto concierne a la localización del sonido. Si se considera una señal binaural, es decir, una señal independiente para cada oído, es de interés saber cómo cierto sonido es percibido en cada oído. La **HRTF** por sus siglas en inglés *Head Related Transfer Function* es

²Se dice que A_8 tiene la misma altura tonal que A_4 , pero tiene una mayor altura espectral [16]

una función característica del receptor del sonido. Esta establece cómo cierto sonido es percibido por el usuario dado su tiempo de propagación, intensidad, frecuencia y ángulo con el que llega al receptor (ver Figura 2.13). Cada oído tiene su propia HRTF.

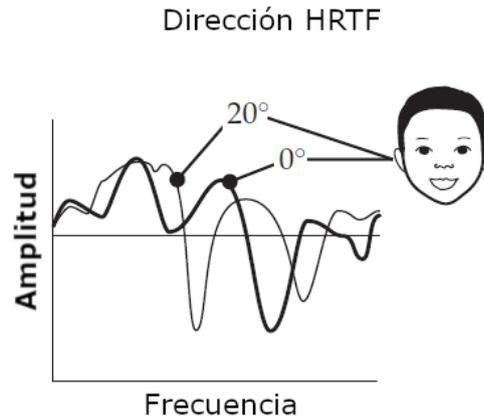


Figura 2.13: La HRTF muestra cómo una frecuencia es percibida de forma diferente dependiendo del ángulo de incidencia al oído. Gráfico tomado de [25].

A una frecuencia de 1000 Hz , el umbral de audibilidad del oído humano se da en los $2 \times 10^{-5} \text{ Pa}$ y el umbral del dolor en los 20 Pa [17].

2.1.7. Nivel Sonoro

Por lo expuesto en la sección anterior resulta de interés tener medidas que permitan linealizar la percepción logarítmica del sonido. El nivel sonoro se calcula respecto a valores base que surgen del umbral de audición del oído humano.

$$L_p = 10 \log_{10} \frac{p_{eff}^2}{p_0^2} \quad (2.6)$$

$$L_I = 10 \log_{10} \frac{I}{I_0} \quad (2.7)$$

$$L_W = 10 \log_{10} \frac{W}{W_0} \quad (2.8)$$

Donde:

- L_p es el nivel sonoro en presión.

- L_I es el nivel sonoro en intensidad.
- L_W es el nivel sonoro en potencia.
- $p_0 = 2 \times 10^{-5} [Pa]$ es la presión base.
- $I_0 = 10^{-12} [W \cdot m^{-2}]$ es la intensidad base.
- $W_0 = 10^{-12} [W]$ es la potencia base.

Al utilizar el logaritmo de estos cocientes, es posible utilizar la diferencia de los niveles sonoros como una estimación de la percepción de la intensidad del sonido mencionada en la sección anterior.

Notar que todos los niveles sonoros carecen de unidades ya que en el cociente las unidades se cancelan. Al valor resultante se lo mide en una nueva medida llamada **Decibelio** o **Decibel** (dB). Como puede verse, el nivel sonoro de un sonido con una presión, intensidad o potencia que sea igual al valor base corresponde a $0 dB$. De esta forma el umbral del sonido y el umbral de dolor corresponden a $0 dB$ y $120 dB$ respectivamente, a $1000 Hz$.

En la Figura 2.14 puede verse como la sonoridad percibida depende de estas dos medidas. La sonoridad está medida en fonios.

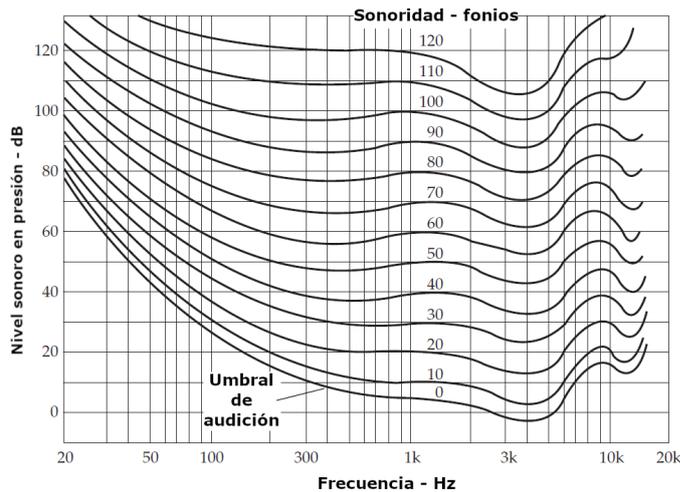


Figura 2.14: Gráfica de la sonoridad en función de la frecuencia y el nivel sonoro. Notar que para una frecuencia de $1 KHz$ la sonoridad y el nivel sonoro coinciden. Gráfico tomado de [25].

2.1.8. Respuesta al impulso

Con los conceptos y definiciones explicados hasta ahora es posible definir lo que es una respuesta al impulso. Cuando una fuente emite sonido en una escena, las ondas de sonido emitidas se propagan experimentando todos los fenómenos explicados en la Sección 2.1.4 hasta que finalmente llegan al receptor. Por lo tanto, lo que es escuchado por el receptor no solo consiste en la excitación del medio producido por la fuente, también incluye la respuesta de la escena. Esta última incluye reflexiones atenuadas y retrasadas [41].

Si la señal original consiste de un único pulso “instantáneo”, y se utiliza un micrófono como receptor, la respuesta medida es lo que se conoce como **Respuesta al Impulso** o **IR** por sus siglas en inglés. Si la escena consiste de un cuarto cerrado la IR también puede escribirse como **RIR** por sus siglas en inglés *Room Impulse Response*.

Esta medida es muy útil ya que da cuenta de la energía que llega al receptor en cada instante de tiempo. Esta energía puede expresarse en intensidad o con el cuadrado de la presión eficaz.

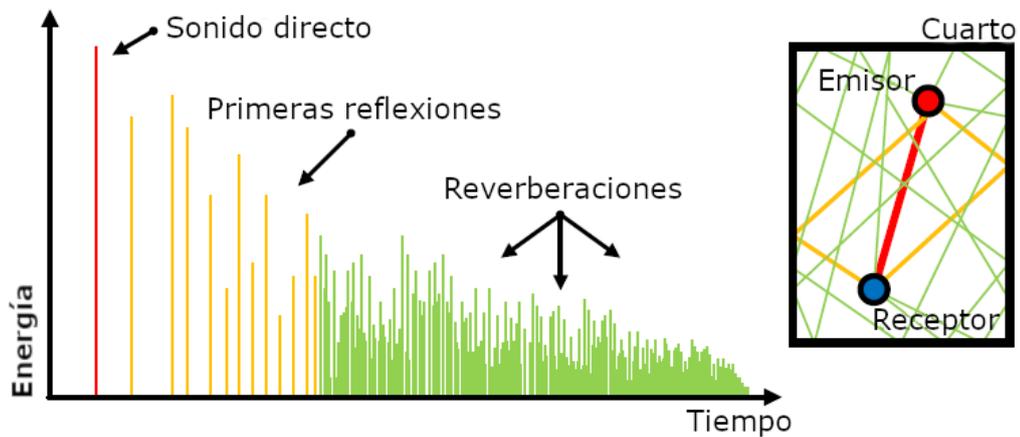


Figura 2.15: Respuesta al impulso de un cuarto. Gráfico tomado de [41].

Como puede verse en la Figura 2.15, en la IR normalmente se distinguen 3 componentes [41]:

1. Sonido directo: Corresponde al impulso que llega directamente al receptor.
2. Primeras reflexiones: Consiste en el impulso resultante de las reflexiones

con la escena. El receptor no es capaz de distinguir entre estas y el sonido directo.

3. Reverberaciones: Se trata de un impulso percibido entre 50 y 80 *ms* luego del sonido directo. Las reverberaciones son las que mejor describen las propiedades acústicas de una escena. Están directamente asociadas con la forma y el volumen de un cuarto.

Dado un recinto y dos coordenadas que definen la ubicación del emisor y el receptor, se define una respuesta al impulso o IR. A su vez, una vez obtenida la IR, es posible simular cómo un sonido generado por el emisor es percibido por el receptor. Para esto, se modela a la interacción entre los tres elementos (recinto, emisor, receptor) como un sistema lineal invariante en el tiempo (o SLIT) [43][50], donde a partir de una señal de entrada (emitida por el emisor) y una IR generada por esos tres elementos, se puede calcular la señal de salida (recibida por el receptor) del sistema. La operación por la que se obtiene la salida se llama “convolución”, y se estudia con mayor detalle en la Sección 4.3. Esa salida es utilizada para la auralización, concepto que se analiza en la Sección 2.4.

2.1.9. Tiempo de reverberación y EDT

El tiempo de reverberación es una medida de qué tan rápido decae la energía de un impulso. Se define como el tiempo necesario para que el nivel sonoro de presión disminuya 60 *dB* [29]. Conociendo la IR de la escena sería posible calcular el tiempo de reverberación, sin embargo, es difícil lograr una medición de un impulso que baje 60 *dB* en su nivel sonoro. Por ejemplo, en el caso de realizar una medición al aire libre, el ruido de fondo puede tapar la señal que se desea estudiar o simplemente la señal puede ser muy tenue como para que el micrófono capte esta disminución [25]. Por este motivo, el tiempo de reverberación suele estimarse a partir de otras medidas como se indica a continuación.

La **curva de decaimiento** o *decay curve* mide cómo disminuye el nivel sonoro en función del tiempo, como puede verse en la Figura 2.16.

Es posible utilizar distintas porciones de esta curva para obtener estimaciones del tiempo de reverberación. Con estas porciones de la curva es posible deducir una recta mediante regresión lineal como puede verse en la Figura

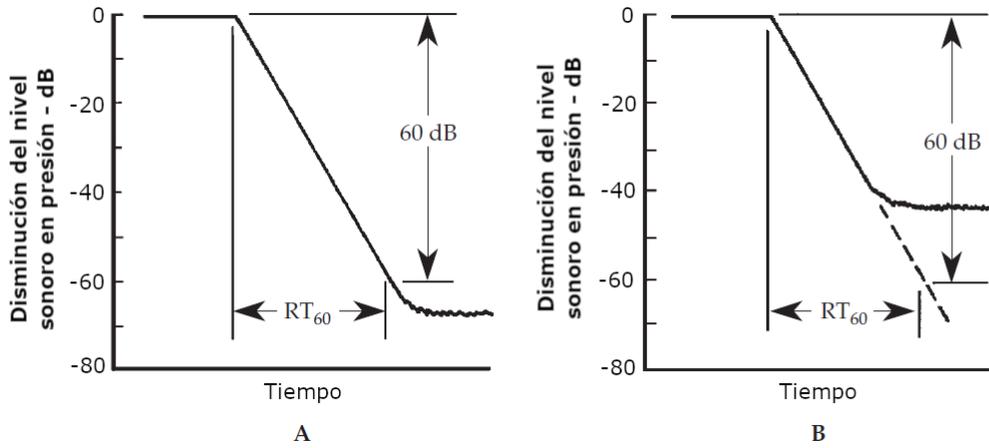


Figura 2.16: Ejemplos de curvas de decaimiento. (A) Se puede registrar una disminución de 60 *dB* en el nivel sonoro, el tiempo de reverberación se puede medir directamente. (B) El nivel de la señal cae por debajo del ruido de fondo antes de registrarse la disminución de 60 *dB* y por consiguiente el tiempo de reverberación se estima a partir de la pendiente de la curva. Gráfico tomado de [25].

2.17. Esta pendiente de recta d , representa la velocidad con la que el nivel sonoro decae y se expresa en $\frac{dB}{s}$ [20]. A partir de esta pendiente se desprende la ecuación 2.9 con la cual puede estimarse el tiempo de reverberación.

$$d \cdot t_{60} = -60 \longrightarrow t_{60} = \frac{-60}{d} \quad (2.9)$$

Donde t es el tiempo necesario para que el nivel sonoro baje 60 *dB*.

Se denota T_{30} a la estimación que proviene de tomar la porción de la curva comprendida entre los -5 *dB* y los -35 *dB*, análogamente T_{20} y T_{10} se obtienen de las regiones de -5 *dB* a -25 *dB* y de -5 *dB* a -15 *dB* respectivamente [20].

Otra forma similar de estimar el tiempo de reverberación, es medir t_{10} , t_{20} y t_{30} , los tiempos que demora la señal en bajar 10, 20 y 30 *dB* respectivamente y estimar de forma lineal el tiempo de reverberación como $T_{10} = 6 \cdot t_{10}$, $T_{20} = 3 \cdot t_{20}$ y $T_{30} = 2 \cdot t_{30}$

Una posible forma de obtener la curva de decaimiento es a través de la integral de la Respuesta al impulso (IR) [40] (recordar que la IR no es otra cosa que la energía acústica en función del tiempo, ver Figura 2.15). Sin embargo, también es posible obtenerla calculando el nivel sonoro a partir de la Ecuación 2.6, utilizando como valor base la suma de todos los valores de la IR. Como puede verse en la Ecuación 2.10, el nivel sonoro en el tiempo t se calcula con

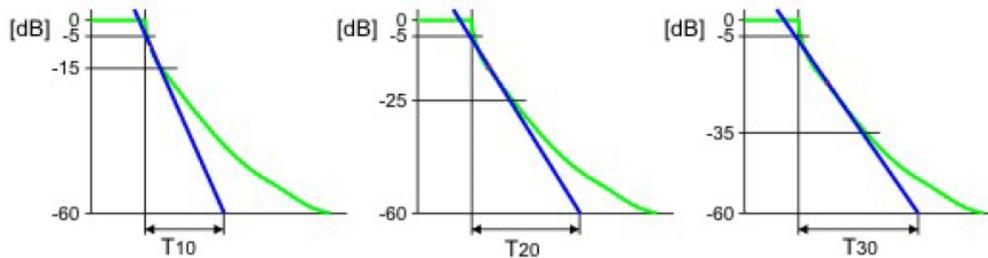


Figura 2.17: El tiempo de reverberación puede estimarse a partir de distintas secciones de la curva de decaimiento. Gráfico tomado de [9].

los valores de la IR desde el tiempo t en adelante. De esta forma el nivel sonoro disminuye a medida que el tiempo avanza.

$$D(t) = 10 \log \frac{\sum_{i=t}^n IR_i}{\sum_{i=1}^n IR_i} \quad (2.10)$$

Donde n es la cantidad de valores registrados de la IR e IR_i el i -ésimo de estos valores.

El **Early Decay Time** o **EDT** es otra de las posibles aproximaciones y se deriva de la región de la curva comprendida desde el inicio hasta que baja 10 dB . Se considera que esta estimación mide el tiempo de reverberación percibido y contempla solo a las reflexiones tempranas. El EDT es un mejor estimador del tiempo de reverberación para fuentes continuas ya que en el caso de estas, las reverberaciones son tapadas por las reflexiones tempranas de la onda en un tiempo posterior.

2.2. Representación digital del sonido

Hasta este momento se han definido las características, propiedades y modelos del sonido como fenómeno físico. Sin embargo, es imperativo desde un punto de vista computacional, estudiar cómo el sonido es transformado en una señal digital y cómo esta es almacenada dentro de una computadora, en forma de bits.

El proceso de registrar una señal de audio comienza a partir de un micrófono. Los micrófonos son capaces de detectar las variaciones de presión en el aire. Estas variaciones son transformadas en voltaje que luego es enviado a un convertor analógico/digital para ser transformada en bits y finalmente ser almacenados en una computadora [15]. En este trabajo, el enfoque está en

comprender cómo está formada la señal digital y cómo es posible procesarla. Los detalles de cómo los micrófonos transforman presión en voltaje y cómo esta se transforma a bits quedan a cargo del lector, sin embargo, no es necesario comprender estos detalles para comprender el resto de este trabajo.

En esta sección se presentan algunos de los conceptos más importantes que conciernen al procesamiento del audio como una señal digital.

2.2.1. Frecuencia de muestreo

Para poder convertir una señal real a una digital es necesario discretizarla. La **frecuencia de muestreo** o *sample rate* refiere a la frecuencia con la que se miden los valores de la presión efectiva de la señal real. Cada uno de estos valores se llama **muestra** o *sample* y representan el valor de desplazamiento de la onda en ese instante de tiempo. Normalmente se consideran la cantidad de *samples* medidos en 1 segundo. En este caso el *sample rate* se mide en Hertz (*Hz*). Por ejemplo, para la producción de audio profesional se utiliza un *sample rate* estandar de 48.000 *Hz*, es decir, los valores de la presión efectiva se miden 48.000 veces por segundo.

Es fácil ver que un *sample rate* mayor mejora la calidad de la señal digital. Al subdividir el tiempo en más intervalos la onda de la señal digital se asemeja más a la onda real (ver Figura 2.18). En este sentido, un concepto importante es el teorema de muestreo de Nyquist-Shannon [43], que establece que una señal continua puede ser correctamente *sampleada*, sólo si no contiene frecuencias superiores a la mitad del *sample rate*. Por ejemplo, un *sample rate* de 2000 muestras por segundo requiere que la señal analógica esté compuesta por frecuencias inferiores a los 1000 *Hz*.

Sin embargo, un *sample rate* mayor también implica un mayor uso de memoria para almacenar la señal y tiempo mayor para procesarla [15]. Los estándares usados para el *sample rate* se determinan a partir de las características necesarias para la señal dependiendo del uso que tengan. Por ejemplo, si bien en la producción musical se utiliza un *sample rate* de 48.000 *Hz*, para las comunicaciones telefónicas se utiliza un *sample rate* de 8000 *Hz* que es suficiente para lograr la inteligibilidad de la voz de un ser humano.

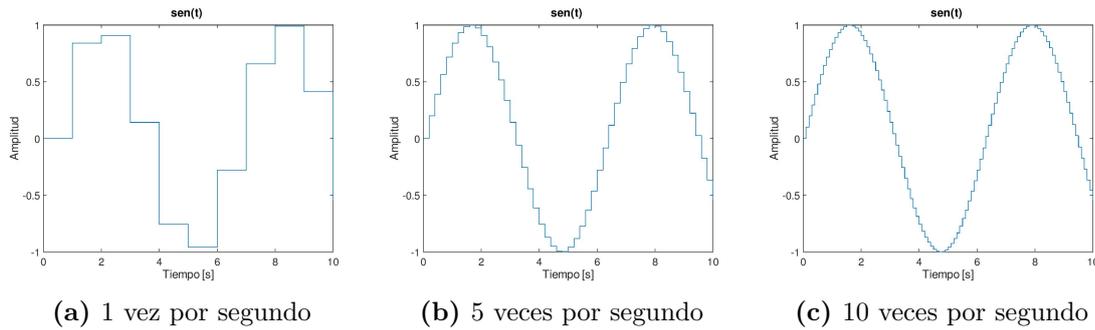


Figura 2.18: Gráfica de la función $\text{sen}(t)$ donde t se mide en segundos. La función es *sampleada* con distintas frecuencias. A medida que aumenta el *sample rate* las gráficas se parecen más a la gráfica de la función $\text{sen}(x)$.

2.2.2. Bit depth

Así como es necesario discretizar una señal continua en el tiempo, también es necesario pasar de valores continuos para la amplitud a valores discretos representables con una resolución finita. [15]. El *bit depth* refiere a la cantidad de bits utilizados para representar cada *sample*. En este caso corresponde el mismo análisis que para el *sample rate*. Con un *bit depth* mayor, es posible lograr una mayor resolución en las amplitudes de onda registradas, pero también implica un mayor uso de memoria.

El procesamiento de una señal digital de audio puede incluir operaciones que requieran cierta precisión. En algunos casos puede utilizarse una precisión de 32 bits para el procesamiento y luego exportar el resultado en una precisión menor de 24 o 16 bits.

A este proceso de pasar un valor de amplitud a su valor más cercano en la representación elegida se le suele llamar **cuantización** (*quantization*). La cuantización agrega ruido y distorsión a una señal, cuanto mayor es el *bit depth* estos efectos son menos nocivos.

2.2.3. Canales de audio

Los archivos de audio pueden contener datos para uno o más canales. Tanto los dispositivos de entrada como los de salida pueden tener más de un canal por donde ser detecte o se reproduzca audio. Por lo general, los micrófonos graban 1 solo canal de audio, mientras que los dispositivos de reproducción, como audífonos, reproducen audio por 2 altavoces distintos para lograr el sonido estéreo. Cada una de estas entradas o salidas se corresponde con un *buffer*

de *samples*. El contenido de los *buffers* de cada canal no tiene que ser distinto necesariamente. Para reproducir una señal en estéreo a partir de una grabación monofónica de un micrófono, alcanza con reproducir la misma señal grabada por ambos canales.

2.2.4. Adimensionalidad

Si bien los valores que se registran en un archivo de audio provienen de las variaciones de presión en el aire, los valores finales no representan valores de presión, es decir, no están medidos en pascales. Esto se debe a que el pasaje de presión a voltios y posteriormente a bits ocurren transformaciones que dependen de las características y calibraciones del micrófono. Por ejemplo, cada micrófono produce un voltaje distinto como respuesta a una cierta presión recibida. A esta propiedad de los micrófonos se le llama sensibilidad o *sensitivity* [15].

Los valores medidos entonces, no tienen validez como medidas de presión. Sin embargo, si se considera que el ratio entre dos valores reales de presión es el mismo al de sus correspondientes valores medidos, entonces las diferencias en decibelios entre ambos pares de valores se mantiene constante.

Una muestra o *sample* puede considerarse como se ve en la Ecuación 2.11. Por conveniencia, se decide normalizar el valor medido.

$$A(t) = s(t) \times k \tag{2.11}$$

Donde:

- t es el tiempo discretizado en los intervalos determinados por la frecuencia de muestreo.
- $A(t)$ es la amplitud real de la onda en el tiempo t .
- $s(t)$ es el valor del *sample* en el tiempo t , es decir, el valor medido en el tiempo t .
- k una constante que depende de las características del micrófono.

Luego, si se calcula el nivel sonoro en presión para un valor de presión real, se obtiene:

$$L_p(t) = 10 \log \frac{s(t) \cdot k}{(2 \cdot 10^{-5})^2} = 10(\log \frac{s(t)}{(2 \cdot 10^{-5})^2} + \log k) = 10 \log \frac{s(t)}{(2 \cdot 10^{-5})^2} + C \quad (2.12)$$

Donde C es una nueva contante que depende de las características del micrófono. Entonces, se puede apreciar que la diferencia entre el nivel sonoro entre 2 valores reales de presión coincide con la diferencia del nivel sonoro de 2 *samples*, debido a que las constantes C se anulan.

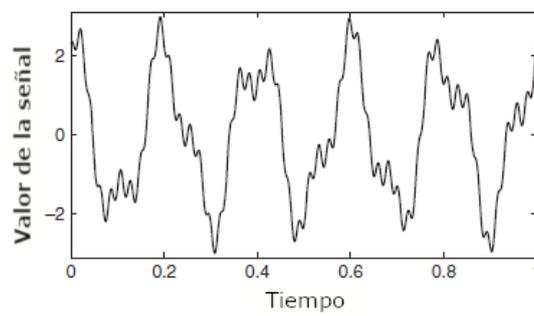
$$\begin{aligned} L_p(t_1) - L_p(t_0) &= 10 \log \frac{s(t_1)}{(2 \cdot 10^{-5})^2} + C - (10 \log \frac{s(t_0)}{(2 \cdot 10^{-5})^2} + C) \\ &= 10 \log \frac{s(t_1)}{(2 \cdot 10^{-5})^2} - 10 \log \frac{s(t_0)}{(2 \cdot 10^{-5})^2} \\ &= L'_p(t_1) - L'_p(t_0) \end{aligned} \quad (2.13)$$

Donde L_p es el nivel sonoro del valor real de la onda y L'_p el nivel sonoro del valor medido, es decir, el nivel sonoro del *sample*.

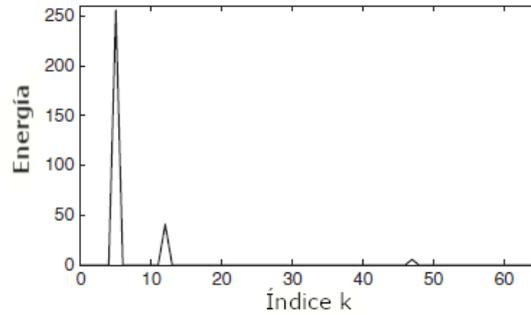
2.3. Transformada de Fourier

En lo expuesto hasta este momento se ha hablado del sonido como una función de la amplitud en el tiempo. Esta forma de expresar el sonido se conoce como una función de dominio en el tiempo y es la forma utilizada para representarlo en los usos más comunes como se mencionó anteriormente. Sin embargo, como también fue mencionado, el sonido también puede expresarse como una combinación de ondas de distintas frecuencias. Es posible entonces, expresar al sonido como una función que indica cuanto contribuye cada frecuencia a la señal original, es decir, como una función de dominio en frecuencias. En la Figura 2.19 puede verse un ejemplo en ambos dominios.

La transformada de Fourier permite hallar el dominio en frecuencias a partir del dominio en el tiempo y viceversa utilizando su inversa. Para lograr esto se hallan las ondas de distintas frecuencias que componen a la onda original [38]. Para una explicación más detallada de cómo funciona la transformada de Fourier, se refiere al lector a la bibliografía citada.



(a) Dominio en el tiempo.



(b) Dominio en frecuencias.

Figura 2.19: Ejemplo de una señal expresada en dominio de tiempo y dominio de frecuencias. La señal está compuesta por ondas de frecuencias de 5, 12 y 47 *Hz*. Gráfico tomado de [38].

2.4. Simulación vs Auralización

En este trabajo se exploran 2 problemas distintos; **Simulación** y **Auralización**.

En el problema de la simulación se intenta simular la propagación del sonido en una escena de interés. Dadas ciertas condiciones iniciales como propiedades de la fuente, posición del oyente y las propiedades de la escena en cuestión, es posible obtener una simulación de la IR de la escena. Luego, esta IR puede compararse con mediciones reales en escenas con las mismas características simuladas.

La auralización refiere al proceso de hacer audible, a través de un modelo físico o matemático, el sonido de una fuente en un espacio, ya sea arquitectónico o natural, de forma tal que se simule la experiencia de escucha en una posición dada en el espacio modelado [25] [15]. El proceso de la auralización entonces, consiste en tomar el resultado de la simulación para hacer la convolución de esta y una señal de audio elegida. Con este resultado es posible comprender cómo se percibiría dicha señal dentro de la escena en cuestión.

Estos 2 problemas están claramente relacionados y el término “auralización” puede utilizarse para referirse al proceso que involucra a ambos problemas [25]. Sin embargo, los resultados de estos problemas pueden ser evaluados de forma independiente. Por ejemplo, distintas soluciones para la auralización puede evaluarse a partir de un IR dado. Además, los resultados de la simulación pueden ser cuantificados, lo que da lugar a un análisis cuantitativo de estos comparándolos con una medición. Por otro lado, los resultados de la auralización son subjetivos ya que refieren a cómo una persona experimenta un sonido en cuestión. Como se mencionó en la Sección 2.1.6, el sonido percibido depende entre otras cosas, de la HRTF, que es única para cada persona. Corresponde entonces, que el análisis de estos resultados sea cualitativo. Por estas razones, en este trabajo se utilizan los términos simulación y auralización para referirse a problemas distintos.

2.5. Soluciones existentes a la simulación del sonido

A continuación se presentan algunas soluciones para el problema de la simulación del sonido utilizando trazado de rayos.

2.5.1. VRWorks Audio

VRWorks Audio es el SDK de NVIDIA para la simulación y auralización del sonido [4]. Se trata de una biblioteca en C para el desarrollo de aplicaciones que requieren audio inmersivo. Las operaciones correspondientes a la simulación se hacen en GPU y están optimizadas para el hardware RTX de NVIDIA. El uso de GPU acelera los cálculos de la simulación, sin embargo esto limita su uso a entornos de desarrollo que posean GPUs.

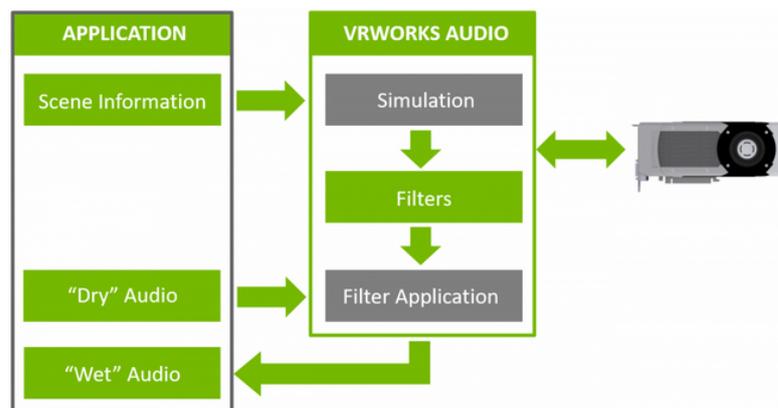


Figura 2.20: Esquema del SDK de VRWorks Audio. Gráfico tomado de [4].

Esta biblioteca modela la reflexión difusa y especular, la transmisión y provee una aproximación de la difracción para el sonido directo. La escena se puede construir utilizando varios *meshes* con distintos coeficientes de absorción y transmisión, y puede constar de varias fuentes de sonido. Ésta puede ser modificada dinámicamente, lo que incluye agregar fuentes de sonido, y modificar su posición y orientación, lo mismo para el oyente. También permite mover la geometría de la escena.

El resultado de la simulación son filtros que luego pueden aplicarse a una

señal de audio para lograr la auralización. Los filtros obtenidos tienen un *sample rate* de hasta 48.000 *Hz* y una duración de hasta 2 segundos.

Detalles sobre la implementación de VRWorks Audio no están disponibles.

2.5.2. Wayverb

Wayverb es una aplicación que permite simular la IR de un recinto a partir de un modelo CAD [48]. Es una aplicación hecha en C++ y acelerada por GPU. Funciona sobre MacOS. Los cálculos de la simulación se hacen *offline*, es decir, no se hacen en tiempo real y la aplicación no provee la funcionalidad de aplicar el IR obtenido a una señal de audio para obtener la auralización.

Wayverb provee una interfaz gráfica para visualizar la escena junto con algunas configuraciones.

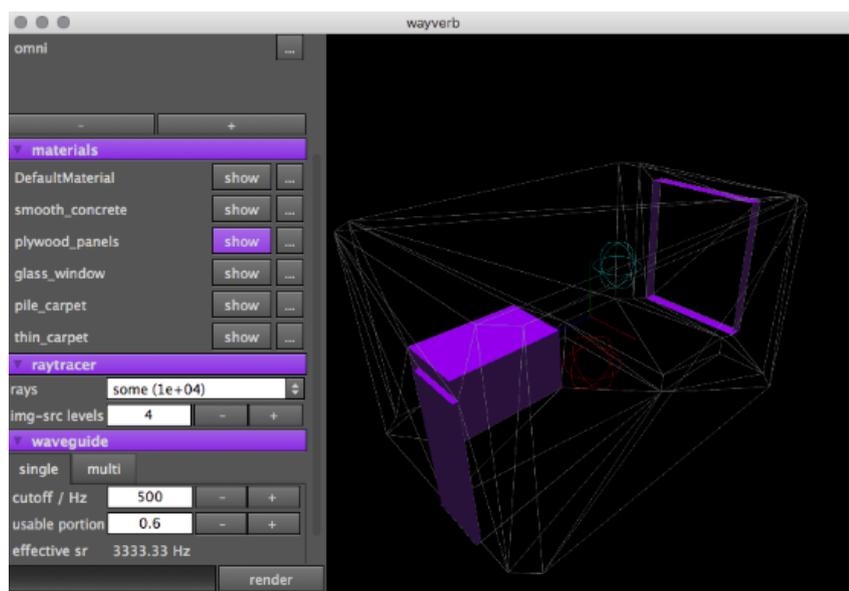


Figura 2.21: Visualización de la escena en Wayverb. Gráfico tomado de [48].

Si bien no se describen explícitamente las prestaciones de esta aplicación el código fuente está disponible y el repositorio del proyecto cuenta con una explicación detallada de la implementación.

2.6. Bibliotecas útiles

En esta sección se presentan algunas bibliotecas que resuelven problemas de programación relacionados al procesamiento del sonido y la traza de ra-

yos. Estas bibliotecas son utilizadas en la solución propuesta presentada en el Capítulo 3.

2.6.1. Embree

Embree es la biblioteca para el trazado de rayos de Intel [7]. Se trata de una biblioteca en C++ con soporte para las plataformas Windows, Linux y MacOS. Embree realiza las operaciones para la traza de rayos en CPU y soporta ejecuciones en multiples hilos, por lo tanto las aplicaciones desarrolladas con embree no necesitan correr en un entorno con GPU.

Embree provee una API (*Application Programming Interface*) que simplifica el trazado de rayos. Mediante el uso de esta API se puede cargar una escena especificando su geometría (vértices e índices) y las primitivas a utilizar (triángulos, cuadrados, etc). Una vez cargada la escena, otras funciones de la API realizan la intersección entre esta y un rayo con ciertas características.

Cabe mencionar que Embree es una biblioteca mucho más completa de lo que se detalla aquí. Sin embargo, uno de los atractivos de esta biblioteca es la simplicidad de su uso, dado que es posible valerse de esta herramienta sin tener conocimiento de algunos temas complejos. Además, el uso de esta biblioteca en este proyecto se limita a estas funcionalidades más básicas, por lo que se deja la investigación de los aspectos más avanzados al lector. Algunas de las prestaciones más avanzadas de Embree son: Primitivas de geometría para cabello, deformación de *meshes* para *motion blur*, instanciación, entre otras.

Embree es un proyecto *open source*.

2.6.2. OpenGL

OpenGL es una biblioteca gráfica diseñada por la organización Khronos Group [10]. El cometido de este tipo de bibliotecas es proporcionar una API para la utilización de la rasterización en distintos sistemas operativos y sobre distinto hardware (por ej. tarjetas gráficas). En otras palabras, se trata de una biblioteca que permite el dibujados de escenas en 2 o 3 dimensiones.

Si bien no es necesaria para lograr la simulación o auralización del sonido en una escena, la visualización es imprescindible para algunos casos de uso como es el caso de los videojuegos. Además la visualización permite la exploración de la escena, lo que puede simplificar las pruebas dentro de la misma.

2.6.3. RtAudio

RtAudio es una biblioteca que provee una API para el uso de operaciones de entrada y salida de audio en varias plataformas. Fue desarrollada por Gary P. Scavone en la Universidad McGill en 2001 y es mantenida hasta el día de hoy [39]. El propósito de una biblioteca de audio es simplificar el uso del hardware de audio de una computadora, permitiendo así que el programa desarrollado cuente con la capacidad de grabar y reproducir sonido.

RtAudio permite acceder a los datos obtenidos de un micrófono en tiempo real de forma programática. Es decir, permite obtener los *samples* de audio. Esto simplifica mucho el procesamiento del audio en tiempo real. También permite definir el *sample rate* y el tipo de datos utilizados para los *samples*.

2.7. Trabajos relacionados

El trazado de rayos aplicado a la simulación de la propagación del sonido es estudiado desde hace varios años y las aplicaciones de este método han variado a lo largo de los años junto con el avance de la tecnología. Un breve repaso en orden cronológico de estos trabajos puede encontrarse en [34]. En esta sección se detallan algunos de estos trabajos.

En 1968 Krokstad [32] aplicó la traza de rayos para simular la distribución en el tiempo y el espacio del sonido en salones de concierto. Al calcular la intensidad y la demora para distintos puntos en el público es posible evaluar como será el sonido percibido y si la distribución lograda es la deseada. El modelado en este trabajo cuenta con varias limitaciones, algunas de las cuales son aceptadas hasta el día de hoy, como por ejemplo, el rango de frecuencias para las que son válidos los resultados. En este trabajo se evidencia la utilidad de este método para ayudar en tareas como el diseño de estos espacios.

Posteriormente en 1985, Kulowski [33] presenta un trabajo con un enfoque en la eficiencia del método. Si bien muchas de las limitaciones de Krokstad están presentes en este trabajo (por ejemplo, tener un solo coeficiente de absorción, no contemplar la diferencia entre sonidos de distintas frecuencias, escenas descritas a partir de ecuaciones de planos, entre otras), esta vez se presenta una descripción de la implementación. Se presenta que los tiempos de computo para la mayoría de las escenas estudiadas no superó los 15 minutos y para las escenas más complejas no superó los 180 minutos. Se presenta que,

en general, el número de rayos utilizados es de algunos cientos, siendo 2000 la mayor cantidad documentada. La cantidad de rebotes máxima registrada fue de 7.

En 2010, Alpkocak y Sis [13] presentan un enfoque para calcular la respuesta al impulso de un cuarto utilizando el trazado de rayos. Este enfoque se basa en modelar al cuarto como un sistema lineal invariante en el tiempo (*linear time-invariant system*) que recibe como entrada un pulso de Dirac. Luego, como cualquier señal puede ser construida a partir de una sumatoria de pulsos, la IR puede construirse a partir de las IRs de dichos pulsos. El modelo contempla las bandas de octavas construyendo la IR como la unión de las IR de cada banda. Para validar este modelo se desarrolló un programa de computadora llamado DAAD (Dokuz Eylul Architectural Acoustic Design) que permite visualizar y editar el cuarto donde se desea realizar la simulación y realizar la auralización de una señal de audio provista por el usuario. El programa permite realizar la simulación con más de 1 fuente y varios coeficientes de absorción distintos. Los resultados de este programa se evaluaron comparando el T_{30} y el *Early Decay Time* con las mediciones correspondientes. Se registraron errores de entre 0,10 y 0,75 segundos. No se hace mención sobre tiempos de ejecución ni la cantidad de rayos utilizada.

Otros trabajos como los de Mo, Yeh, Lin y Manocha [35] en 2016 amplían su modelo de trazado de rayos para simular las trayectorias curvilíneas de los rayos en escenas abiertas.

Capítulo 3

Solución propuesta

En esta sección se explora el diseño de la solución propuesta. Se detalla el proceso de toma de decisiones y se explican brevemente diseños alternativos explorados. También se detallan todos los componentes de la solución final y como estos interactúan entre sí.

3.1. Alcance y objetivos

El propósito principal de este trabajo es entender todas las etapas del proceso de simulación y auralización del sonido. El objetivo primario es lograr la implementación de una solución de principio a fin que resuelva ambos problemas en cierta medida. Es decir, a partir de una escena y sus características, se busca obtener la Respuesta al Impulso (IR) de la misma para posteriormente realizar la convolución de esta con una señal de audio.

En este trabajo no se busca una solución total del problema donde todos los aspectos de la simulación estén resueltos. Por ejemplo, no se busca modelar todos los aspectos de la propagación de sonido, ni todos los aspectos del emisor y del receptor. Además, las escenas estudiadas se limitan a habitaciones o lugares al aire libre, por lo tanto, no se contempla la propagación del sonido en otros medios como el agua.

3.2. Diseño

La solución propuesta sirve como un primer acercamiento a los problemas de la simulación y auralización. A continuación se detallan sus propiedades:

- Permite cargar la escena a través de un archivo de configuración. Las escenas están limitadas a una fuente, un oyente y un coeficiente de absorción para toda la escena.
- El archivo de configuración también permite editar los parámetros de la simulación como el *sample rate* y la cantidad de rayos a emitir.
- Provee 2 modos de ejecución. Es posible realizar solamente la simulación para obtener la IR o utilizar este resultado para experimentar la auralización en tiempo real.
- El modo de simulación permite especificar un archivo de audio para que la IR resultante tenga los mismos valores de *sample rate*, tipo de datos y duración.
- El modo de simulación permite especificar un intervalo de tiempo para registrar los rayos que llegan en dicho instante.
- Permite realizar la auralización de una señal en vivo desde un micrófono o una señal grabada desde un archivo. No se implementa la HRTF, en su lugar para lograr el sonido estéreo se reproduce el resultado de la auralización en ambos canales.
- En el modo de auralización, la simulación se hace *offline*, es decir, se hace una vez al principio y luego a pedido del usuario.
- Permite visualizar la escena de forma gráfica. Las posiciones de la fuente y del oyente pueden modificarse de forma interactiva.
- La simulación de la propagación considera solo las reflexiones especulares. El coeficiente de absorción es el mismo para todas las frecuencias del sonido.
- La aplicación está compuesta por varios módulos que interactúan entre sí.

3.2.1. Módulo principal

Es el módulo de entrada al programa, en él se inicializan el resto de los módulos. El bucle principal del programa tiene lugar aquí y en este se realizan las llamadas a las operaciones de dibujado y de operaciones de audio. También maneja el input del usuario y el movimiento de la cámara. Ver Figura 3.1.

El programa se ejecuta a través de línea de comandos especificando 2 parámetros:

.\AudioRendering.exe {modo de ejecución} {archivo de configuración}

- El modo de ejecución puede ser *auralize* o *simulate*
- El segundo parámetro especifica la ruta del archivo de configuración.

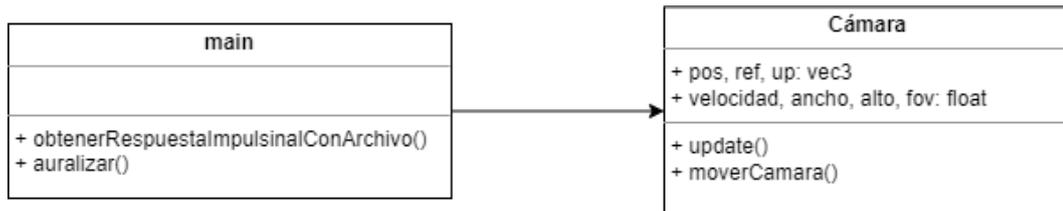


Figura 3.1: Módulo principal.

El archivo de configuración es un archivo XML [1] donde se especifican los parámetros de la escena como modelo CAD, coeficiente de absorción, tamaño, entre otros. Además los archivos XML también incluyen información sobre la fuente de sonido (como potencia y posición) y del oyente (posición y tamaño). En el apéndice A puede verse un ejemplo de este archivo.

3.2.2. Módulo de geometría

El módulo de geometría se encarga de cargar y procesar la geometría de la escena. Esta geometría se obtiene a partir de un archivo OBJ [5] especificado en el archivo de configuración XML (ver Figura 3.2).

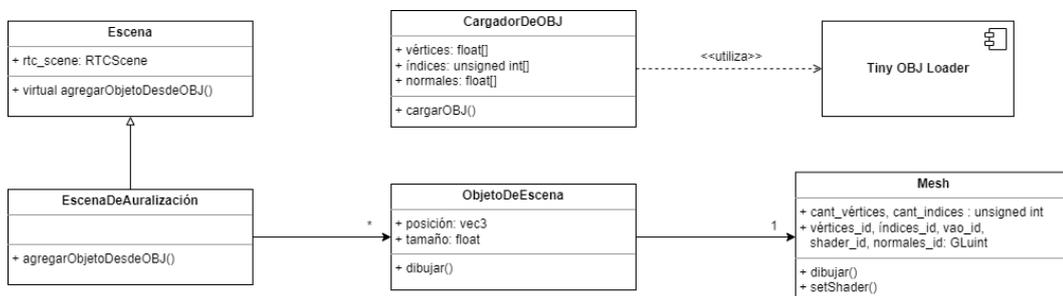


Figura 3.2: Módulo de geometría.

La clase **Escena** se encarga de crear la escena utilizada por Embree para realizar los cálculos. Esto lo hace transformando los datos obtenidos del cargador de archivos OBJ, al tipo de datos **RTCScene**.

Además, la escena creada necesita información adicional dependiendo del modo de ejecución elegido. Debido a que el modo de auralización dibuja la escena donde se realiza la simulación, es necesario crear los datos necesarios para el módulo de renderizado. Por este motivo, la clase **EscenaDeAuralización** expande la clase anterior para poder almacenar dichos datos. La clase **Mesh** almacena la geometría de cada objeto de la escena, mientras que **ObjetoDeEscena** se encarga de posicionar y escalar al *mesh* dentro de la escena. Como puede verse, en este caso se guardan 2 copias de la escena, una para el módulo de trazado de rayos y otra para el módulo de renderizado.

Para implementar la clase **CargadorDeOBJ** se utilizó la implementación provista por *Tiny OBJ Loader* [6], una implementación completa y muy sencilla de incluir en cualquier proyecto.

3.2.3. Módulo de render

El módulo de render está implementado en su totalidad por OpenGL. El proceso para el dibujado de un objeto comienza en el módulo principal, en éste se llama a las funciones para actualizar la cámara. Posteriormente se llama a las funciones del módulo de geometría para dibujar la escena cargada. Finalmente, el módulo de geometría llama a las APIs correspondientes del módulo de render para dibujar cada objeto de la escena (ver Figura 3.3).

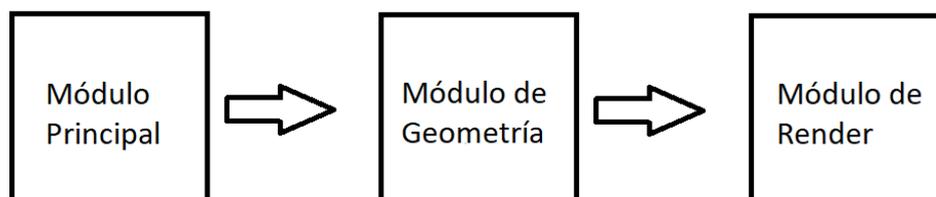


Figura 3.3: Cada módulo se encarga de pasar la información necesaria al módulo siguiente.

3.2.4. Módulo de trazado de caminos

El módulo de trazado de caminos se encarga de generar los rayos y llamar a las APIs de Embree correspondientes para calcular la intersección de estos con la geometría. La clase **TrazadorDeCaminos** es creada con los datos cargados del archivo de configuración necesarios para los cálculos. Además, implementa los algoritmos para la generación de las direcciones de los rayos (ver Figura 3.4).

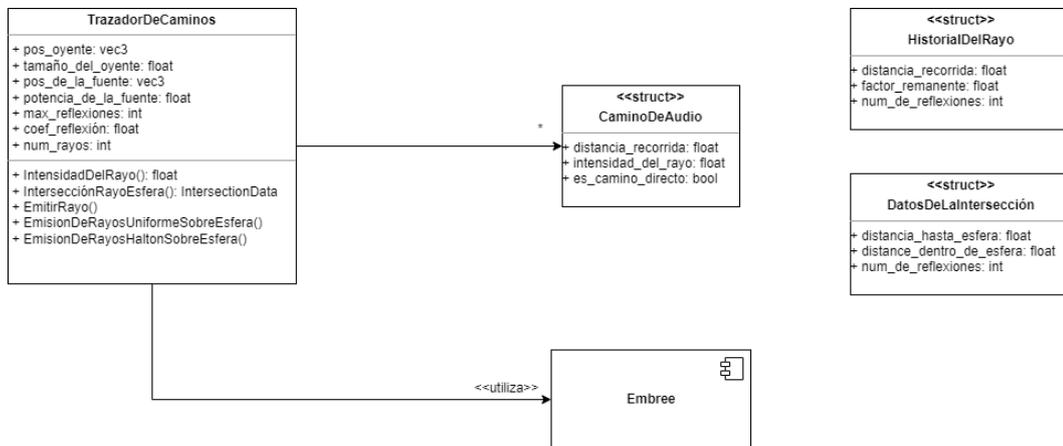


Figura 3.4: Módulo de traza de rayos.

Los tipos de datos **HistorialDelRayo** y **DatosDeLaIntersección** se usan internamente para guardar la información de cada rayo durante su propagación. El factor remanente refiere a la fracción de la energía inicial del rayo que aún se conserva. Una vez que el rayo llega al receptor, estos datos se utilizan para calcular el **CaminoDeAudio**. Al final de la propagación de todos los rayos, todos los caminos hallados hasta el receptor estarán almacenados, junto con su distancia recorrida e intensidad.

Dado que Embree solo puede devolver la distancia de la intersección en coordenadas de la escena, es imprescindible que los modelos de la escena estén en escala. Por ejemplo, puede considerarse que la unidad de la escena corresponde a 1 metro, centímetro, etc. Para este programa se considera que los modelos están en una escala de 1 metro por unidad, por lo tanto, la distancia de los caminos se mide en metros.

3.2.5. Módulo de audio

El módulo de audio provee las operaciones necesarias para el trazado de rayos y el procesamiento del audio al módulo principal (ver Figura 3.5).

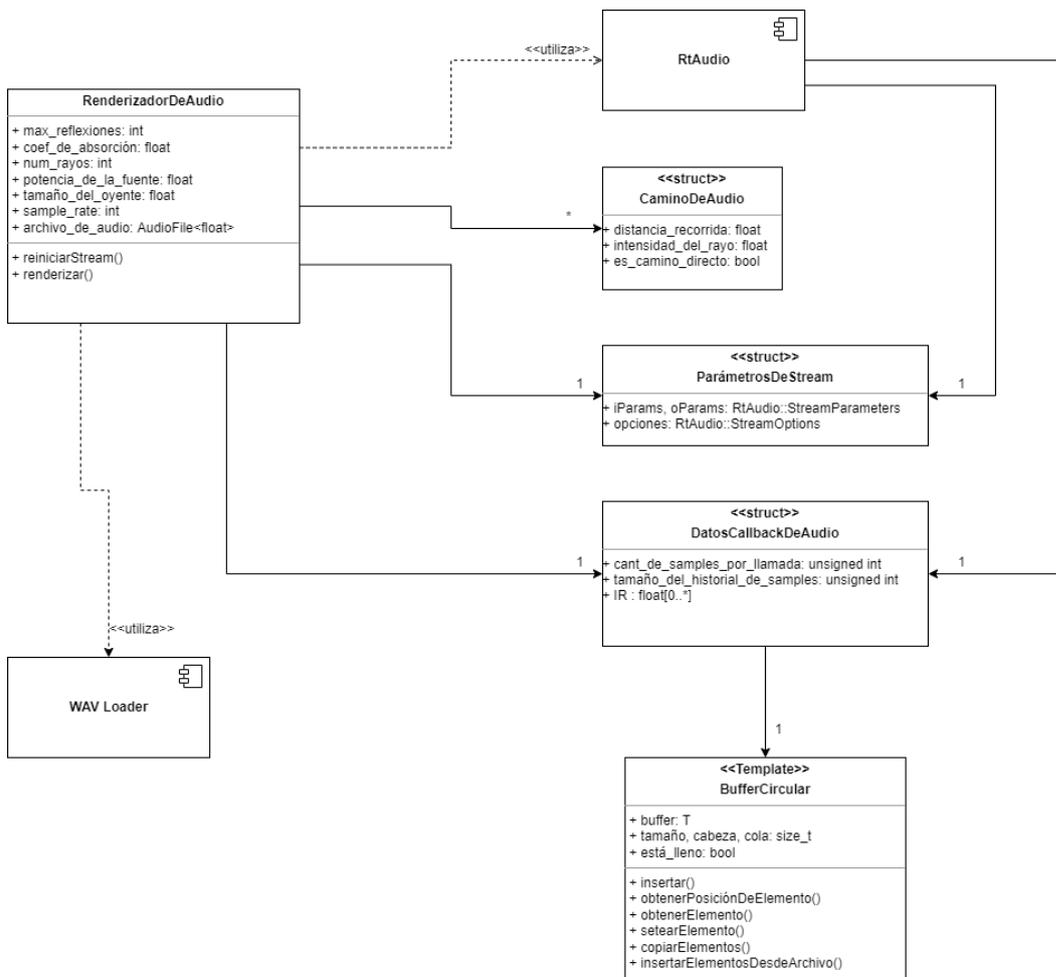


Figura 3.5: Módulo de audio.

La clase **RenderizadorDeAudio** se encarga de llamar al módulo de trazado de rayos a través de la función **renderizar**. Con los caminos obtenidos se realizan los cálculos correspondientes para obtener la IR de la escena (ver Sección 4.2).

En el modo de auralización, **RenderizadorDeAudio** se encarga de inicializar la biblioteca **RtAudio** y crear un *stream* de audio. Un *stream* es una clase de **RtAudio** que se encarga de tomar los *samples* de un dispositivo de entrada, procesarlos en una función de *callback* y enviarlos a un dispositivo de salida. Esta función se ejecuta en su propio hilo, por este motivo se cuenta con la clase

DatosCallbackDeAudio que se utiliza para compartir los datos para la auralización entre éste y el hilo principal. Cada llamada de la función procesa una cierta cantidad de samples determinada por *cantidad_de_samples_por_llamada*. En este caso, la función de *callback* realiza la convolución de la IR obtenida junto con la señal de audio entrante para generar el audio de salida.

La clase **BufferCircular** se usa para almacenar los *samples* de audio de tiempos anteriores. Para poder hacer la convolución de ambas señales, es necesario contar con un historial de los *samples* de la señal (ver Sección 4.3).

Finalmente, el modo de simulación utiliza una API para la carga de archivos de audio. A partir de los datos obtenidos por esta API es posible generar una IR con las mismas características. En este trabajo se utilizó el formato WAV [3] para archivos de audio. Por otro lado, en el modo de auralización, es posible elegir entre recibir una señal de audio desde un dispositivo de entrada (micrófono) o reproducir el audio de un archivo. En este último caso también se hace uso de esta API. Para la carga de archivos WAV se utiliza la implementación **AudioFile** [44].

Capítulo 4

Implementación

El siguiente capítulo desarrolla la implementación del diseño descrito en la Sección 3.2.

4.1. Traza de rayos

4.1.1. Generación de los rayos

Como se mencionó en la Sección 2.1.2 los rayos que modelan la onda de sonido irradian hacia afuera de la fuente de sonido. Por lo tanto, si la fuente se modela como una esfera, los rayos deben irradiar hacia afuera cubriendo toda la superficie de la misma. Es decir que, la generación de los rayos de sonido se corresponde con la generación de vectores con direcciones ortogonales a la superficie de la esfera, por toda la esfera (ver la Figura 4.1).

Debido a que la cantidad de rayos que es posible utilizar en un programa es acotada (más aún en el caso de una solución en tiempo real), la elección de estas direcciones es particularmente importante. Cuanto mejor sea la distribución lograda, menor será el impacto de tener pocos rayos. Existen varios algoritmos para generar direcciones en una esfera, en [26] puede encontrarse un estudio comparativo de algunas de ellas. En este trabajo se implementaron 2 alternativas; una distribución uniforme pseudoaleatoria y una distribución creada a partir de la secuencia cuasialeatoria de Halton.

Las secuencias pseudoaleatorias son construidas por un algoritmo a partir de una semilla. Son secuencias de números que no presentan ningún patrón o regularidad aparente desde un punto de vista estadístico. Por otro lado, las

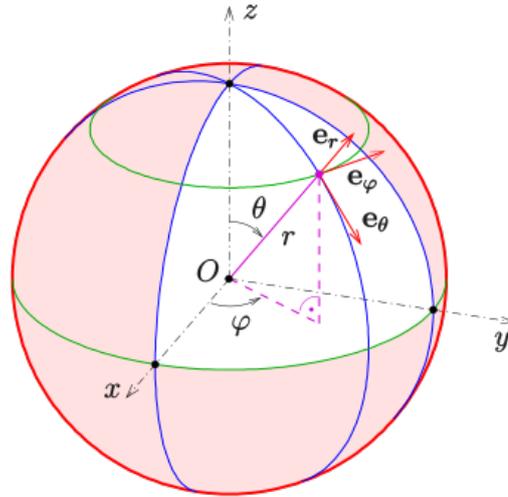


Figura 4.1: Vector unitario en coordenadas esféricas.

secuencias cuasialeatorias se construyen de forma que cada nuevo elemento esté lo más alejado del resto de la secuencia, logrando así una distribución relativamente uniforme [37].

Algoritmo 1 Algoritmo para la generación de rayos

```

function generarUniforme
  dirs = crearArray(cantidadRayos)
  loop i ∈ cantidadRayos
    theta = 2 · π · rand() //Ángulo theta en coordenadas esféricas.
    //rand() es la distribución uniforme en [0..1].
    phi = arcos(1 - 2 · rand()) //Ángulo phi en coordenadas esféricas
    dx = sen(phi) · cos(theta)
    dy = sen(phi) · sen(theta)
    dz = cos(phi)
    dir = normalizar(dx, dy, dz) //Dirección en coordenadas cartesianas
    dirs[i] = dir
  end loop
  return dirs
end function

```

En el Algoritmo 1 puede observarse cómo se generan las direcciones de los rayos. Los ángulos de las coordenadas esféricas se calculan utilizando una distribución pseudo aleatoria. En [23] puede encontrarse una explicación más detallada del algoritmo. Para generar la distribución uniforme se utilizó la función *uniform_real_distribution* junto con la secuencia pseudoaleatoria *mt19937* [11].

El algoritmo para generar el vector a partir de Halton es análogo, simplemente se utilizan los números de esa secuencia en lugar de la secuencia pseudoaleatoria. Para generar los números de Halton se utilizó la implementación disponible en [42].

4.1.2. Generación de Caminos

Como se mencionó brevemente en los Capítulos 1 y 2 el tiempo de propagación es importante al simular el sonido con traza de rayos. Esta es una de las principales diferencias con la traza de rayos para iluminación, ya que en esta se asume que la propagación de la luz es inmediata. Por este motivo, es necesario medir la distancia recorrida para cada rayo. En la Figura 4.2 puede observarse cómo esta información se actualiza en cada tramo de su recorrido.

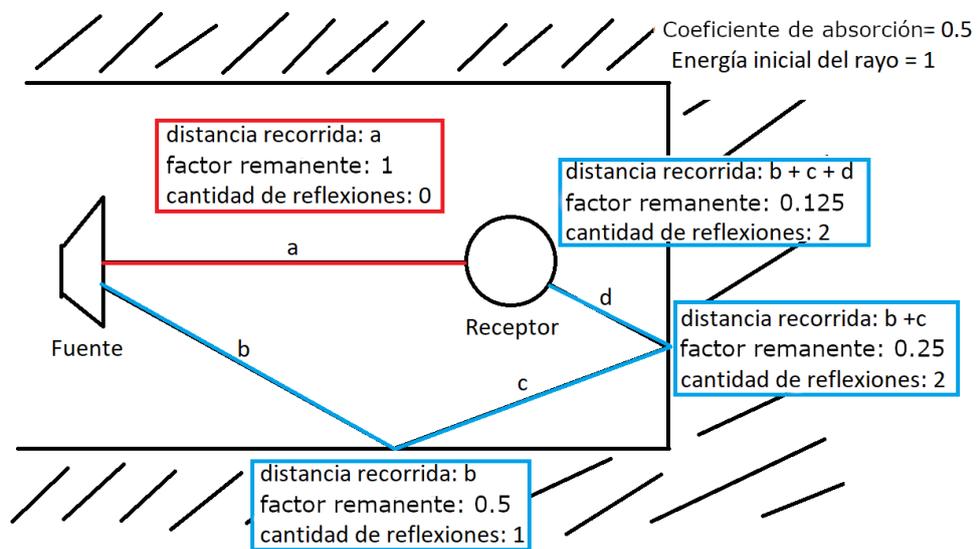


Figura 4.2: Propagación de los rayos de sonido. Los datos se actualizan en cada reflexión.

Puede verse cómo el rayo se propaga, hasta que se cumpla alguna de las condiciones de parada: El rayo llega al receptor, el rayo agota su cantidad máxima de reflexiones, o el rayo no colisiona con ningún objeto de la escena y se pierde en el infinito. Este algoritmo puede implementarse entonces con una función recursiva para la emisión de 1 rayo. Si el rayo se refleja, el *Historial-DelRayo* se actualiza y la función se vuelve a ejecutar, emitiendo así un rayo en la nueva dirección. Para cada uno de los rayos generados por el Algoritmo 1 se ejecuta el Algoritmo 2. Como puede verse, cuando uno de los rayos alcanza al receptor, se crea un nuevo *CaminoDeAudio* formado por todos los rayos emitidos en la recursión. Este camino es agregado a la lista de caminos encontrados que será utilizada posteriormente para el cálculo de la IR. La distancia de este camino es igual a la sumatoria de las distancias de todos sus rayos. El cálculo de la intensidad se explica en la Sección 4.1.3.

Algoritmo 2 Algoritmo para el cálculo de un camino

```
function emitirRayo(rayo, historial_del_rayo)
  intersecarRayoConEscena(rayo) //Utilizando Embree
  intersecarRayoConReceptor(rayo) //Verificar si hay intersección con el receptor
  if rayoIntersecaEscena then
    if rayoIntersecaReceptor then
      if distanciaAReceptor < distanciaAEscena then
        //El rayo llega al receptor
        camino = crearNuevoCamino(distanciaRecorrida(), calcularIntensidadDelRayo())
        return camino
      end if
    end if
  if cantidadDeReflexiones(historial) == max_reflexiones then
    return null
  end if
  //El rayo es reflejado. Se obtiene la dirección para el nuevo rayo reflejándolo respecto a
  //la normal en el punto de intersección.
  nuevo_rayo = reflejarRayoConEscena()
  //Se actualiza el factor remanente, la cantidad de reflexiones y la distancia recorrida.
  actualizarHistorial(historial_del_rayo)
  camino = emitirRayo(nuevo_rayo, historial_del_rayo)
  return camino
else
  if rayoIntersecaReceptor then
    //El rayo llega al receptor.
    camino = crearNuevoCamino(distanciaRecorrida(), calcularIntensidadDelRayo())
    return camino
  else
    return null
  end if
end if
end function
```

4.1.3. Cálculo de la intensidad

El cálculo de la intensidad de un *CaminoDeAudio* se realiza a partir de la Ecuación 2.3. Como este modelo no considera la absorción por el aire, la ecuación puede simplificarse. Por otro lado, el receptor se modela como una esfera, por lo que la ecuación para calcular la intensidad I puede escribirse como se ve en la Ecuación 4.1,

$$I = \frac{sE_0 \prod_i (1 - \alpha_i)}{\frac{4}{3} \cdot \pi \cdot r^3}, \quad (4.1)$$

donde r es el radio de la esfera que modela al oyente.

El campo *factor_remanente* del *HistorialDelRayo* se inicializa con la energía inicial E_0 para cada rayo como se mostró en la Sección 2.1.5. Por cada reflexión del rayo, el *factor_remanente* se multiplica por el coeficiente de reflexión $(1 - \alpha_i)$, de esta forma, el *factor_remanente* representa al término $E_0 \prod_i (1 - \alpha_i)$ de la Ecuación 4.1.

La distancia recorrida dentro de la esfera puede calcularse simplemente a partir de la intersección calculada en la propagación. El sistema de ecuaciones proveniente de la ecuación del rayo y de la esfera, tiene como solución 0, 1 o 2 puntos en el espacio. En el caso de haber 2 soluciones, la distancia recorrida es la distancia entre ambos puntos [2]. Esto puede verse de forma gráfica en la

Figura 4.3.

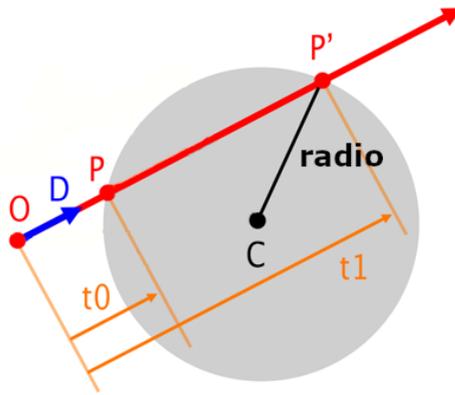


Figura 4.3: Intersección Rayo/Esfera. t_0 y t_1 , las soluciones del sistema, miden la distancia desde el origen del rayo a cada punto de intersección. Gráfico tomado de [2].

4.2. Construcción de la IR

Una vez hallados los caminos hasta el oyente y sus correspondientes intensidades, es posible generar la IR de la escena. Además de estos datos, un elemento imprescindible para el cálculo de la IR es el *sample rate* deseado. Cada camino contribuye a la IR en un tiempo dado correspondiente al tiempo de propagación del mismo. Al querer representar la IR en un formato digital, el tiempo se verá discretizado en muestras o *samples*, y la contribución de cada camino se realizará en una muestra de tiempo determinada. Estos intervalos de tiempo dependen del *sample rate* elegido (ver Figura 4.4). La IR puede entonces representarse como un array, donde cada una de sus celdas representa el valor único de la IR para cada intervalo de tiempo de largo $\frac{1}{\text{sample_rate}}$. El array final tendrá, en cada celda, el acumulado de las intensidades de los caminos con tiempos de propagación similares. Las celdas para las que no se encontraron caminos con ese tiempo de propagación, se completan con 0.

En el Algoritmo 3 puede verse cómo la IR se construye encontrando la celda correspondiente para sumar la contribución de cada camino. Aquí finalmente se ve la importancia del tiempo de propagación de los rayos, sin este no sería posible construir la IR. Como todos los caminos parten del tiempo inicial $t = 0$, el tiempo de llegada del camino i se calcula como $t = \frac{\text{DistanciaRecorrida}(i)}{\text{velocidad.del.sonido}}$. Luego,

la intensidad del camino se suma en la celda $\text{round}(t * \text{sample_rate})$, la celda que corresponde al intervalo de tiempo en el que el camino llega al receptor.

De aquí en más, por simplicidad, se considerará a cada elemento de la IR como la intensidad de un camino hasta el receptor (en lugar de la sumatoria de las intensidades de varios caminos). Además, en lugar de intervalos de tiempo, se hablará de k instantes de tiempo, donde k es el largo del array IR.

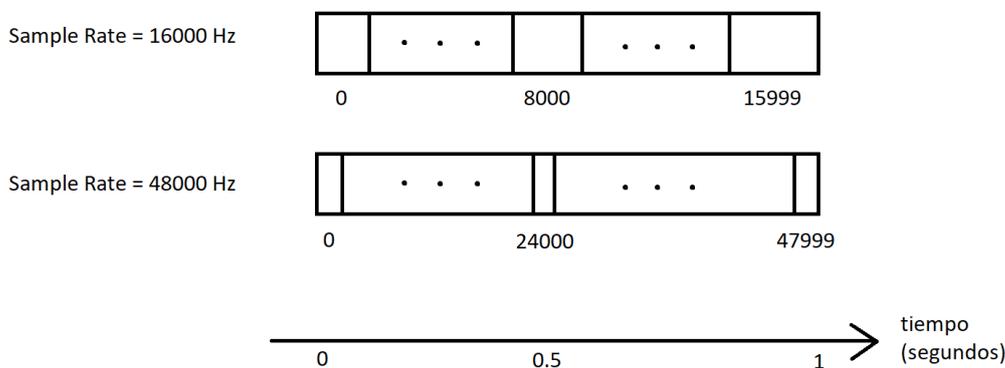


Figura 4.4: Las IRs tienen distinta cantidad de elementos para distintos *sample rates*. Para 16 KHz un camino de duración 0,5 s contribuye a la celda 8000, mientras que para 48 KHz contribuye a la celda 24.000.

Algoritmo 3 Algoritmo para la creación de la respuesta al impulso

```

function generarIR(duración, caminosEncontrados)
  IR = crearArray(duración) //Crea un array de largo sample_rate * duración
  loop  $i \in \text{caminosEncontrados}$ 
    tiempo_de_propagación =  $i.\text{DistanciaRecorrida}() / \text{velocidad\_del\_sonido}$ 
    índice = Redondear(tiempo_de_propagación * sample_rate)
    if índice < duración * sample_rate then
      //El índice cae dentro del array. Se suma la intensidad del camino en la celda correspondiente.
      IR[índice] +=  $i.\text{Intensidad}()$ 
    end if
  end loop
  return IR
end function

```

4.3. Implementación de la auralización

4.3.1. Convolución de las señales

Con el array que contiene la IR, es posible auralizar una señal de audio realizando su convolución. ¿Pero cómo se realiza la convolución? Para enten-

der cómo se realiza la convolución, es necesario pensar en cómo cada impulso de audio (correspondiente a un *sample*) se propaga por cada uno de los distintos caminos encontrados hasta el receptor y cómo esto genera que dicho impulso tenga una contribución en distintos tiempos para el sonido final ¹. Un camino más largo hasta el receptor tiene una duración mayor, por lo tanto la contribución de dicho camino al sonido final será posterior (ver Figura 4.5).

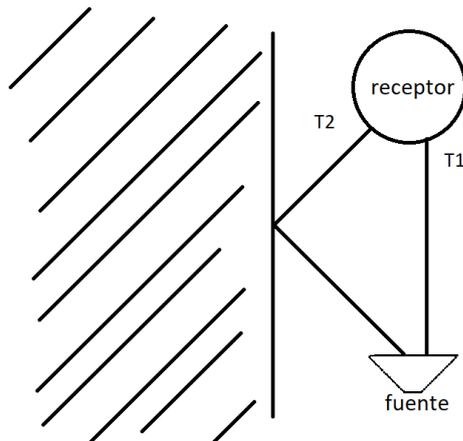


Figura 4.5: 2 caminos distintos con distinto tiempo de propagación $T_2 > T_1$. Un cierto *sample* s_i emitido en el instante t_i llega al receptor en el instante $t_i + T_1$ y nuevamente en el instante $t_i + T_2$

Considérese la siguiente nomenclatura para facilitar la comprensión de esta sección:

- Sea s el conjunto de *samples* de la señal de entrada del programa, es decir, la señal que es emitida por la fuente en la escena.
- Sea s' el conjunto de *samples* de la señal de salida del programa, es decir, la señal percibida por el receptor en la escena.
- Sea c el conjunto de caminos hallados hacia el receptor. La i -ésima celda del vector IR representa la intensidad que llega al receptor a través del camino c_i .

La forma de calcular s'_i (el *sample* de s' en el tiempo t_i) es determinar qué *samples* de s llegan al receptor en el tiempo t_i a través de alguno de los caminos de c . Para visualizar esto se plantea el siguiente ejemplo:

¹En el procesamiento de señales, las señales de audio se descomponen en varios impulsos de Dirac, por lo que la convolución se escribe en términos de impulsos y no de muestras. [43][50]. Sin embargo, el razonamiento es el mismo para muestras e impulsos, por lo que se utiliza este enfoque orientado a la programación.

1. Para el instante de tiempo inicial t_0 , el único *sample* emitido por la fuente es s_0 (el primero s). Además, el único camino por el cual el *sample* s_0 podrá llegar al receptor en el instante t_0 , es a través del camino inmediato c_0 ² (ver Figura 4.6).

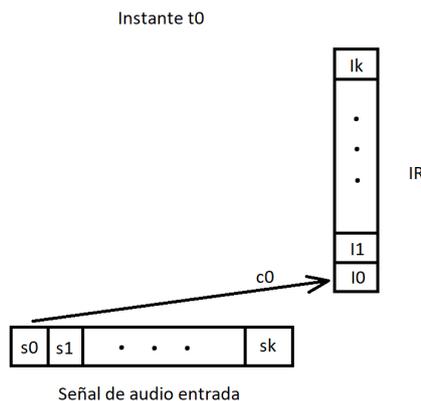


Figura 4.6: En el instante de tiempo t_0 el *sample* s_0 llega al receptor a través del camino c_0 .

2. Para el siguiente instante de tiempo t_1 , puede observarse algo interesante: El *sample* s_0 es emitido por la fuente $\frac{1}{\text{sample.rate}}$ segundos antes del instante t_1 . Por lo tanto, para llegar al receptor en dicho instante, deberá propagarse por c_1 , el camino con tiempo de propagación $\frac{1}{\text{sample.rate}}$ segundos. Por otro lado, el *sample* s_1 , que es emitido en el instante t_1 , llegará en el mismo instante a través del camino inmediato c_0 (ver Figura 4.7).
3. Al aplicar el mismo paso varias veces pueden obtenerse las contribuciones para el *sample* de s' en el tiempo t_k (ver Figura 4.8).

Con este ejemplo entonces, se ha descrito cuales son las contribuciones para los *samples* s'_0 , s'_1 y s'_k de la señal s' . La forma de calcular los *samples* de la señal s' es ponderar cada *sample* de s por la intensidad del camino que le corresponde. En la Ecuación 4.2 puede verse el cálculo de forma genérica.

²El camino inmediato c_0 es el camino con tiempo de propagación 0. Cabe aclarar que en la práctica no hay caminos de tiempo de propagación 0, porque no hay caminos de largo 0 que vayan de la fuente al receptor.

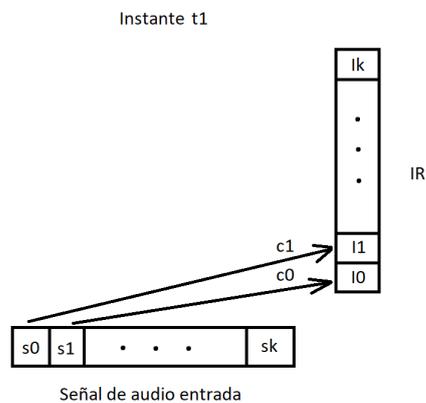


Figura 4.7: En el instante de tiempo t_1 el *sample* s_0 llega al receptor a través del camino c_1 , mientras que s_1 llega a través de c_0 .

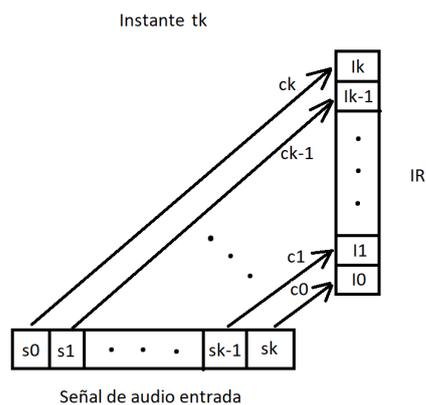


Figura 4.8: Cada *sample* se corresponde con un camino cuyo tiempo de propagación es igual a la antigüedad de dicho *sample*.

$$s'_i = \sum_{j=0}^i s_j \cdot I_{i-j} \quad (4.2)$$

Donde I_i es la intensidad del camino i .

El lector puede haber intuido, a partir de la presentación de este ejemplo y de la Ecuación 4.2, que el cálculo de los *samples* de salida s'_i puede expresarse como el producto de 2 vectores de largo i . Entonces, la señal de salida s' puede expresarse como el producto del vector de la señal de entrada s de largo n y una matriz ρ de dimensión $n \times n$, donde n es el número de *samples* considerados en la convolución. La fila j de ρ está formada por los primeros j elementos el vector IR.

$$\rho = \begin{pmatrix} I_0 & 0 & \cdots & 0 \\ I_1 & I_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_k & \cdots & I_1 & I_0 \end{pmatrix} ; \quad s = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_k \end{pmatrix} \quad (4.3)$$

$$s' = \rho \cdot s \quad (4.4)$$

4.3.2. Buffer circular

Como puede verse, para que el producto de la convolución tenga sentido, es necesario que las dimensiones de ρ y s coincidan. Ya que el *sample_rate* de la IR y de la señal de audio coinciden, solo hace falta que tengan la misma duración. La elección de la duración de ambas señales se hizo en base a cuántos *samples* pasados se quieren recordar al momento de calcular un *sample* de salida. En este caso se optó por una duración de 1 segundo. Es decir, tanto la IR como la señal de audio almacenada en memoria (el historial de *samples*) deben tener 1 segundo de duración.

Sin embargo, el hecho de que el historial de *samples* esté limitado a 1 segundo no quiere decir que el audio al cual se le aplica la convolución también deba estar limitado a 1 segundo. El *buffer* que almacena el historial de *samples* puede actualizarse cada vez que se obtengan nuevos *samples* para reemplazar a los más antiguos. Debido a la alta frecuencia con la que este *buffer* es actualizado, sería muy costoso borrarlo y crearlo nuevamente cada vez. Las operaciones de

reserva de memoria actuarían como un cuello de botella. Además, reescribir el *buffer* completo cada vez consumiría mucho más tiempo que solo reescribir los *samples* más antiguos. Notar que este problema empeora a medida que aumentan el *sample rate* y el tiempo del historial³. Para resolver estos problemas se utiliza un **buffer circular**.

Un *buffer* circular es una estructura de datos que consiste de un array de tamaño fijo y 2 punteros o índices que denotan el elemento inicial y final. Si se agrega un elemento, el puntero de la cola se incrementa en 1. Si se borra el primer elemento, el puntero de la cabeza se incrementa en 1. Si alguno de los punteros apunta a la última posición del array y se quiere incrementar, el puntero se mueve a la primera posición del array. De ahí el nombre *buffer* circular.

Para el caso de la auralización, como los elementos con más de 1 segundo de antigüedad no son utilizados, simplemente pueden sobrescribirse cuando el *buffer* está lleno. En la Figura 4.9 puede verse un ejemplo de esto.

Algoritmo 4 Algoritmo para insertar elementos en un buffer circular

```

function insertarElementos(elementos, cantidad)
  if cola + tamaño.buffer >= cantidad then
    //Se hace la copia en 2 partes. Una desde la posición de la cola hasta el final y
    //otra para el resto de los elementos a partir de la posición inicial del array.
    tamaño.primer_copia = tamaño.buffer - 1 - cola
    copiar(buffer[cola + 1], elementos[0], tamaño.primer_copia)
    copiar(buffer[0], elementos[tamaño.primer_copia], cantidad - tamaño.primer_copia)
    cola = cantidad - tamaño.primer_copia - 1
    cabeza = (cola + 1) % tamaño.buffer
    buffer.lleno = true
  else
    //La inserción puede realizarse con una sola copia.
    if cola == cabeza then
      //Se insertan elementos en un buffer vacío.
      copiar(buffer[cola], elementos[0], cantidad)
      cola = cola + cantidad - 1
    else
      copiar(buffer[cola + 1], elementos[0], cantidad)
      cola = cola + cantidad
    end if
    if buffer.lleno then
      //Si el buffer está lleno la cabeza es desplazada por los nuevos elementos.
      cabeza = (cola + 1) % tamaño.buffer
    end if
    if !buffer.lleno and cola == tamaño.buffer - 1 then
      buffer.lleno = true
    end if
  end if
end function

```

El *buffer* circular se implementó como un template de C++, que toma el tipo de datos de los *samples* como argumento. De esta forma, la implementación

³Con un *sample rate* de 16 KHz y un historial de 1 segundo se tiene un *buffer* de largo 16000. Con 40 KHz y 2 segundos el *buffer* es de largo 80000 y con 48 KHz y 3 segundos es de largo 144.000.

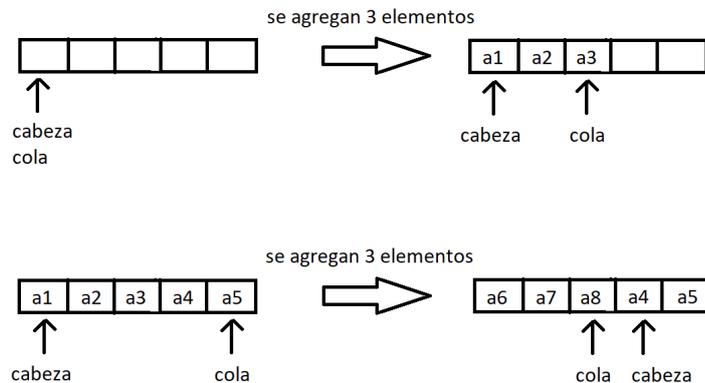


Figura 4.9: Ejemplos de buffer circular.

puede utilizar *samples* de cualquier tipo de datos.

Algoritmo 5 Algoritmo para obtener un elemento del buffer circular

```

function obtenerElementoDeBuffer(indice)
  if cabeza + indice < tamañoHistorialDeSamples then
    return buffer[cabeza + indice]
  else
    return buffer[indice - (tamañoHistorialDeSamples - cabeza)]
  end if
end function

```

4.3.3. Procesamiento de audio desde micrófono

Como se mencionó en el Capítulo 3 la solución provee la posibilidad de realizar la auralización de una señal de audio en vivo o una señal grabada en un archivo. En esta sección se desarrolla la implementación para el primer caso, donde el usuario puede experimentar la auralización de lo que capta un micrófono en tiempo real. Por ejemplo, este modo puede usarse para escuchar la voz del usuario o un instrumento musical.

Al inicializar el *stream* de audio de RtAudio es posible especificar un dispositivo de entrada disponible, por donde registrar la señal de entrada. La función de *callback* luego toma una cantidad de *samples* dada de este dispositivo por llamada. Estos *samples* son insertados en el historial de *samples* y se procede a calcular la convolución.

Debido a lo explicado en la Sección 4.3.2, el lector puede estar preguntándose: Si el *array* de la señal de entrada no está ordenado como un *array* común

¿Cómo se realiza el producto $\rho \cdot s$? ¿Es necesario reordenarlo? Si bien reordenar el array podría ser una solución, esto provocaría un *overhead* que echaría por tierra la ventaja de usar un *buffer* circular.

Revisando el producto $\rho \cdot s$ pueden observarse algunas cosas. La matriz ρ es diagonal inferior, por lo tanto, muchos de los productos escalares del producto matriz vector son innecesarios. Aún más, como se mencionó en la Sección 3.2.5, cada ejecución de la función de *callback* calcula una cantidad reducida de *samples* para reproducir antes de realizar la próxima llamada. RtAudio permite procesar un número potencia de 2 *samples* por llamada y se eligió el número 512 debido a que con esa cantidad no se percibieron cortes en el audio. Esto quiere decir, que no es necesario calcular el producto entero de ρ y s . Por estos motivos, es preferible realizar el producto de la Ecuación 4.4 como se ve en el Algoritmo 6, a utilizar alguna biblioteca para operaciones con matrices. En este caso, el orden del vector de la señal de entrada puede considerarse al realizar el cálculo del producto.

Algoritmo 6 Algoritmo para el cálculo de la convolución

```

function calcularConvolución
  loop  $i \in [0, 512)$ 
     $sample = 0$ 
    loop  $j \in [0, tamañoHistorialDeSamples - i)$ 
       $sample += IR[j] * obtenerElementoDeBuffer(tamañoHistorialDeSamples - 1 - i - j)$ 
    end loop
    //Para obtener una señal binaural duplicamos el sample. 1 para cada canal.
    //Los samples en este caso se cuentan de a pares. Los samples para ambos canales en el mismo
    //instante de tiempo van seguidos.
     $indiceSalida = (512 * 2) - 1 - (i * 2)$ 
     $señalSalida[indiceSalida] = sample$ 
     $señalSalida[indiceSalida - 1] = sample$ 
  end loop
end function

```

El resultado de este cálculo son 512 *samples* de salida. Estos *samples* de salida se pasan a RtAudio para ser reproducidos hasta la próxima llamada de la función *callback*, donde se calcularán los siguientes 512 *samples*.

4.3.4. Procesamiento de audio desde archivo

En esta sección se detalla la implementación para realizar la auralización de una señal grabada en un archivo. En este caso se quiere cargar la señal del archivo y reproducirla para que el usuario experimente la auralización en tiempo real.

Para procesar una señal de audio cargada desde un archivo deben hacerse algunas consideraciones con respecto a la señal de entrada. Se cuenta con

la señal completa desde el inicio, que no se ajusta del todo a la estructura de *buffer* circular utilizada en la parte anterior. Una posible forma de resolver este problema utilizando el mismo diseño es considerando un *array* de largo $1 + d$ segundos, donde d es la duración del archivo de audio. Las celdas adicionales se completan con 0 para empezar en silencio e incorporar las muestras del archivo a medida que avanza el tiempo. Dentro de dicho *array* se considera una ventana de largo 1 segundo que se desplaza para simular el efecto de una señal recibida en tiempo real. En la Figura 4.10 puede observarse una descripción gráfica de lo antes descrito.

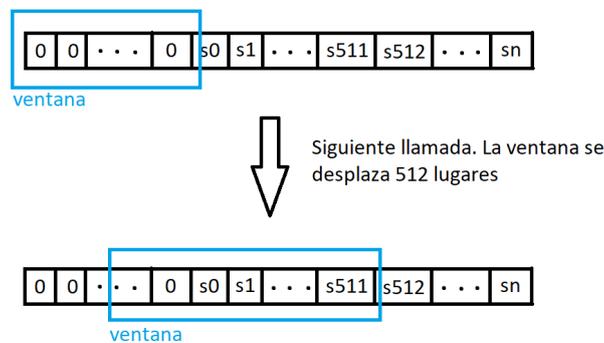


Figura 4.10: La ventana se desplaza a lo largo del *buffer* circular para crear el efecto de una señal recibida en tiempo real.

Algoritmo 7 Algoritmo para el cálculo de la convolución con una señal cargada desde archivo

```

function calcularConvoluciónConSeñalDeArchivo
    loop  $i \in [0, 512)$ 
         $sample = 0$ 
        loop  $j \in [0, sample\_rate - i)$ 
             $sample += IR[j] * obtenerElementoDeBuffer(sample\_rate - 1 - i - j)$ 
        end loop
        //Para obtener una señal binaural duplicamos el sample. 1 para cada canal.
         $indiceSalida = (512 * 2) - 1 - (i * 2)$ 
         $senalSalida[indiceSalida] = sample$ 
         $senalSalida[indiceSalida - 1] = sample$ 
    end loop
    loop  $i \in [0, 512)$ 
         $setearElementoDeBuffer(i, 0)$  //Los samples utilizados se setean en 0.
    end loop
     $cabeza = (cabeza + 512) \% tamaño\_buffer$  //Se actualiza la posición de la ventana.
     $cola = (cola + 512) \% tamaño\_buffer$ 
end function

```

Capítulo 5

Experimentos y resultados

En este capítulo se muestran detalles de las pruebas realizadas, con el objetivo de determinar las características positivas y negativas de los algoritmos implementados en las dimensiones de rendimiento computacional y precisión de los resultados.

5.1. Ambiente de prueba

Procesador	Intel i7 6500U - 4 CPUs - 2,5 GHz
Gráficos integrados	Intel HD Graphics 520
RAM	8 GiB - 2133 MHz

Tabla 5.1: Características del hardware utilizado

SO	Windows 10 Home
Embree	v3.12.1
OpenGL	v4.5
RtAudio	v5.1.0

Tabla 5.2: Características del entorno de desarrollo utilizado

5.2. Escenas estudiadas

Para analizar los resultados obtenidos se utilizaron las mismas escenas que en el trabajo de Inès de Bort [24]. En dicho trabajo se estudiaron 3 espacios abiertos de distintas características: la fachada de un edificio, un cañón urbano

y un patio interior. Estas escenas pueden ser fácilmente representadas en modelos CAD como puede verse en la Figura 5.2. De aquí en más se referirá a las escenas como 1D, 2D y 3D como se hace referencia en [24] siguiendo el orden en el que aparecen en la Figura 5.1.

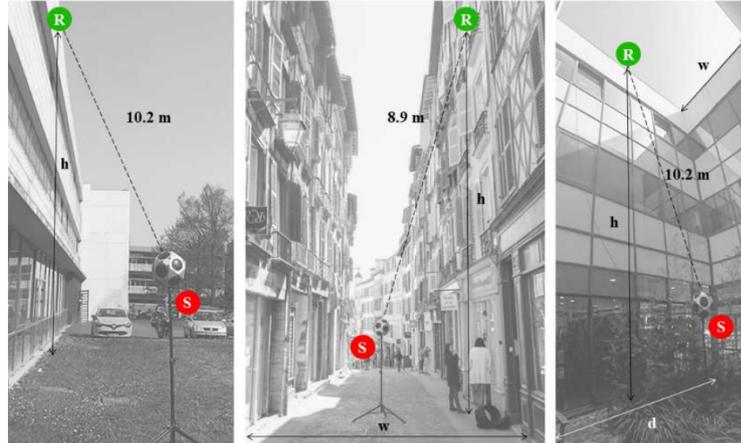


Figura 5.1: Escenas estudiadas. S marca la posición de la fuente (*source*) y R marca la posición del receptor (*receiver*). Imágenes tomadas de [24].

Escena	Alto	Ancho	Profundidad
1D	11,7m		
2D	10,3m	5m	
3D	11,7m	5m	8m

Tabla 5.3: Medidas de las escenas estudiadas.

Unas de las ventajas más importantes de utilizar estas escenas es que en el trabajo mencionado, se realizaron mediciones de la IR en cada una de las escenas. Estas mediciones son imprescindibles si se quiere hacer un análisis de los resultados de la simulación.

5.3. Análisis de la simulación

Para analizar el resultado de la simulación en las escenas estudiadas se realizó un análisis comparando las mediciones obtenidas con distintas simulaciones. En estas simulaciones se utilizan distintos valores para algunos parámetros como cantidad de rebotes, coeficientes de absorción y cantidad de rayos.

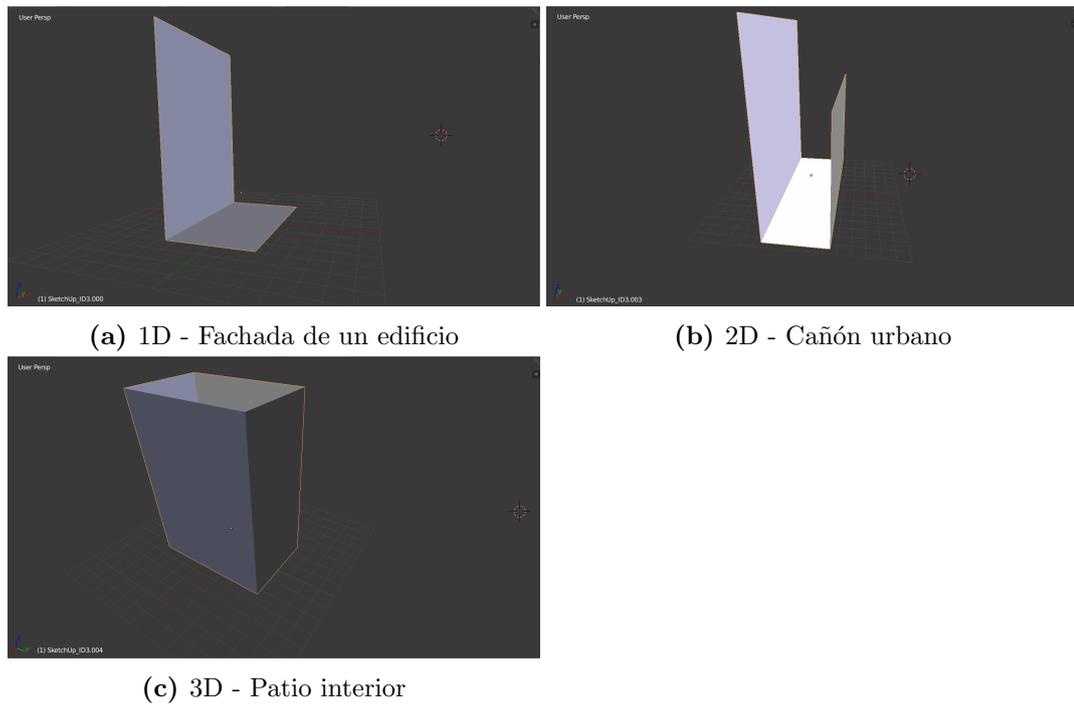


Figura 5.2: Modelos CAD de las escenas estudiadas.

Por otro lado, otros parámetros se mantienen constantes como posición y nivel sonoro de la fuente y posición y tamaño del micrófono. Estos últimos valores se toman del estudio en cuestión.

En concreto se realizaron 2 pruebas separadas. En primera instancia, se analizó el impacto de la cantidad de rebotes y el coeficiente de absorción. En esta comparación se mantuvo constante la cantidad de rayos emitidos con la secuencia pseudoaleatoria. En segunda instancia se analizó el impacto de modificar la cantidad de rayos y la distribución utilizada para la generación de rayos (pseudoaleatoria y Halton).

Las simulaciones se obtuvieron con el modo *simulate* para que las IR obtenidas coincidan con el *sample rate* y tipo de datos de las mediciones. El *sample rate* de las mediciones es de 48.000 Hz.

5.3.1. Análisis cualitativo

En esta sección se analizan las gráficas obtenidas de las IR medidas y simuladas. Como se mencionó en el Capítulo 2, la medición de la IR posee valores positivos y negativos al tratarse de los valores de la presión efectiva. Sin embargo, como pudo verse en la ecuación 2.2, la intensidad es directamente

proporcional al cuadrado de la presión efectiva. Por lo tanto, los valores de la medición son elevados al cuadrado para poder compararlos con la simulación.

Se utilizaron 10^7 rayos y un nivel sonoro en potencia de $100dB$ igual al de la fuente utilizada en las mediciones mencionadas anteriormente.

Escena 1D

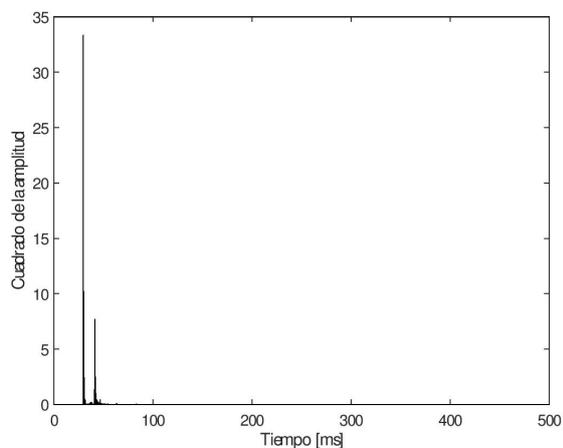
En la Figura 5.3 pueden observarse las gráficas para la escena 1D. Como puede verse en la Figura 5.3a, la IR de esta escena presenta 2 picos bien distinguidos. El primero ocurre a los $29\ ms$ mientras que el segundo a los $41\ ms$, aproximadamente. Por otro lado, puede apreciarse como todas las simulaciones presentan los mismos 2 picos. Los tiempos de los picos también son bastante similares, situándose el primero en el orden de los $28\ ms$ y el segundo a los $39\ ms$, aproximadamente.

En cuanto a las diferencias entre las simulaciones, puede observarse una clara diferencia al variar el coeficiente de absorción. El coeficiente de absorción $0,5$ es el que generó resultados más similares a la medición. Por otro lado, al observar la forma de esta escena es fácil ver que no hay posibilidad de que el sonido llegue por un camino con más de 2 rebotes. Los únicos caminos indirectos hasta el oyente, son caminos de 1 o 2 rebotes, donde el rayo rebota en el piso, en la pared o en ambos, antes de llegar al oyente. Por este motivo, aumentar la cantidad de rebotes de 10 a 30 no tiene un efecto visible en los resultados. En la Figura 5.3b se dibuja el último rebote en la propagación de un rayo antes que este alcance al oyente.

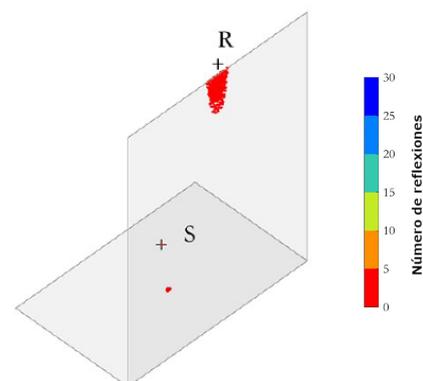
Escena 2D

En esta escena se evidencia el incremento de la cantidad de rebotes con respecto a la escena anterior (ver Figura 5.4). En la Figura 5.4a puede observarse cómo la cantidad de picos se ha incrementado de manera notoria y como la señal puede percibirse por mucho más tiempo. Inicialmente se aprecian unos 3 picos seguidos alrededor de los $26\ ms$, $33\ ms$ y $37\ ms$, seguidos de una cola de decaimiento exponencial.

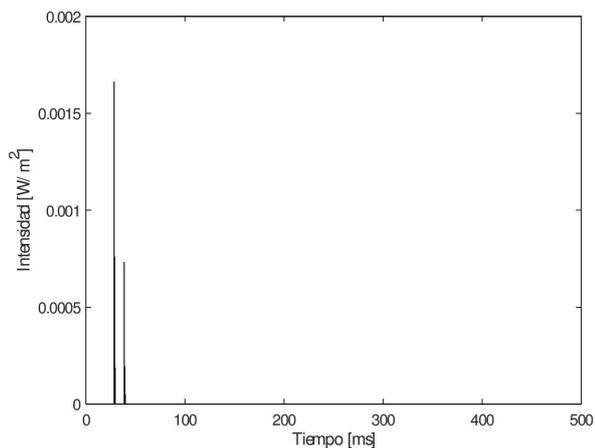
Al contrario de la escena 1D, las simulaciones de esta escena distan bastante de la medición. Puede observarse como con un coeficiente de absorción de $0,5$ la señal decae demasiado rápido. Con la absorción de $0,1$ la señal puede percibirse



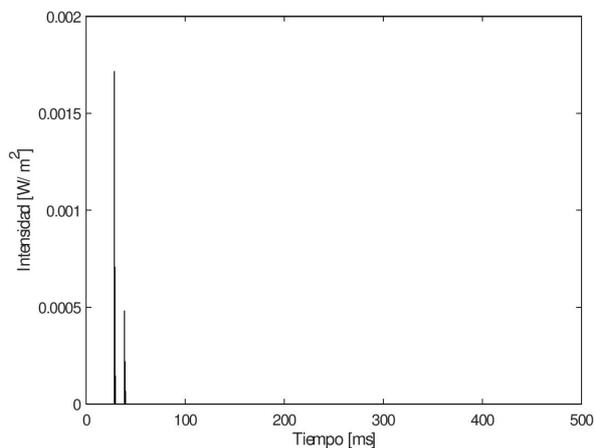
(a) Respuesta al Impulso medida en la escena 1D.



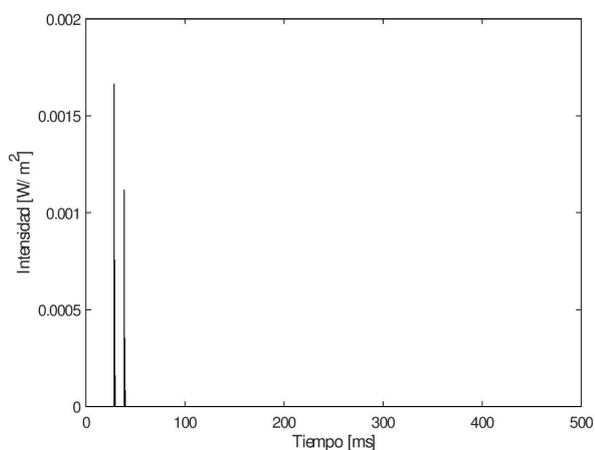
(b) Rebotes de los rayos en la escena 1D. Gráfico tomado de [24]



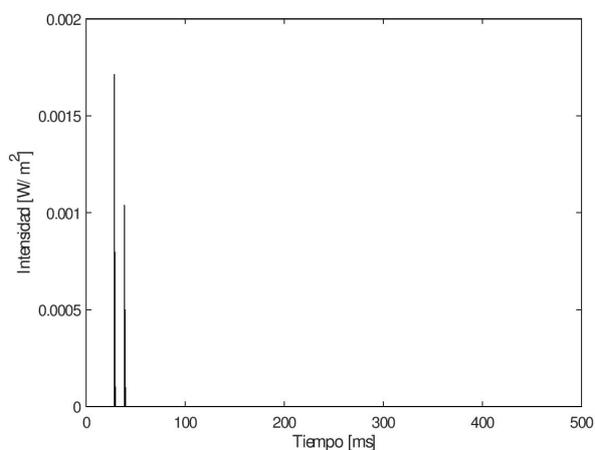
(c) Respuesta al Impulso simulada
Absorción: 0,5 - Máx. rebotes: 10



(d) Respuesta al Impulso simulada
Absorción: 0,5 - Máx. rebotes: 30



(e) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 10



(f) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 30

Figura 5.3: Escena 1D.

por más tiempo, superando los 400 *ms* con una cantidad máxima de rebotes de 30, pero sin llegar a los 200 *ms* para un máximo de 10 rebotes. Los picos iniciales presentan tiempos similares a los de la medición situándose alrededor de los 24 *ms*, 31 *ms* y 35 *ms*.

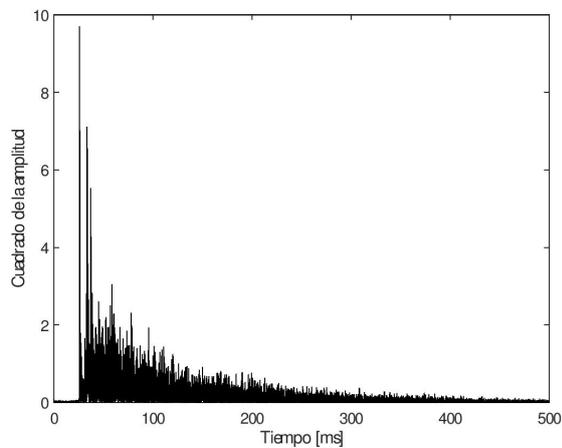
Una de las principales diferencias entre la simulación y la medición es la distribución de los picos. La medición presenta una distribución homogénea de los picos creando una imagen densa de la señal, mientras que en la simulación los picos se acumulan en ciertos intervalos de tiempo. Por otro lado, aunque el coeficiente de absorción de 0,1 genera resultados similares, la proporción de los primeros picos no coincide con los de la medición para ninguno de los parámetros.

Es posible que la diferencia entre la medición y las simulaciones se deba a las diferencias entre la escena real y el modelo utilizado. Como puede verse en la Figura 5.1, la escena 2D es la que más difiere de su modelo, ya que presenta relieves y huecos más pronunciados.

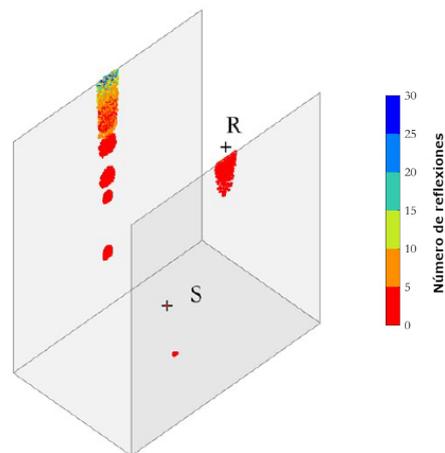
Escena 3D

Para esta escena, los resultados de la simulación lucen cualitativamente de forma similar a los de la medición. Se aprecia que la escena real tiene paredes vidriadas lisas, por lo que se la escena utilizada en la simulación se asemeja mucho más a la real, en comparación con lo ocurrido en la escena 2D. Si bien la escena 3D es la escena con más paredes y por lo tanto contiene los caminos con mayor cantidad de rebotes, aumentar la cantidad máxima de rebotes a más de 30 no afecta visiblemente los resultados.

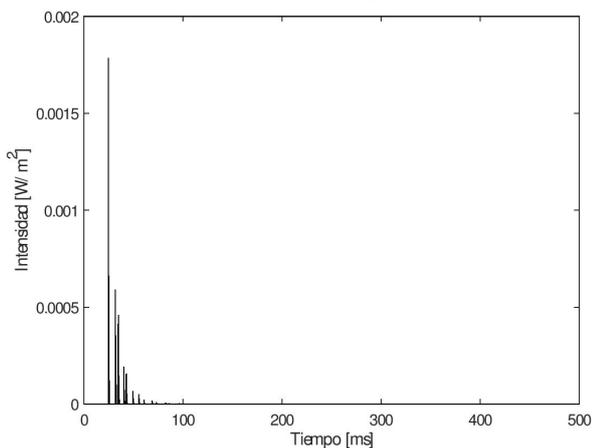
Tanto en la medición como en las simulaciones pueden verse 2 picos iniciales de intensidad similar. En la medición los primeros 2 picos se presentan alrededor de los 30 *ms* y 36,5 *ms* respectivamente, el tercer pico se encuentra a los 39 *ms* y el cuarto a los 46,5 *ms*. En la medición los picos más altos en el mismo orden son: 28 *ms*, 36 *ms*, 44,5 *ms* y 57,3 *ms*.



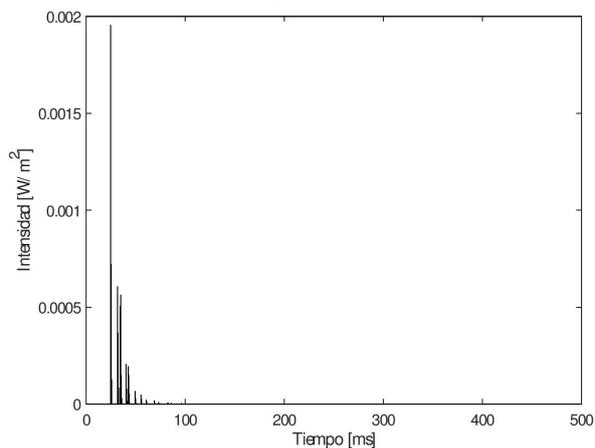
(a) Respuesta al Impulso medida en la escena 2D.



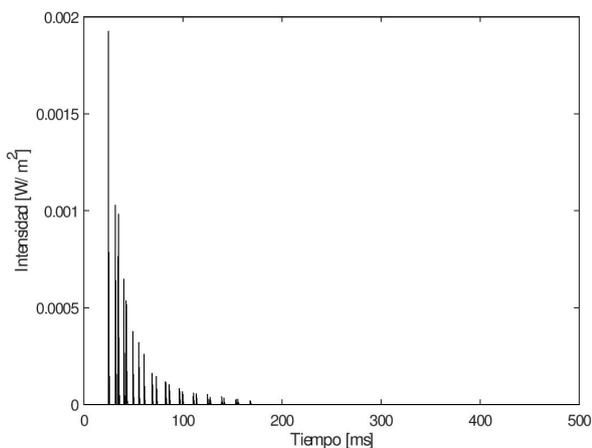
(b) Rebotes de los rayos en la escena 2D. Gráfico tomado de [24]



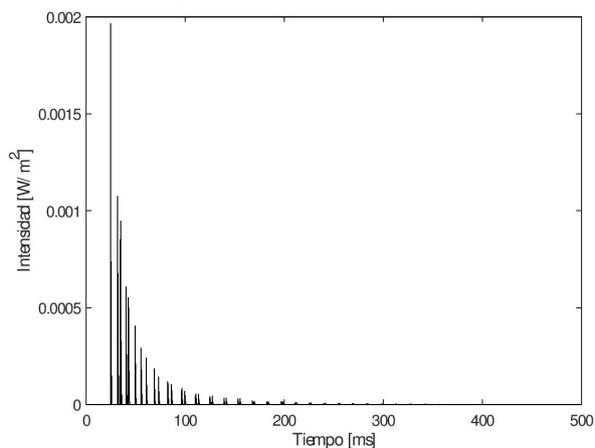
(c) Respuesta al Impulso simulada
Absorción: 0,5 - Máx. rebotes: 10



(d) Respuesta al Impulso simulada
Absorción: 0,5 - Máx. rebotes: 30

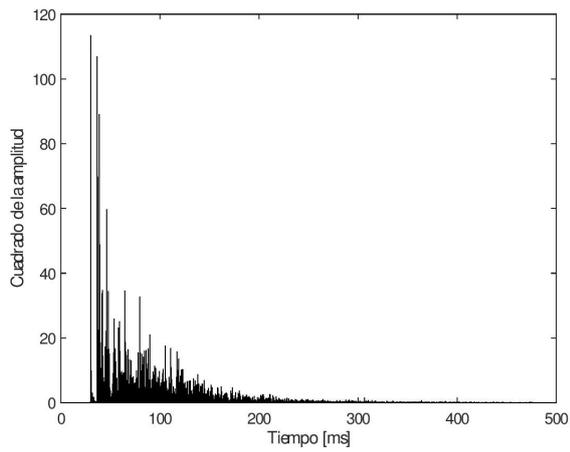


(e) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 10

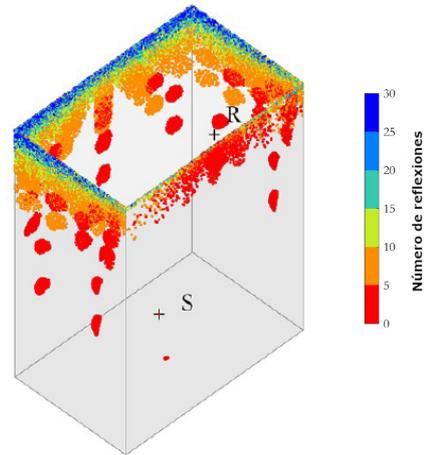


(f) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 30

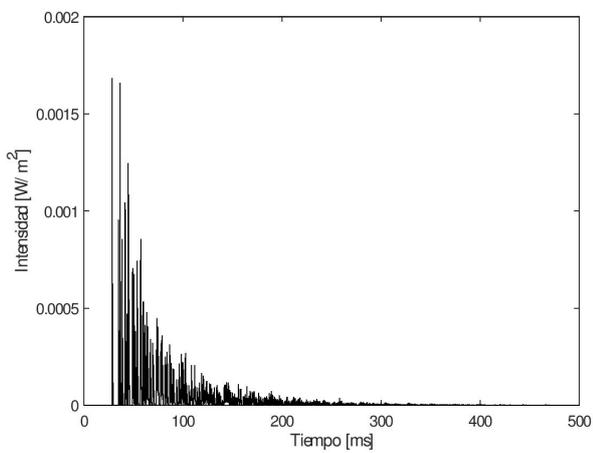
Figura 5.4: Escena 2D.



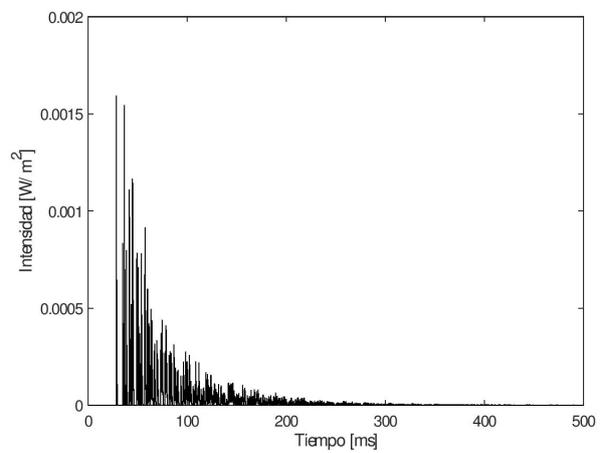
(a) Respuesta al Impulso medida en la escena 3D.



(b) Rebotes de los rayos en la escena 3D. Gráfico tomado de [24].



(c) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 30



(d) Respuesta al Impulso simulada
Absorción: 0,1 - Máx. rebotes: 60

Figura 5.5: Escena 3D.

5.3.2. Forma de los picos

Al analizar las gráficas más detalladamente, puede observarse como los picos tienen una forma característica. Los picos que se perciben como en un instante de tiempo, en realidad corresponden a la intensidad percibida por varios rayos que llegan en un intervalo de tiempo. En la Figura 5.6 puede observarse la diferencia entre el primer pico de la medición y la simulación en la escena 1D.

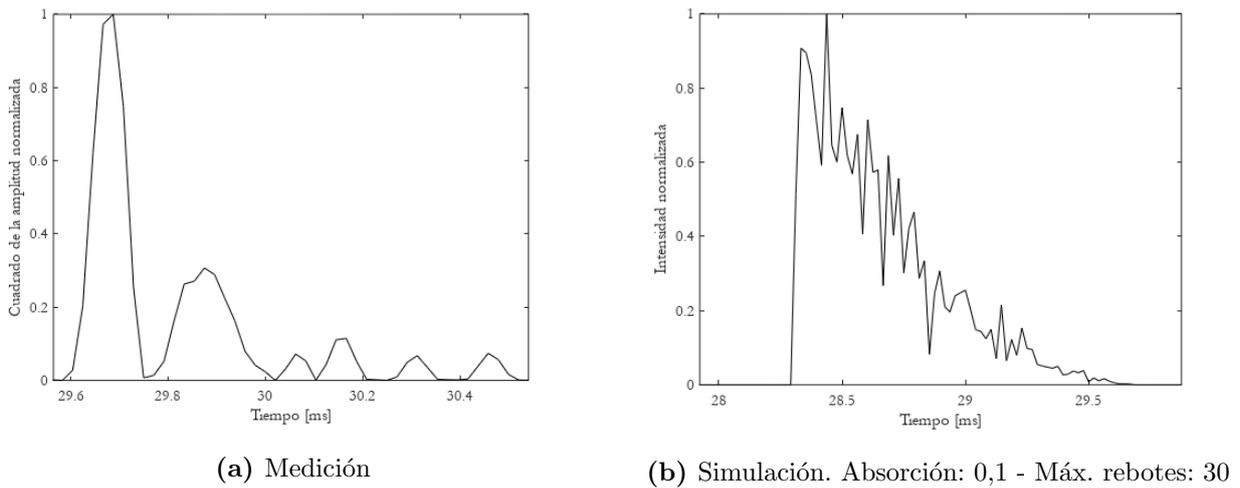


Figura 5.6: Formas del primer pico en la escena 1D. Gráficos normalizados.

El pico de la medición está compuesto por 258 *samples* acumulados en algunos “subpicos”, mientras que el pico de la simulación está compuesto por 66 *samples* sin huecos entre ellos.

La “duración” de los picos es bastante similar, siendo de aproximadamente 1 *ms* y 1,5 *ms* para la medición y la simulación respectivamente.

Cabe aclarar que la duración del pico de la simulación es consecuencia del tamaño de la esfera que modela al receptor. Una esfera más grande recibirá más rayos pertenecientes al mismo frente de onda, lo que provocará un pico de mayor duración. Se utilizó una esfera de 1 *m* de diámetro como en [24], pero utilizar una esfera de un diámetro menor resultaría en un pico de menor duración.

5.3.3. Cantidad de rayos y distribución

En esta sección se analiza cuánto influye la cantidad utilizada de rayos y la distribución utilizada para generarlos. Para este análisis se utilizan un coeficiente de absorción de 0,1 y una cantidad de rebotes máxima de 30 ya que produjeron buenos resultados en la parte anterior.

Escena 1D

Como puede verse en la Figura 5.7 la cantidad de rayos no modifica la forma de la gráfica. Sin embargo, puede observarse que la intensidad registrada en la simulación con 10^5 rayos es mayor. Lo mismo puede observarse en la Figura 5.8, donde también puede observarse una diferencia visible entre las intensidades de los picos de las simulaciones con Halton. En particular, puede verse cómo la intensidad de los picos disminuye a medida que aumenta la cantidad de rayos.

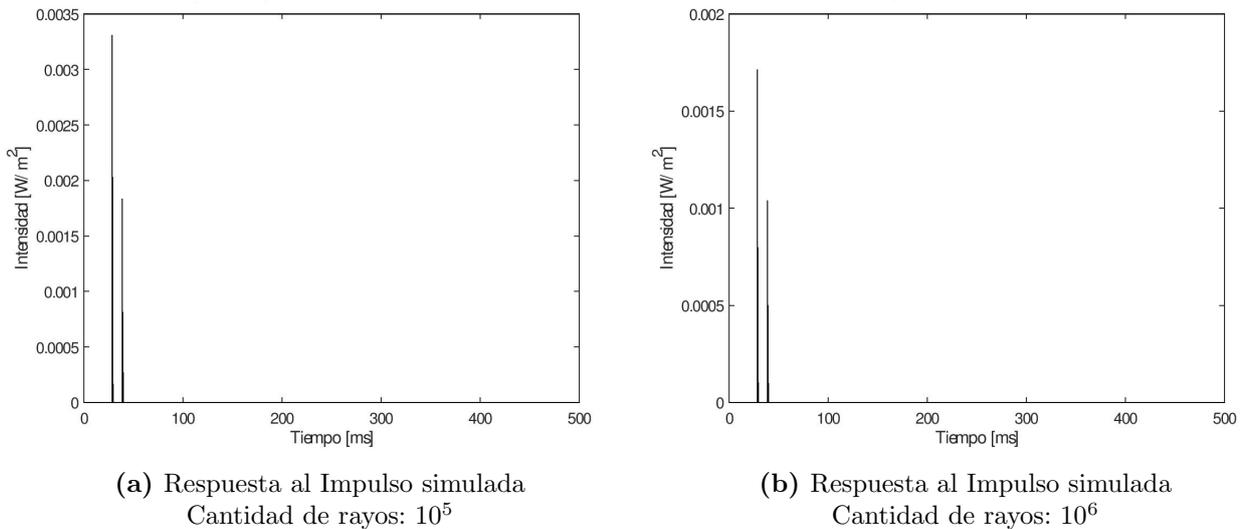
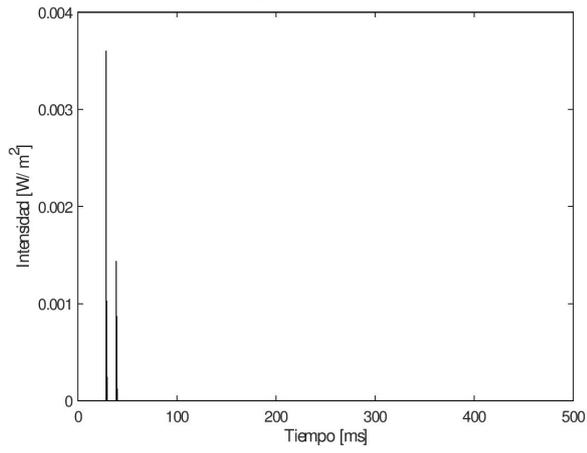
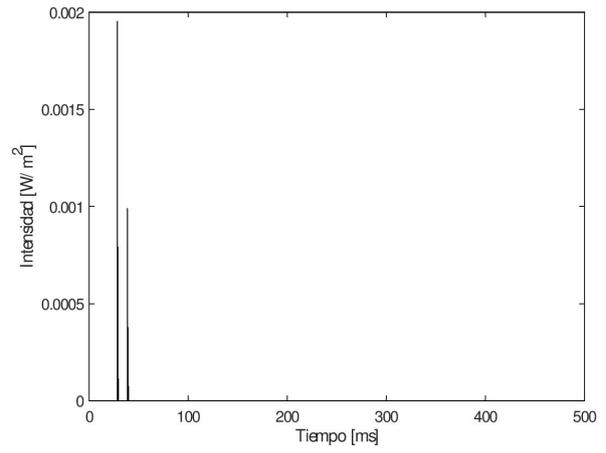


Figura 5.7: Simulaciones de la escena 1D. Distribución pseudoaleatoria.

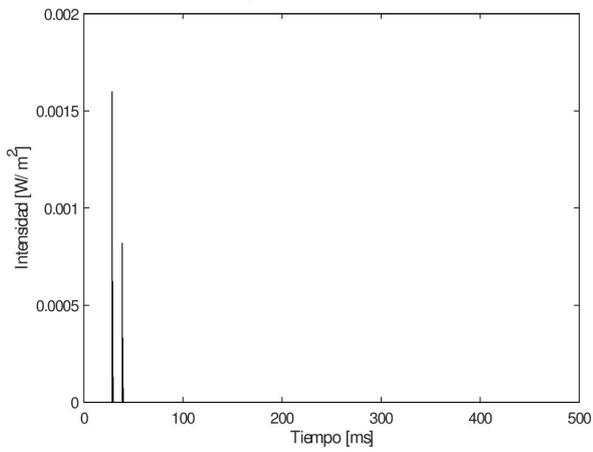
Si bien el valor de la intensidad no es particularmente importante (ver Sección 2.2.4), esto es una indicación de que los rayos están acumulándose en ciertos intervalos de tiempo. Recordar que al disminuir la cantidad de rayos, la energía inicial de estos aumenta, por lo tanto, cada camino tendrá una mayor contribución a la señal final. Por ejemplo, para la simulación con Halton y 10^5 rayos se registran 52 rayos que contribuyen al primer pico, mientras que para la



(a) Respuesta al Impulso simulada
Cantidad de rayos: 10^5



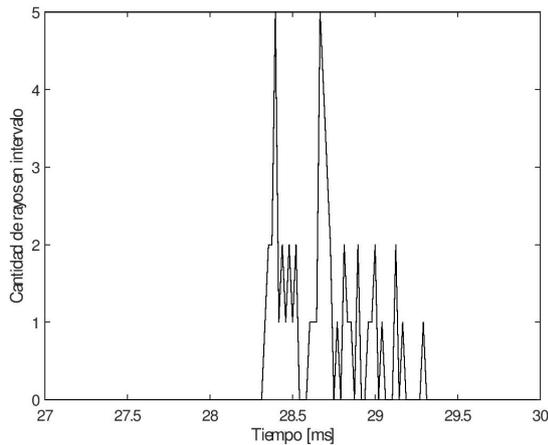
(b) Respuesta al Impulso simulada
Cantidad de rayos: 10^6



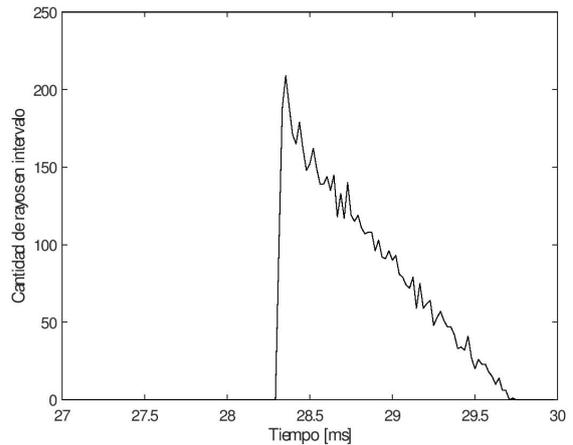
(c) Respuesta al Impulso simulada
Cantidad de rayos: 10^7

Figura 5.8: Simulaciones de la escena 1D. Distribución Halton.

simulación con 10^7 rayos se registran 6007. Esto indica que si bien la cantidad total de rayos que contribuyen al pico se mantiene de forma proporcional, la distribución de los mismos es bastante diferente. Como puede verse en la Figura 5.9, los rayos se distribuyen más homogéneamente en todo el intervalo cuanto mayor es el número total de rayos utilizados.



(a) Distribución de rayos en el primer pico
Cantidad de rayos: 10^5



(b) Distribución de rayos en el primer pico
Cantidad de rayos: 10^7

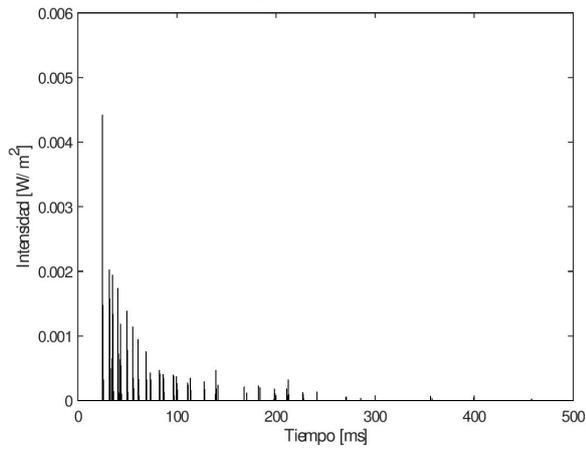
Figura 5.9: Acumulación de los rayos para el primer pico de la simulación de la escena 1D. Distribución Halton.

Escena 2D

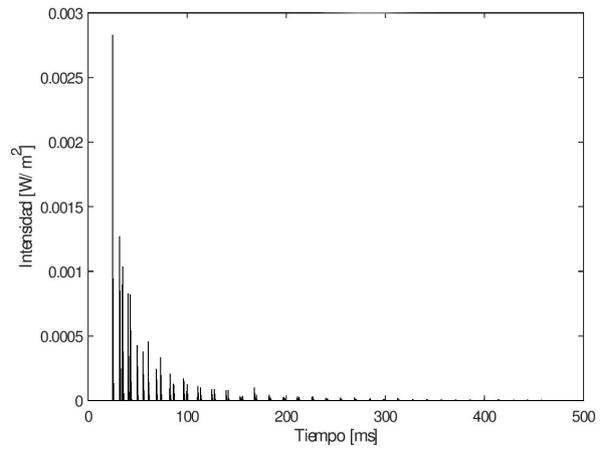
En esta escena el efecto de utilizar menos rayos es más visible que en la escena anterior. Tanto en la Figura 5.10 como en la 5.11, puede apreciarse como las simulaciones con 10^5 rayos presentan picos que “sobresalen”, por ejemplo, aproximadamente a los 150 *ms* en la Figura 5.11a. También puede observarse en menor medida comparando las simulaciones de 10^6 y 10^7 rayos en la distribución pseudoaleatoria (Figura 5.10b entre los 150 *ms* y 200 *ms*).

Escena 3D

En esta escena pueden observarse los mismos fenómenos que en las escenas anteriores. Adicionalmente puede verse cómo la intensidad de la señal per-



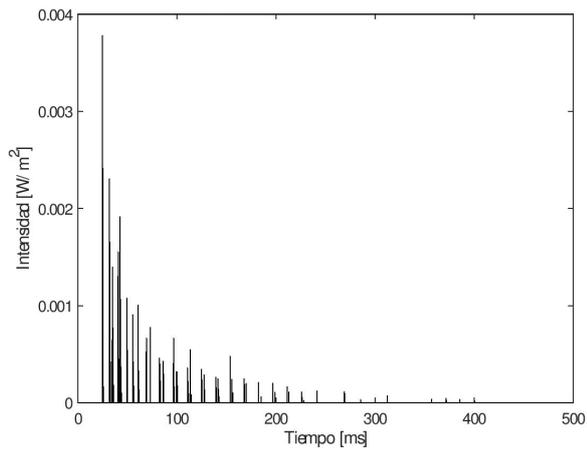
(a) Respuesta al Impulso simulada
Cantidad de rayos: 10^5



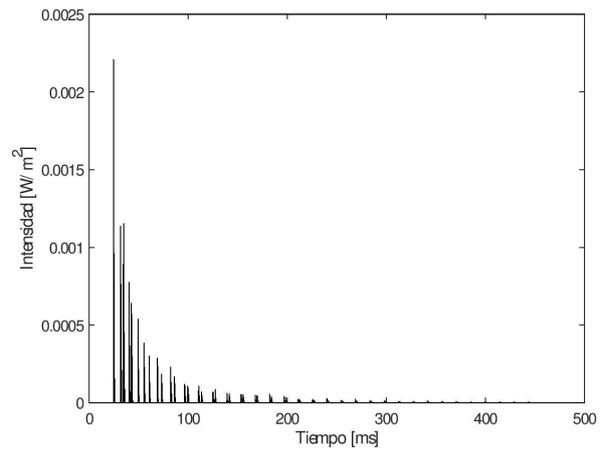
(b) Respuesta al Impulso simulada
Cantidad de rayos: 10^6

Figura 5.10: Simulaciones de la escena 2D. Distribución pseudoaleatoria.

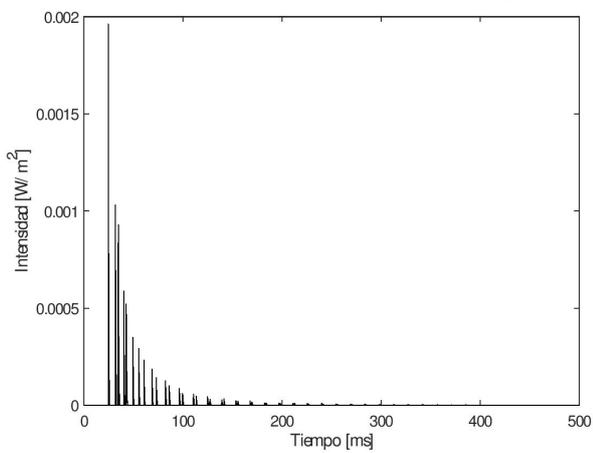
dura al final. Incluso comparando las simulaciones de 10^6 y 10^7 rayos, puede apreciarse una pequeña diferencia entre los 300 *ms* y 400 *ms*.



(a) Respuesta al Impulso simulada
Cantidad de rayos: 10^5

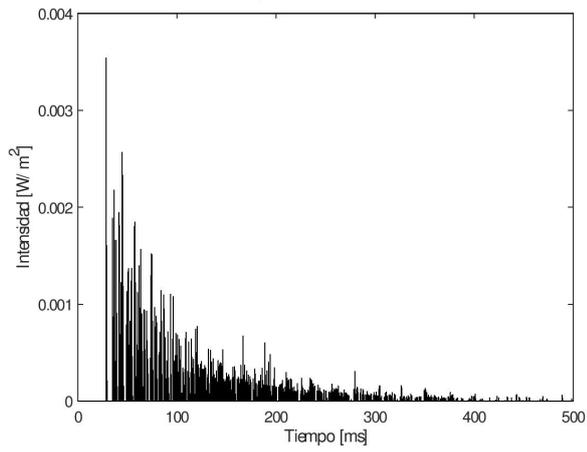


(b) Respuesta al Impulso simulada
Cantidad de rayos: 10^6

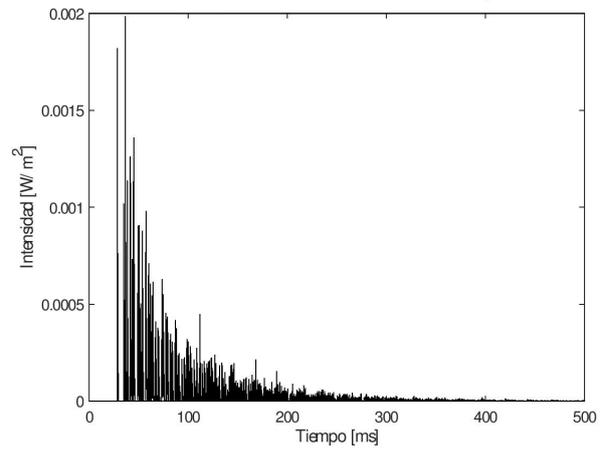


(c) Respuesta al Impulso simulada
Cantidad de rayos: 10^7

Figura 5.11: Simulaciones de la escena 2D. Distribución Halton.

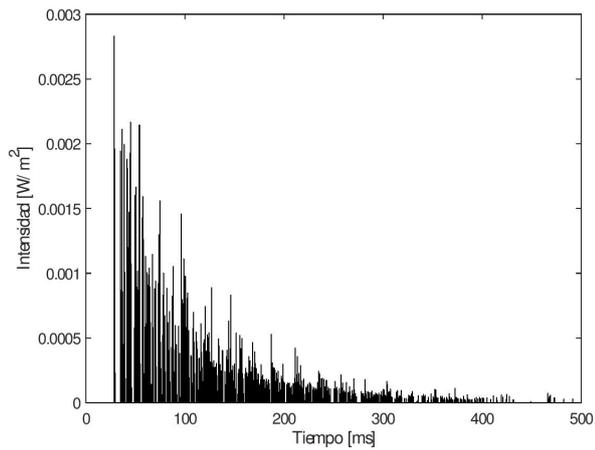


(a) Respuesta al Impulso simulada
Cantidad de rayos: 10^5

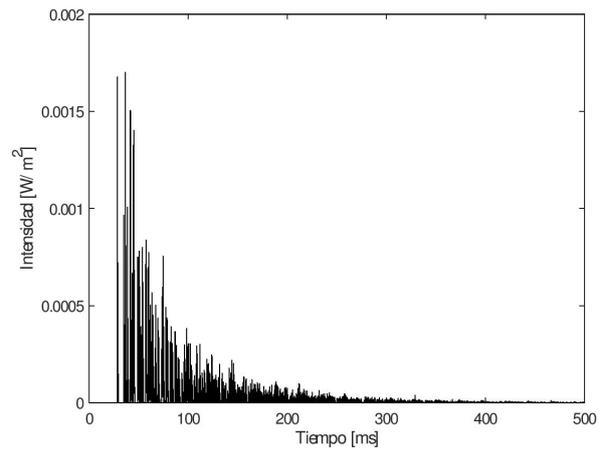


(b) Respuesta al Impulso simulada
Cantidad de rayos: 10^6

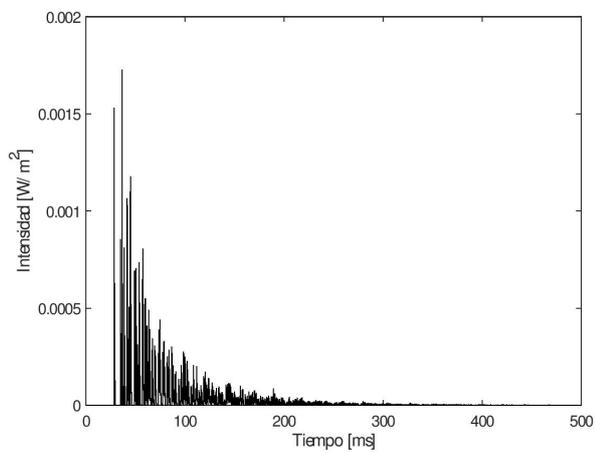
Figura 5.12: Simulaciones de la escena 3D. Distribución pseudoaleatoria.



(a) Respuesta al Impulso simulada
Cantidad de rayos: 10^5



(b) Respuesta al Impulso simulada
Cantidad de rayos: 10^6



(c) Respuesta al Impulso simulada
Cantidad de rayos: 10^7

Figura 5.13: Simulaciones de la escena 3D. Distribución Halton.

5.3.4. Análisis cuantitativo

En esta sección se presentan y analizan los resultados obtenidos del nivel sonoro y el tiempo de reverberación (ver Sección 2.1). Con estos resultados se pretende evaluar a la IR de una forma cuantificable y tener una noción del error obtenido. Se mantienen los parámetros obtenidos para la cantidad de rebotes y coeficiente de absorción obtenidos en el análisis anterior. En la Tabla 5.4 se presentan los valores del nivel sonoro y Early Decay Time (EDT) (Ver Sección 2.1.9 y Ecuación 2.10) para las mediciones de todas las escenas. En las Tablas 5.5, 5.6 y 5.7 se presentan los mismos parámetros calculados para las simulaciones. Además también se muestra el error del EDT que se calcula como el valor absoluto de la diferencia del valor de la medición y el valor de la simulación. En este caso el EDT se obtiene calculando el tiempo que la señal demora en disminuir su nivel sonoro en 10 dB (t_{EDT}) y estimando el EDT de forma lineal como se explicó en la Sección 2.1.9 ($EDT = 6 \cdot t_{EDT}$).

Se realiza el cálculo del nivel sonoro de toda la señal de la misma forma que en [24], es decir que se sustituye la sumatoria de todos los valores de la IR en la Ecuación 2.6.

Escena	$L_p(dB)$	$EDT(ms)$
1D	104,1	0,251
2D	110,31	0,798
3D	115,42	1,017

Tabla 5.4: Nivel sonoro y EDT de las mediciones.

Escena	Distribución	Cant. Rayos	$L_p(dB)$	$EDT(ms)$	EDT Error
1D	Uniforme	10^5	105,62	0,234	0,017
1D	Uniforme	10^6	108,48	0,234	0,017
1D	Uniforme	10^7	108,61	0,234	0,017
1D	Halton	10^5	104,99	0,234	0,017
1D	Halton	10^6	107,74	0,234	0,017
1D	Halton	10^7	108,63	0,234	0,017

Tabla 5.5: Nivel sonoro y EDT de las simulaciones. Escena 1D.

Escena	Distribución	Cant. Rayos	$L_p(dB)$	$EDT(ms)$	EDT Error
2D	Uniforme	10^5	109,31	0,440	0,358
2D	Uniforme	10^6	111,14	0,441	0,357
2D	Uniforme	10^7	112,78	0,495	0,303
2D	Halton	10^5	109,97	0,497	0,301
2D	Halton	10^6	112,29	0,495	0,303
2D	Halton	10^7	112,79	0,495	0,303

Tabla 5.6: Nivel sonoro y EDT de las simulaciones. Escena 2D.

Escena	Distribución	Cant. Rayos	$L_p(dB)$	$EDT(ms)$	EDT Error
3D	Uniforme	10^5	116,13	1,006	0,011
3D	Uniforme	10^6	118,48	1,0070	0,010
3D	Uniforme	10^7	119,25	1,001	0,016
3D	Halton	10^5	117,03	1,015	0,002
3D	Halton	10^6	119,22	1,000	0,017
3D	Halton	10^7	119,15	0,999	0,018

Tabla 5.7: Nivel sonoro y EDT de las simulaciones. Escena 3D.

Para facilitar la lectura, los valores para el EDT se redondearon para usar 3 dígitos después de la coma. Como puede verse en la Tabla 5.5, los valores para el EDT son idénticos hasta el tercer dígito luego de la coma. Sin embargo, cabe aclarar que dichos valores presentan diferencias a partir del 5 dígito luego de la coma.

En general los errores del EDT son bastante bajos, al rededor del 1%. Sin embargo, puede observarse un error mayor para la escena 2D, lo que condice con lo observado en la Sección 5.3.1 (ver Figura 5.4). Por otro lado, no se observa una relación directa entre la cantidad de rayos utilizados y el error obtenido, en algunos casos, una menor cantidad de rayos produce un error menor. Lo mismo puede observarse en cuanto a las distribuciones, ninguna presenta mejores resultados que la otra para todos los casos. Sin embargo, puede observarse que para 10^7 rayos los errores son muy parecidos para ambas distribuciones.

A continuación se presentan las diferencias entre los niveles sonoros. Como se mencionó en la Sección 2.2.4, se comparan las diferencias obtenidas de los niveles sonoros de las mismas escenas.

1D (Referencia)	2D (2D-Ref)	3D (3D-Ref)
104,1	110,31 (+6,21)	115,42 (+11,32)

Tabla 5.8: Variación del nivel sonoro entre las mediciones.

Distribución	Cant. Rayos	1D (Referencia)	2D (2D-Ref)	3D (3D-Ref)
Uniform	10^5	105,62	109,31 (+3,69)	116,13 (+10,51)
Uniform	10^6	108,48	111,14 (+2,66)	118,48 (+10)
Uniform	10^7	108,61	112,78 (+4,17)	119,25 (+10,64)
Halton	10^5	104,99	109,97 (+4,98)	117,03 (+12,04)
Halton	10^6	107,74	112,29 (+4,55)	119,22 (+11,48)
Halton	10^7	108,63	112,79 (+4,16)	119,15 (+10,52)

Tabla 5.9: Variación del nivel sonoro entre las simulaciones.

En la Tabla 5.10 se muestran los errores de los niveles sonoros calculados. El error se calcula tomando el aumento del nivel sonoro respecto a la escena 1D (lo que está entre paréntesis en las Tablas 5.8 y 5.9) y calculando el valor absoluto de la diferencia de este valor para la medición y la simulación correspondiente.

Distribución	Cant. Rayos	Error 2D - 1D	Error 3D - 1D
Uniform	10^5	2,52	0,81
Uniform	10^6	3,55	1,32
Uniform	10^7	2,04	0,68
Halton	10^5	1,23	0,72
Halton	10^6	1,66	0,16
Halton	10^7	2,05	0,8

Tabla 5.10: Error de la variación del nivel sonoro.

Nuevamente los errores son bajos, aunque más altos para la escena 2D. En cuanto a la cantidad de rayos y distribución utilizada se observa lo mismo que para el EDT, es decir no son conclusivos, aunque puede observarse como convergen a valores similares al aumentar la cantidad de rayos.

5.4. Análisis de la auralización

En cuanto a la auralización se probaron los 2 modos: i) con una señal directa, desde un micrófono y ii) con una señal cargada desde archivo. Para la señal de archivo se utilizó la grabación de una guitarra clásica. La grabación tiene una duración de aproximadamente 5 segundos con un *sample rate* de 16.000 *Hz*.

Los resultados de la auralización coinciden con lo observado en la Sección 5.3. Al aumentar la cantidad de paredes en la escena, se puede percibir el eco de forma más clara. Esto es incluso más notorio al disminuir el coeficiente de absorción. Para un coeficiente de absorción de 0,01 la escena 3D produce un sonido similar al de un galpón. Para el caso del sonido tomado desde un micrófono, se percibe un eco claro en las palabras y otros sonidos como aplausos o clicks de un mouse. Este mismo efecto puede apreciarse en el caso del sonido de una guitarra tomado desde un archivo, creando una sensación similar al sonido en un escenario.

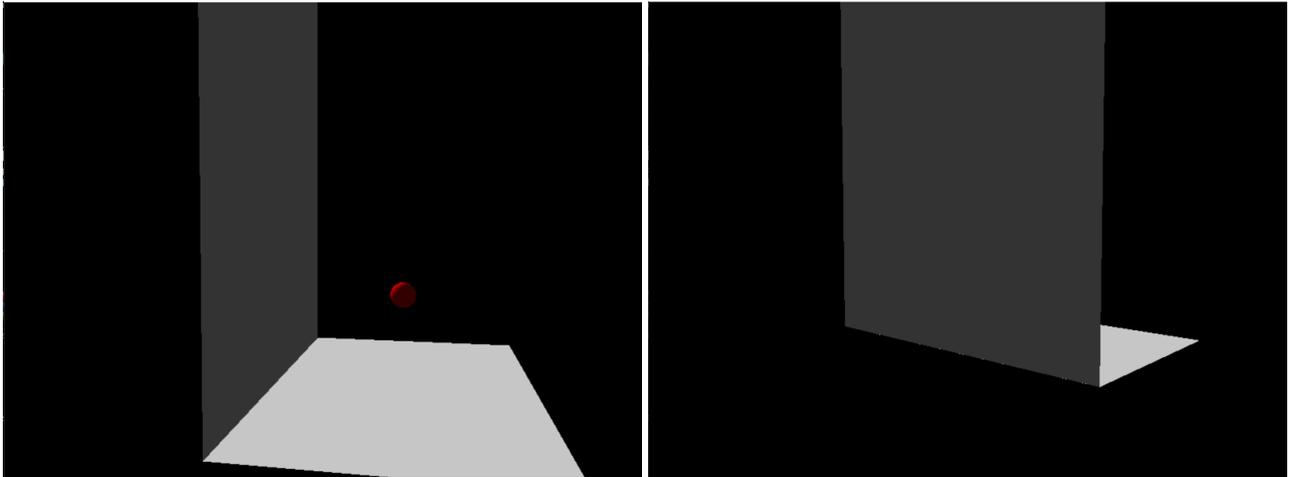
El eco también afecta al ruido de la señal. A medida que se disminuye el coeficiente de absorción puede percibirse un ruido de fondo con mayor intensidad que opaca el sonido de interés. Esto es de esperarse y puede mitigarse usando un micrófono mejor calibrado de forma de no detectar el sonido del ambiente.

Por otro lado, alejarse o acercarse a la fuente influye de manera notoria en la intensidad percibida del sonido. Si el oyente se aleja lo suficiente de la fuente, el sonido deja de percibirse.

Debido a que el modelo de este trabajo no contempla la difracción, muchos de los caminos directos no son hallados. Esto puede percibirse claramente cuando la fuente es ocluida por algún elemento de la escena. Por ejemplo, si el oyente se posiciona detrás de la pared en la escena 1D, no se escuchará ningún sonido. Ese ejemplo puede verse de forma gráfica en la Figura 5.14.

5.4.1. Cuellos de botella

Para poder experimentar la auralización en tiempo real, es necesario que los cálculos se realicen a una velocidad que permita percibir la señal saliente de forma ininterrumpida. Si bien los cálculos de la propagación de los rayos se hacen solo 1 vez, los cálculos de la convolución deben hacerse de forma frecuente. Por



(a) Fuente visible. El sonido se percibe sin problemas. (b) Fuente oculta. El sonido deja de percibirse.

Figura 5.14: Diferencias entre una fuente visible y oculta para el usuario.

ejemplo, con un *sample_rate* de 16 *KHz* y *cantidad_de_samples_por_llamada* de 512, la función de *callback* deberá llamarse aproximadamente $\frac{16000}{512} \approx 31$ veces por segundo.

Como se mencionó en la Sección 4.3.1, el tamaño de la matriz ρ aumenta junto con la duración del historial de *samples* y junto con el *sample rate*. Para un historial de 1 segundo de largo, el mayor *sample rate* que fue posible utilizar sin experimentar cortes en el audio fue de 16.000 *Hz*.

5.5. Tiempos de ejecución

En las Tablas 5.11 y 5.12 se presentan los tiempos de ejecución de las simulaciones.

Es de esperarse que para la misma escena, los tiempos de ejecución aumenten junto con la cantidad de rayos emitidos. También puede apreciarse una diferencia en el tiempo de ejecución para distintas escenas con la misma cantidad de rayos. Al aumentar la cantidad de paredes en una escena, mayor será la probabilidad de encontrar una intersección con ésta, esto conduce a más llamadas recursivas en el Algoritmo 2.

Por otro lado, también puede observarse que los tiempos obtenidos con Halton son visiblemente mayores que los tiempos obtenidos con la secuencia pseudoaleatoria. En particular puede observarse que para la escena 1D los tiempos de ejecución son entre 21 y 30 veces más grandes en Halton, y en

la escena 2D entre 12 y 18 veces más grandes, y para la escena 3D entre 6 y 9 veces más grandes. Esta disminución en la proporción de los tiempos de ejecución pueden deberse al aumento de las reflexiones antes mencionados. A medida que el número de reflexiones aumenta, el tiempo del computo de los caminos de audio pasa a ser una parte mayor del tiempo de ejecución total.

Escena	Cant. Rayos	Tiempo[ms]
1D	10^5	34
1D	10^6	312
1D	10^7	3107
2D	10^5	65
2D	10^6	510
2D	10^7	5227
3D	10^5	134
3D	10^6	1200
3D	10^7	12347

Tabla 5.11: Tiempo medido de la simulación. Distribución pseudoaleatoria.

Escena	Cant. Rayos	Tiempo[ms]
1D	10^5	717
1D	10^6	8745
1D	10^7	91675
2D	10^5	761
2D	10^6	8024
2D	10^7	94481
3D	10^5	815
3D	10^6	8875
3D	10^7	109690

Tabla 5.12: Tiempo medido de la simulación. Distribución Halton.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones sobre los resultados obtenidos de este trabajo en base a la investigación realizada y a las métricas estudiadas.

6.1. Conclusiones

Como se mencionó en el Capítulo 3, el objetivo principal de este trabajo fue investigar y documentar todos los pasos necesarios para resolver el problema de la simulación y la auralización utilizando el trazado de rayos. El programa desarrollado es la consolidación del trabajo de investigación inicial, y la documentación del proceso detalla todos los pasos seguidos en la resolución del problema. Los resultados de la propuesta, muestran que se trata de una implementación simple y con resultados viables.

Se constató que el tipo de escena juega un rol importante en cómo se debe modelar la propagación del sonido. En escenas cerradas simples, el modelo propuesto resuelve bien el problema de la propagación, esto se debe a que al tratarse de un espacio cerrado, la probabilidad de que un rayo llegue al receptor a partir de reflexiones puramente especulares es alta. Sin embargo, en casos de escenas abiertas o escenas donde la fuente puede estar ocluida para el oyente, resulta imprescindible modelar los fenómenos de difracción y transmisión para lograr un mayor realismo en la simulación y auralización del sonido.

La simulación y la auralización son problemas distintos pero relacionados. Los problemas que conciernen a ambos son bastante diferentes, siendo la auralización un problema donde la física queda en segundo plano, pero al analizar el resultado de ambos problemas en forma conjunta se obtiene una visión comple-

ta de la solución. La relevancia de los resultados de la simulación y auralización dependen del problema a resolver y si se debe poner el foco en “realismo” o en lo físicamente correcto. Por ejemplo, para el caso de un videojuego donde se quiere simular un efecto de eco de una cueva o una catedral, alcanzaría con utilizar una IR prefabricada para crear una sensación de realismo, por más que dicha IR no coincida exactamente con el espacio en cuestión. En este caso, el uso de trazado de rayos sería una sobre ingeniería del problema.

Los resultados cualitativos de la simulación son variados. Los picos correspondientes al sonido directo y las reflexiones tempranas coinciden temporalmente con los mismos picos en las mediciones realizadas. Sin embargo, los gráficos de la IR y las mediciones divergen a medida que la complejidad de la escena aumenta. Por otro lado, al modificar el coeficiente de absorción y la cantidad de rebotes, la proporción de los picos varía de forma acorde. Es decir, al disminuir el coeficiente de absorción y aumentar al cantidad de rebotes los picos correspondientes a las reflexiones tempranas y a las reverberaciones, son visiblemente más grandes. Esto permitió encontrar coeficientes que se ajustan a las escenas estudiadas.

Las métricas estudiadas sirven para cuantificar el error. Pudo observarse que el incremento del nivel sonoro de la IR simulada, se asemeja en cierta medida al aumento en las mediciones. El error más grande se observó en la escena 2D con una diferencia de entre $1,16 \text{ dB}$ y $3,55 \text{ dB}$, esta diferencia puede deberse a la naturaleza de la escena 2D. Es posible que la escena 2D tenga un nivel de ruido mayor al resto de las escenas al tratarse de una zona residencial (Figura 5.1). Esto también puede verse en la Figura 5.4a, notar como la intensidad se mantiene de forma visible hasta el final de la medición. Por otro lado, la escena 3D presenta errores menores de entre $0,16 \text{ dB}$ y $1,32 \text{ dB}$. Esta diferencia resulta imperceptible a efectos prácticos ya que se trata de un error relativo que corresponde al 1% de los valores medidos.

Al analizar la diferencia de niveles sonoros en función de la cantidad de rayos y de la distribución utilizada se observan resultados diferentes. En particular puede observarse que para la escena 2D los mejores resultados se dan con 10^5 rayos en Halton, mientras que para la distribución pseudoaleatoria se logran con 10^7 rayos. Por otro lado, para la escena 3D los mejores resultados se dan con 10^7 rayos en Halton, mientras que para la distribución pseudoaleatoria se logran con 10^6 rayos. Puede concluirse entonces, que las distribuciones convergen de forma diferente a la solución.

Al utilizar la distribución pseudoaleatoria pueden observarse diferencias entre distintas ejecuciones del programa ya que la semilla es diferente cada vez. Estas diferencias disminuyen a medida que el número de rayos utilizados aumenta. Esto no ocurre con Halton ya que los números de la secuencia están predeterminados.

El tiempo de reverberación admite el mismo análisis que el nivel sonoro. Los errores de las escenas 1D y 3D son mucho más pequeños que los errores de la escena 2D. Nuevamente, esto puede deberse al ruido de la escena 2D. En general puede observarse que la cantidad de rayos utilizada no hace una diferencia sustancial al EDT de las simulaciones.

El programa implementado tiene utilidad como herramienta de trabajo. Como pudo verse en la parte experimental, fue posible realizar pruebas con distintos parámetros para poder estudiar una escena bajo distintas condiciones.

Por último, si bien el *buffer* circular cumple su cometido de evitar reescribir todos los *samples* del historial, se vio que el costo del cálculo de la convolución es mucho más importante. Como en el entorno de prueba utilizado solo se logró hacer la convolución en tiempo real con un historial de 1 segundo y un *sample rate* de 16 *KHz*, el beneficio del *buffer* circular se vio reducido.

6.2. Trabajo futuro

En cuanto a trabajo futuro, se presentan 2 tipos de líneas de trabajo. Una de ellas para continuar la implementación del programa generado, mejorando sus resultados y su uso como herramienta de trabajo. La otra consiste en la investigación e implementación de técnicas alternativas y resolución de problemas adyacentes que surgen durante la investigación de este trabajo, pero que exceden el alcance del mismo.

Muchos de los conceptos que surgieron de la investigación inicial realizada no pudieron incluirse en el alcance de la solución por restricciones de tiempo. En esta sección se detallan aquellos aspectos más importantes que podrían profundizarse para mejorar la solución propuesta.

6.2.1. Modelado de la escena

Las escenas modeladas por el programa implementado se limitan a 1 oyente y una fuente. Adicionalmente, solo se permite un coeficiente de absorción para

toda la escena.

El programa puede ser fácilmente extendido para incluir más de una fuente y un oyente dado que solo implica duplicar los cálculos hechos. Por otro lado, es posible asignar un coeficiente de absorción distinto a cada triángulo de la escena. Si el archivo OBJ tiene información de los materiales del modelo, por ejemplo color, el programa puede extenderse para guardar esta información y pasarla a Embree. Luego, utilizando Embree es posible definir algún tipo de relación entre color y coeficiente de absorción. El color es representado internamente con 8 bits para cada color RGB (rojo, verde y azul) y la transparencia, esos bits pueden interpretarse como un número en punto flotante de 32 bits.

6.2.2. Difracción, transmisión y frecuencias

Varios de los aspectos introducidos en el Capítulo 2 no fueron considerados para el modelo propuesto. El modelado de estos aspectos podría ayudar a mejorar los resultados tanto de la simulación como de la auralización.

Como ya se mencionó brevemente, reducir la propagación del sonido a reflexiones puramente especulares presenta limitaciones dependiendo de la escena estudiada. Incluir los fenómenos de transmisión y difracción aumentarían significativamente el realismo percibido en la auralización.

Otro aspecto no modelado en este trabajo es la composición del sonido por distintas frecuencias. Cada frecuencia tiene un coeficiente de reflexión distinto para cada material, lo que genera distintas IR para cada frecuencia. Una posible manera de implementar esto es utilizando bandas de octavas donde se definen 8 rangos de frecuencia distintos, esto genera 8 IR distintas (1 para cada rango de frecuencias). Luego, utilizando la transformada de Fourier es posible separar una señal de audio en sus respectivas frecuencias, hacer las 8 convoluciones y volver hacer la inversa de Fourier para generar la señal de audio de salida.

6.2.3. Ancho de los picos

Es posible estudiar cómo el ancho de los picos afecta al nivel sonoro, EDT y a la señal resultante de la convolución. Como se mencionó en la Sección 5.3.2, la duración de los picos se acorta si se reduce el diámetro de la esfera. Sin embargo, reducir el tamaño de la esfera también significa una menor probabilidad de encontrar caminos hasta el receptor.

Una forma posible de “afinar” los picos es llevando todos los caminos que pertenecen al mismo “frente de onda” al mismo tiempo t_0 en la IR, como puede observarse en la Figura 6.1.

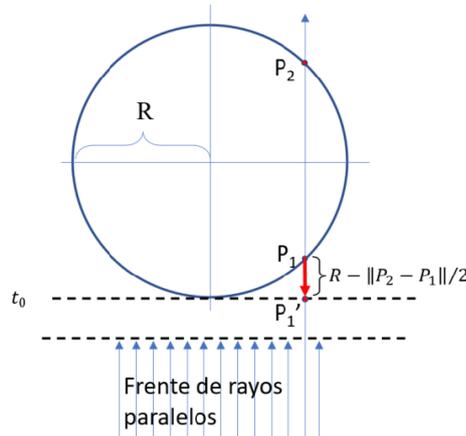


Figura 6.1: Se pueden retroceder los puntos de intersección de los diferentes rayos pertenecientes al mismo frente, para que todos intersequen en el mismo instante t_0 .

6.2.4. Fourier como solución de la convolución

Una particularidad de la matriz ρ de la Ecuación 4.4 es que cumple las propiedades de una matriz de Toeplitz. Una matriz de Toeplitz es aquella cuyas diagonales están compuestas por un mismo elemento.

A partir de una matriz de Toeplitz A , es posible realizar el cálculo de la convolución utilizando la transformada de Fourier y su inversa. Strang [45] expone un método para realizar el cálculo del producto $A \cdot x$ construyendo una matriz circular C a la que se le aplica la transformada de Fourier. Cuando se calculan eficientemente, la transformada de Fourier se denomina FFT (del inglés Fast Fourier Transform), y a su operación inversa se la denomina IFFT. Este método permite calcular el producto matriz-vector de la Ecuación 4.4 de forma más eficiente ($O(n \cdot \log(n))$) en lugar del $O(n^2)$ resultante de trabajar directamente con la matriz y el vector). En la Tabla 6.1 se presentan unos resultados iniciales de los tiempos de ambos métodos.

En consecuencia, utilizando este método es posible realizar la auralización a un *sample rate* mayor.

Dimensiones de A	Tiempo producto usual[s]	Tiempo Fourier[s]
5x5	0,0000183	0,0000663
50x50	0,0000274	0,000109
500x500	0,000221	0,000179
5000x5000	0,00858	0,000998
10.000x10.000	0,0381	0,00283
20.000x20.000	0,169	0,00649
40.000x40.000	0,663	0,0191

Tabla 6.1: Tiempo medido del producto matriz vector. Cálculos hechos en MATLAB.

6.2.5. Soluciones a problemas alternativos

Dada la forma de calcular la convolución presentada en la Sección 4.3.1, resulta interesante investigar si el mismo planteo es aplicable para otro tipo de problemas.

Dependiendo de qué datos iniciales se conocen, la Ecuación 4.4 puede tener distintos usos. Si se conocen s' y ρ sería posible hallar la señal original s resolviendo el sistema lineal resultante. Puede verse que, dada la naturaleza del problema, muchas de las entradas del vector de la IR son nulas. En particular, las primeras entradas del vector IR siempre serán nulas ya que incluso el camino directo tiene un tiempo de propagación mayor a 0. Sin embargo, es posible reducir el problema tomando solo la porción del vector a partir del camino directo. El sistema que resulta de tomar esta sub-matriz con los elementos de su diagonal distintos de 0 es determinado, es decir, tiene una única solución.

Otro posible problema a estudiar es el de realizar la “deconvolución”, es decir, determinar la IR conociendo la señal original s y la señal de salida s' , evitando así el trabajo de modelar la escena y aplicar el trazado de rayos. Una forma de hacer esto es considerando las matrices D_i que presentan 1s en la diagonal i . En la Figura 6.2 pueden verse la matrices D_i de dimensión 3x3.

$$D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad D_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad D_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Figura 6.2: Matrices D_i de dimensiones 3x3.

Utilizando estas matrices la Ecuación 4.4 puede escribirse como:

$$s' = (D_1 \cdot IR_1 + D_2 \cdot IR_2 + \dots + D_n \cdot IR_n) \cdot s \quad (6.1)$$

Donde IR_i es la celda i del vector IR . Luego, la ecuación anterior puede escribirse como:

$$s' = [D_1 \cdot s, D_2 \cdot s, \dots, D_n \cdot s] \cdot IR = \sigma \cdot IR \quad (6.2)$$

Donde $D_i \cdot s$ es el vector columna i de la matriz σ .

$$\sigma = \begin{pmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ s_2 & s_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ s_{n-1} & s_{n-2} & s_{n-3} & \dots & s_1 & \vdots \\ s_n & s_{n-1} & s_{n-2} & \dots & s_2 & s_1 \end{pmatrix} \quad (6.3)$$

Luego, este sistema puede resolverse como cualquier otro para encontrar el vector IR . Análogamente al problema anterior, si las primeras entradas del vector s son nulas, se puede tomar la sub-matriz más grande cuya diagonal sea distinta de 0 para que el sistema tenga solución.

Vale aclarar que estas operaciones coinciden del punto de vista algebraico con el de la convolución, y por tanto, también pueden resolverse utilizando la combinación adecuada de FFT e IFFT (ver Sección 6.2.4).

6.2.6. Extensión del programa como herramienta de trabajo

Si bien el programa implementado es útil como herramienta de trabajo, el mismo puede ser extendido para mejorar el flujo de trabajo y las pruebas realizadas. Una posible mejora podría ser incluir una interfaz gráfica que permita modificar los parámetros y visualizar los resultados dentro del programa. Para este propósito podría utilizarse la biblioteca Dear ImGui [22], una biblioteca de C++ bastante utilizada en la industria de los videojuegos para realizar *profiling*.

Lista de figuras

2.1	Desplazamiento de una partícula. Gráfico tomado de [25].	6
2.2	Las regiones del medio donde las partículas se acumulan corresponden con las crestas de la onda. Gráfico tomado de [25].	7
2.3	Ilustración de una onda y sus propiedades. Gráfico tomado de [27].	8
2.4	Las flechas indican la dirección de la propagación de la energía en una onda longitudinal esférica.	9
2.5	<i>Path tracing</i> . Los rayos se propagan desde la cámara hasta la fuente de luz. Gráfico tomado de [12].	10
2.6	<i>Path tracing</i> para audio. Los rayos se propagan desde la fuente de sonido hasta el receptor.	11
2.7	Posibles formas de reflexión de la luz con superficies.	12
2.8	Tipos de reflexión dependiendo de la frecuencia de la onda. Donde: f es la frecuencia de la onda incidente, c la velocidad del sonido, h y a la altura y el ancho de las irregularidades de la superficie. Gráfico tomado de [41].	13
2.9	(A) Un frente de onda con una longitud de onda pequeña continuará su propagación con cambios leves en su dirección. (B) Un frente de onda con longitud de onda mayor ve su propagación modificada de forma más pronunciada. Gráfico tomado de [25].	14
2.10	(A) Una apertura grande con respecto a la longitud de onda dejará pasar los frentes de onda con pocas modificaciones. (B) Si la apertura es pequeña con respecto a la longitud de onda, los frentes de onda que atraviesen la apertura se comportarán como nuevas fuentes. Gráfico tomado de [25].	14

2.11	Los rayos de sonido son refractados cuando atraviesan a un medio con distinta velocidad de propagación. Notar que el frente de onda A-B no es paralelo con el frente de onda C-D. Gráfico tomado de [25].	15
2.12	Transmisión del sonido. Donde I_i es la intensidad incidente, I_r la intensidad reflejada, I_t la intensidad transmitida e I_w la intensidad absorbida. Gráfico tomado de [41].	16
2.13	La HRTF muestra como una frecuencia es percibida de forma diferente dependiendo del ángulo de incidencia al oído. Gráfico tomado de [25].	19
2.14	Gráfica de la sonoridad en función de la frecuencia y el nivel sonoro. Notar que para una frecuencia de 1 <i>KHz</i> la sonoridad y el nivel sonoro coinciden. Gráfico tomado de [25].	20
2.15	Respuesta al impulso de un cuarto. Gráfico tomado de [41].	21
2.16	Ejemplos de curvas de decaimiento. (A) Se puede registrar una disminución de 60 <i>dB</i> en el nivel sonoro, el tiempo de reverberación se puede medir directamente. (B) El nivel de la señal cae por debajo del ruido de fondo antes de registrarse la disminución de 60 <i>dB</i> y por consiguiente el tiempo de reverberación se estima a partir de la pendiente de la curva. Gráfico tomado de [25].	23
2.17	El tiempo de reverberación puede estimarse a partir de distintas secciones de la curva de decaimiento. Gráfico tomado de [9].	24
2.18	Gráfica de la función $\text{sen}(t)$ donde t se mide en segundos. La función es <i>sampleada</i> con distintas frecuencias. A medida que aumenta el <i>sample rate</i> las gráficas se parecen más a la gráfica de la función $\text{sen}(x)$	26
2.19	Ejemplo de una señal expresada en dominio de tiempo y dominio de frecuencias. La señal está compuesta por ondas de frecuencias de 5, 12 y 47 <i>Hz</i> . Gráfico tomado de [38].	29
2.20	Esquema del SDK de VRWorks Audio. Gráfico tomado de [4].	31
2.21	Visualización de la escena en Wayverb. Gráfico tomado de [48].	32
3.1	Módulo principal.	39
3.2	Módulo de geometría.	39

3.3	Cada módulo se encarga de pasar la información necesaria al módulo siguiente.	40
3.4	Módulo de traza de rayos.	41
3.5	Módulo de audio.	42
4.1	Vector unitario en coordenadas esféricas.	46
4.2	Propagación de los rayos de sonido. Los datos se actualizan en cada reflexión.	47
4.3	Intersección Rayo/Esfera. t_0 y t_1 , las soluciones del sistema, miden la distancia desde el origen del rayo a cada punto de intersección. Gráfico tomado de [2].	49
4.4	Las IRs tienen distinta cantidad de elementos para distintos <i>sample rates</i> . Para 16 <i>KHz</i> un camino de duración 0,5 s contribuye a la celda 8000, mientras que para 48 <i>KHz</i> contribuye a la celda 24.000.	50
4.5	2 caminos distintos con distinto tiempo de propagación $T_2 > T_1$. Un cierto <i>sample</i> s_i emitido en el instante t_i llega al receptor en el instante $t_i + T_1$ y nuevamente en el instante $t_i + T_2$	51
4.6	En el instante de tiempo t_0 el <i>sample</i> s_0 llega al receptor a través del camino c_0	52
4.7	En el instante de tiempo t_1 el <i>sample</i> s_0 llega al receptor a través del camino c_1 , mientras que s_1 llega a través de c_0	53
4.8	Cada <i>sample</i> se corresponde con un camino cuyo tiempo de propagación es igual a la antigüedad de dicho <i>sample</i>	53
4.9	Ejemplos de buffer circular.	56
4.10	La ventana se desplaza a lo largo del <i>buffer</i> circular para crear el efecto de una señal recibida en tiempo real.	58
5.1	Escenas estudiadas. S marca la posición de la fuente (<i>source</i>) y R marca la posición del receptor (<i>receiver</i>). Imágenes tomadas de [24].	60
5.2	Modelos CAD de las escenas estudiadas.	61
5.3	Escena 1D.	63
5.4	Escena 2D.	65
5.5	Escena 3D.	66
5.6	Formas del primer pico en la escena 1D. Gráficos normalizados.	67
5.7	Simulaciones de la escena 1D. Distribución pseudoaleatoria.	68

5.8	Simulaciones de la escena 1D. Distribución Halton.	69
5.9	Acumulación de los rayos para el primer pico de la simulación de la escena 1D. Distribución Halton.	70
5.10	Simulaciones de la escena 2D. Distribución pseudoaleatoria.	71
5.11	Simulaciones de la escena 2D. Distribución Halton.	72
5.12	Simulaciones de la escena 3D. Distribución pseudoaleatoria.	73
5.13	Simulaciones de la escena 3D. Distribución Halton.	74
5.14	Diferencias entre una fuente visible y oculta para el usuario.	79
6.1	Se pueden retroceder los puntos de intersección de los diferentes rayos pertenecientes al mismo frente, para que todos intersequen en el mismo instante t_0	85
6.2	Matrices D_i de dimensiones 3x3.	86

Lista de tablas

5.1	Características del hardware utilizado	59
5.2	Características del entorno de desarrollo utilizado	59
5.3	Medidas de las escenas estudiadas.	60
5.4	Nivel sonoro y EDT de las mediciones.	75
5.5	Nivel sonoro y EDT de las simulaciones. Escena 1D.	75
5.6	Nivel sonoro y EDT de las simulaciones. Escena 2D.	76
5.7	Nivel sonoro y EDT de las simulaciones. Escena 3D.	76
5.8	Variación del nivel sonoro entre las mediciones.	77
5.9	Variación del nivel sonoro entre las simulaciones.	77
5.10	Error de la variación del nivel sonoro.	77
5.11	Tiempo medido de la simulación. Distribución pseudoaleatoria. .	80
5.12	Tiempo medido de la simulación. Distribución Halton.	80
6.1	Tiempo medido del producto matriz vector. Cálculos hechos en MATLAB.	86

Referencias bibliográficas

- [1] XML. <https://www.w3.org/TR/xml/> (2021/12/5), 2008.
- [2] Intersección de un rayo con una esfera. <https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-sphere-intersection> (2022/1/8), 2015.
- [3] Audio File Format Specifications. <http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html> (2021/11/19), 2017.
- [4] NVIDIA VRWorks. <https://developer.nvidia.com/vrworks/vrworks-audio> (2021/11/13), 2017.
- [5] OBJ. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml> (2021/12/12), 2020.
- [6] Tiny OBJ Loader. <https://github.com/tinyobjloader/tinyobjloader> (2021/12/12), 2020.
- [7] Embree. <https://www.embree.org/> (2021/11/24), 2021.
- [8] Frequencies for equal-tempered scale. <https://pages.mtu.edu/~suits/notefreqs.html> (2021/10/19), 2021.
- [9] Manual del programa Dirac de Acoustics Engineering. <https://www.acoustics-engineering.com/html/dirac.html> (2022/3/22), 2021.
- [10] OpenGL. <https://www.opengl.org/> (2021/11/27), 2021.
- [11] Secuencia Pseudoaleatoria en C++. https://en.cppreference.com/w/cpp/numeric/random/uniform_real_distribution (2022/4/23), 2022.

- [12] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, Angelo Pesce, Michał Iwanicki, and Sébastien Hillaire. Real-Time Rendering. CRC Press, 4th edition, 2018.
- [13] Adil Alpkocak and Kemal Sis. Computing Impulse Response of Room Acoustics Using the Ray-Tracing Method in Time Domain. Archives of Acoustics, 35(4):505–519, 2010.
- [14] Charles J. Ammon, Aaron A. Velasco, Thorne Lay, and Terry C. Wallace. Body waves and ray theory – travel times. In Foundations of Modern Global Seismology, chapter 12, pages 339–362. Academic Press, second edition, 2021.
- [15] Glen Ballou. Handbook for Sound Engineers. Taylor & Francis Ltd, 2008.
- [16] Gustavo Basso. Percepción auditiva. Editorial UNQ, 1st edition, 2018.
- [17] Benoit Beckers. Acústica Técnica I. http://heliodon.net/downloads/Beckers_2002_Acou_001_es_Nociones_de_acustica_tecnica.pdf (2022/3/23), 2002.
- [18] Benoit Beckers. Calcul par la méthode des images de la réflexion spéculaire dans des enceintes prismatiques convexes. http://www.heliodon.net/downloads/Beckers_Benoit_2017_Images.pdf (2021/10/23), 2017.
- [19] Laurence Bobis and James Lequeux. Cassini, Rømer, and the velocity of light. Journal of Astronomical History and Heritage, 11:97–105, 07 2008.
- [20] Claus Christensen, George Koutsouris, and Jens Rindel. The ISO 3382 parameters: Can we simulate them? Can we measure them? Akustika, 20, 06 2013.
- [21] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. In SIGGRAPH '81, 1981.
- [22] Omar Cornut. Dear ImGui. <https://github.com/ocornut/imgui> (2022/2/13), 2022.
- [23] Simon Cory. Distribución uniforme en una esfera. <http://corysimon.github.io/articles/uniformdistn-on-sphere/> (2022/1/7), 2015.
- [24] Inès de Bort and Benoit Beckers. Impact of multiple reflections on urban acoustics. Journal of Physics: Conference Series, 2042:012052, 11 2021.

- [25] F. Alton Everest and Ken C. Pohlmann. Master Handbook of Acoustics. The McGraw-Hill, 2009.
- [26] Eduardo Fernández. Generadores de puntos aleatorios uniformemente distribuidos en un círculo y estimación de sus discrepancias. 10.13140/RG.2.2.26814.15685/1 (2022/4/20), 2022.
- [27] Daniel Fleisch and Laura Kinnaman. A student’s guide to waves. Cambridge University Press, 2015.
- [28] María Gómez and Cano Alfaro. Ruido: Evaluación y acondicionamiento ergonómico. Instituto Nacional de Seguridad y Salud en el Trabajo (INSST), 2021.
- [29] Acoustics — Measurement of room acoustic parameters — Part 2: Reverberation time in ordinary rooms. Standard, International Organization for Standardization, Geneva, CH, June 2008.
- [30] Yuto Izumi and Makoto Otani. Relation between direction-of-arrival distribution of reflected sounds in late reverberation and room characteristics: Geometrical acoustics investigation. Applied Acoustics, 176:107805, 2021.
- [31] Joseph B. Keller. Geometrical theory of diffraction. Journal of the Optical Society of America, 52:116–30, 1962.
- [32] A. Krokstad, S. Storm, and S. Sørsdal. Calculating the acoustical room response by the use of a ray tracing technique. Journal of Sound and Vibration, 8(1):118–125, 1968.
- [33] Andrzej Kulowski. Algorithmic Representation of the Ray Tracing Technique. Applied Acoustics, 18(6):449–469, 1985.
- [34] E. Lakka, A. G. Malamos, K. G. Pavlakis, and Ware J. A. Spatial Sound Rendering – A Survey. International Journal of Interactive Multimedia and Artificial Intelligence, 5(3):33–45, 2018.
- [35] Qi Mo, Hengchin Yeh, Ming Lin, and Dinesh Manocha. Analytic ray curve tracing for outdoor sound propagation. Applied Acoustics, 104:142–151, 03 2016.

- [36] Matt Pharr, Wenzel Jakob, and Greg Humphreys. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2016.
- [37] Steve Rotenberg. Diapositivas de Random Numbers and Mappings. https://cseweb.ucsd.edu/classes/sp17/cse168-a/CSE168_07_Random.pdf (2022/4/10), 2017.
- [38] Kurt Bryan S. Allen Broughton. Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing. Wiley, 2nd edition, 2018.
- [39] Gary Scavone. RtAudio. <https://www.music.mcgill.ca/~gary/rtaudio/> (2021/11/27), 2021.
- [40] M.R. Schroeder. New Method of Measuring Reverberation Time. The Journal of the Acoustical Society of America, 37(3):409–412, 1964.
- [41] Dirk Schröder. Physically Based Real-Time Auralization of Interactive Virtual Environments. PhD thesis, 2011.
- [42] Stefano Sinigardi. Implementación de Halton. <https://github.com/cenit/jburkardt/tree/master/halton> (2022/1/5), 2018.
- [43] Steven W. Smith. The Scientist and Engineer’s Guide to Digital Signal Processing. <http://www.dspguide.com/> (2022/7/1), 2022.
- [44] Adam Stark. Audio File. <https://github.com/adamstark/AudioFile> (2022/1/4), 2017.
- [45] G. Strang. Linear Algebra and Learning from Data. Wellesley-Cambridge Press, 2019.
- [46] Joacim Stålberg and Mattias Ulmstedt. GPU Accelerated Ray-tracing for Simulating Sound Propagation in Water. PhD thesis, 2019.
- [47] Micah Taylor, Anish Chandak, Qi Mo, Christian Lauterbach, Carl Schissler, and Dinesh Manocha. Guided Multiview Ray Tracing for Fast Auralization. IEEE Transactions on Visualization and Computer Graphics, 18:1797 – 1810, 11 2012.
- [48] Reuben Thomas. Wayverb. <https://reuk.github.io/wayverb/> (2021/11/13), 2017.

- [49] Nicolas Tsingos, Ingrid Carlbom, Gary Elko, Robert Kubli, and Thomas Funkhouser. Validation of acoustical simulations in the "bell labs box". Computer Graphics and Applications, IEEE, 22:28 – 37, 08 2002.
- [50] Alan V. Oppenheim and Alan S. Willsky. Señales y sistemas. Prentice Hall, 2nd edition, 1998.
- [51] L.N. YANG and B.M. SHIELD. Development of a ray tracing computer model for the prediction of the sound field in long enclosures. Journal of Sound and Vibration, 229(1):133–146, 2000.

Apéndice A

Archivo de configuración

```
<?xml version="1.0"?>

<SCENE>
<MODEL>assets/models/1D_U.obj</MODEL>
<SIZE>1.0</SIZE> <!--Escala del modelo. Se utiliza 1 para conservar sus
                    dimensiones originales!-->
<MAX_REFLEXIONS>30</MAX_REFLEXIONS>
<ABSORPTION>0.0</ABSORPTION>
<NUM_RAYS>1000000</NUM_RAYS>
<SOURCE>
    <POWER>40</POWER> <!--Potencia de la fuente.
                    Medida en watts.-->
    <POS_X>0.0</POS_X>
    <POS_Y>0.0</POS_Y>
    <POS_Z>0.0</POS_Z>
</SOURCE>
<LISTENER>
    <SIZE>0.5</SIZE> <!--Radio de la esfera que representa
                    al oyente.-->
    <POS_X>-2.5</POS_X>
    <POS_Y>9.9</POS_Y>
    <POS_Z>0.0</POS_Z>
</LISTENER>
<MEASUREMENT>
```

```

<!-->Utilizado en el modo de simulacion<!-->
  <FILE>assets/measurements/1D_030_AllPass.wav</FILE>
  <LENGTH>500</LENGTH> <!-->Es el largo a considerar de la medición.
                          Medido en milisegundos.<!-->
</MEASUREMENT>
<ANALYZE>
  <!-->Utilizado para medir la cantidad de rayos recibidos en un cierto
  intervalo de tiempo. Begin y end marcan el inicio y el final del intervalo
  en milisegundos.<!-->
  <BEGIN>27</BEGIN>
  <END>30</END>
</ANALYZE>
<OUT_SAMPLERATE>16000</OUT_SAMPLERATE> <!-->Utilizado en el modo
                          de auralización. Medido en Hz.<!-->
<SOUND_SAMPLE>assets/sound_samples/guitar_sample_16k.wav</SOUND_SAMPLE>
<!-->Utilizado en la auralización. Ruta del archivo de audio a reproducir.
El sample rate del archivo debe coincidir con el out_samplerate.<!-->
</SCENE>

```