Módulo De Aurilización En Tiempo Real Para Dispositivos De Navegación Asistida Para Personas Con Discapacidad Visual

Real Time Auralization Module for Electronic Travel Aid Devices for People with Visual Disability

Alex Armendáriz, José F. Lucio-Naranjo y Diego Navas

***Resumen*—En este trabajo se presenta un módulo de software de aurilización en tiempo real que será utilizado para recrear la sensación acústica producida por un obstáculo sonoro tanto en ambientes virtuales como reales. Dicho módulo cumple la función de insertar, en una señal de audio cualquiera, el efecto de posicionamiento tridimensional que permite al oyente determinar la ubicación de una fuente de sonido dentro del ambiente de pruebas escogido. Este efecto se logra usando una técnica de procesamiento de señales llamada convolución segmentada y varias funciones contenidas en una base de datos de respuestas impulsivas asociadas a cabeza humana (HRIRs). El módulo fue probado dentro de un ambiente de pruebas real y uno virtual. En el ambiente de pruebas real el usuario llevaba consigo una cámara estereoscópica que cumplía la función de un detector de obstáculos, así como un computador y auriculares, en los cuales se instaló el módulo y se emitían las alertas sonoras tridimensionales respectivamente. De esta forma, los efectos pudieron ser registrados, analizados, discutidos y finalmente validados.**

***Palabras clave*—Aurilización, convolución segmentada, ETA, validación de realidad virtual.**

***Abstract*—This paper presents a software module for real-time auralization that was used to recreate the acoustic perception produced by a sound obstacle in virtual and real environments. This module fulfills the function of inserting, in any audio signal, a three-dimensional positioning effect that allows the listener to determine the location of a sound source within the chosen test environment. This effect was achieved with a processing technique called segmented convolution and several functions contained in a database of head related impulse responses (HRIRs). The module was tested in a real environment and a virtual one. In the real test environment, the user had a stereoscopic camera that fulfilled the function of an obstacle detector, as well as a computer and headphones, in which the module was installed and three-dimensional sound alerts were generated. In this way, the effects could be recorded, analyzed, discussed and finally validated.**

***Index Terms*—Aurilization, segmented convolution, ETA, validation of virtual reality.**

Article history:
Received 13 April 2018
Accepted 28 May 2018

Alex Armendaris fue investigador de la Pontificia Universidad Católica del Ecuador y de la Escuela Politécnica Nacional

José Lucio-Naranjo y Diego Navas son investigadores de la Escuela Politécnica Nacional

I. Introducción

La innovación tecnológica relacionada con Realidad Virtual (RV) ha permitido crear entornos virtuales cada vez más realistas. Su relevancia se refleja en el ámbito comercial, dado que en el año 2016 se registraron ventas aproximadas a mil millones de dólares en productos de realidad virtual [1]. Esto ha aumentado la necesidad de recrear los ambientes sonoros de estos entornos virtuales de maneras más precisas, para lo cual es imprescindible utilizar técnicas de aurilización.

El término aurilización (de auricular) fue introducido por Mendel Kleiner [2] y es análogo a "visualización" que quiere decir "hacer visible" un objeto proveniente de distintas fuentes reales o virtuales. En el caso de la aurilización, se trata de lograr que un efecto acústico ocasionado por un ambiente y las características del receptor sea procesado en un resultado audible [3].

La aurilización, al ser una técnica de RV, resulta sumamente útil en el análisis subjetivo acústico de espacios arquitectónicos tanto los existentes como en fase de proyecto. También tiene aplicaciones en dispositivos de apoyo a personas con capacidades especiales, vídeo juegos, entre otros. De los anterirores, los dispositivos de Navegación Asistida Electrónicamente (ETA - del inglés Electronic Travel Aid) tienen como propósito detectar obstáculos y de alguna manera comunicar al usuario de la ubicación de los mismos. Una opción en ese sentido es generar sonidos sintetizados utilizando técnicas de aurilización, que recreen la impresión de la presencia de una fuente sonora en la posición del obstáculo. Dichas señales de audio son reproducidas al usuario mediante auriculares estéreo [4].

Existen estudios previos sobre la influencia que tienen las HRTFs en sistemas de alertas sonoras en la ubicación de obstáculos por parte de personas no videntes, las cuales fueron previamente entrenadas en el uso del dispositivo [5]. A diferencia de esos estudios, el presente proyecto pretende medir cuantitativamente y cualitativamente la precisión de un sistema de alertas de sonido tridimensional con sujetos que no hayan tenido entrenamiento previo. Para dicho propósito se desarrolló un software que recrea un ambiente virtual de pruebas donde existen diversas fuentes sonoras. Por otro lado, este trabajo también contempló la realización de pruebas en un

ambiente real, para lo cual se adaptó el módulo de aurilización para que funcione como un servidor de audio que se alimenta de un sistema de detección de un obstáculos y recrea la impresión de la presencia de una fuente sonora en la posición del obstáculo.

El trabajo está ordenado de la siguiente manera: en la sección II se aborda todas las técnicas computacionales utilizadas tanto para el procesamiento de señales digitales, la generación del ambiente virtual y la interacción entre ambos. Así mismo, se revisan brevemente las técnicas para la detección de obstáculos y la implementación del servidor de audio que fue utilizado para las pruebas en ambientes reales. En la sección III se presentan los resultados obtenidos y finalmente en la sección IV se discuten los principales hallazgos de la investigación.

II. METODOLOGÍA

Para las pruebas virtuales, el proyecto requirió la implementación de un programa que consta de dos módulos. El primero está relacionado con un motor de aurilización en tiempo real que genera el audio escuchado por el usuario y el segundo permite la simulación de un ambiente tridimensional para la interacción del usuario con fuentes sonoras virtuales mediante los periféricos de un computador. Las pruebas virtuales permitieron depurar el módulo de aurilización para verificar su funcionamiento antes de que fuera integrado al sistema detector de obstáculos.

El módulo de aurilización integrado al sistema detector de obstáculos fue validado con las pruebas en ambientes reales. Dicho sistema utiliza una cámara estereoscópica y procesa la información obtenida por la cámara en un computador embebido para obtener el objeto más cercano. La integración requirió que se adapte el módulo de aurilización para ser usado como servidor de audio, el mismo que se alimenta de las coordenadas obtenidas por el sistema detector de obstáculos. Este servidor funciona de manera independiente y se comunica mediante un socket con el sistema detector de obstáculos y así se recrea por medio de auriculares la impresión de la presencia de una fuente sonora en la posición del obstáculo utilizando técnicas de procesamiento de señales digitales.

Tanto los módulos como el servidor de audio fueron programados usando C++ para reducir problemas de latencia que aparecen con lenguajes de alto nivel en su mayoría introducidos por el manejo automático de memoria de los *garbage collectors* [6], entre otros. Se utilizó Juce como *framework* para facilitar la programación de las clases relacionadas con el audio, procesamiento de señales, gráficos tridimensionales y procesamiento en paralelo.

A. Motor de Aurilización

La aurilización se basa en el principio de los sistemas acústicos lineales invariantes en el tiempo. Esto quiere decir, que la respuesta impulsiva caracteriza completamente el sistema de transmisión acústico lineal, desde una dada ubicación de la fuente sonora hasta la posición del receptor [7]. Un sistema lineal acústico invariable en el tiempo para realidad virtual está determinado por su respuesta impulsiva biauricular y sus efectos pueden ser recreados en una señal de audio arbitraria utilizando un producto de convolución [8]. Para esto, se realiza un producto de convolución entre una señal audio cualquiera con una respuesta impulsiva específica, obteniendo de esta forma el efecto de posicionamiento de la fuente sonora en el espacio. Estos principios serán aplicados tanto para las pruebas en ambientes reales como virtuales.

Basándonos este principio, se debe utilizar una base de datos con pares de HRIRs (del inglés, Head Related Impulse Responses), cada uno de estos pares caracteriza la forma como un sonido llega de un punto del espacio a los oídos de una persona [9]. En ese sentido, cada par de HRIRs tendrá respuestas impulsivas correspondientes a dos sistemas lineales invariantes en el tiempo, que a su vez corresponden a dos receptores (oído izquierdo y derecho) y una fuente sonora ubicada en una posición específica. En este caso, se utilizó la base de HRIRs del MIT, levantada experimentalmente con la cabeza artificial KEMAR dentro de una sala anecoica [9].

Lo primero que se debe tomar en cuenta es que el procesamiento de la señal debe funcionar en tiempo real (con una latencia mínima). Por tal motivo, para este caso se manejó bloques de audio de 512 muestras y una tasa de muestreo de 44100 (el tamaño del bloque puede tener otro tamaño dependiendo de las capacidades de la tarjeta de sonido). Para obtener la latencia se debe dividir el tamaño del bloque entre la tasa de muestreo obteniendo así una latencia aproximada de $11{,}61$ milisegundos [6].

Las frecuencias bajas no son direccionales, es decir no sufren de modificaciones espectrales significativa si se las capta de otra dirección [10]. Por tal motivo, no es necesario añadir el efecto de posicionamiento en ellas [11]. Por lo que es necesario dividir la señal que se va a procesar en dos partes: una que contenga sus componentes de baja frecuencia de 0 Hz a 200 Hz y otra los componentes de alta frecuencia de 200 Hz a 22.05 KHz. Para lograr la separación debemos utilizar dos filtros digitales IIR, un pasa bajos y un pasa altos. A continuación, se procesa únicamente las frecuencias altas y a este resultado se le suma la señal de baja frecuencia.

Al utilizar un producto de convolución, expresado en su forma convencional en la Ec. 1, se debe procesar la señal integralmente. Esto requiere un gran un número de operaciones lo que introduce retardos en el procesamiento [12].

$$x[k] * h[k] = \sum_{n=0}^{N-1} x[n]h[k-n] \tag{1}$$

Estos retardos atentan contra la generación de audio en tiempo real, por lo que es necesario usar un algoritmo de convolución segmentada lo que permite procesar la señal por bloques [12]. El procesamiento por bloques basa su funcionamiento en la Ec. 2.

$$x_L[k] = \sum_{p=0}^{L/P} x_p[k - p \cdot P] \tag{2}$$

Este algoritmo, conocido como "sumas de superposiciones" [12], separa la señal $X_L[k]$ (de tamaño L) en segmentos

de $X_p[k]$ (de tamaño P) que están definidos como:

$$x_p[k] \begin{cases} x_L[k - p \cdot P] & para \;\; k = 0, 1, ..., P-1 \\ 0 & otros \end{cases} \quad (3)$$

La longitud P del bloque depende del tamaño del *buffer* de salida de audio. No obstante, el tamaño de cada elemento procesado será de $P + N - 1$ (donde N es el tamaño de la respuesta impulsiva $h_N[k]$). Este problema se supera sumando las muestras que sobrepasen la longitud P al siguiente bloque, tal como ilustra la Fig. **1**.

Por lo anterior, el resultado de una convolución $X_L[k] * h_N[k]$ se puede expresar como la suma de superposiciones de una serie de convoluciones $X_p[k] * h_N[k]$, cada una desplazada en múltiplos de P [12].

Utilizando el teorema de la convolución [13], descrito en la Eq. 4, es posible realizar una optimización final al algoritmo.

$$x[k] * h[k] = X[w]H[w] \quad (4)$$

De esta forma, en lugar de realizar múltiples productos y sumas en el dominio del tiempo con la Ec. 1, la señal monofónica de entrada es transformada al dominio de la frecuencia usando la transformada rápida de Fourier (FFT). Luego la señal es multiplicada por el par de HRTFs obteniendo una señal estéreo, y el producto resultante es transformado nuevamente al dominio de tiempo utilizando la transformada rápida de Fourier inversa (IFFT).

Dado que se está lidiando con principios aplicables a sistemas lineales invariantes en el tiempo, el efecto de realidad virtual se alcanza actualizando las respuestas impulsivas (HRIRs) dependiendo de la orientación y la posición de la cabeza en un dado instante de tiempo. No obstante, este cambio puede ser brusco debido al movimiento arbitrario de la cabeza en el ambiente de pruebas, lo cual puede generar clics u otros efectos indeseados [14]. Por tal motivo, es necesario contar con dos módulos de convolución, uno de destino y uno actual. El módulo de destino contendrá el par de HRIRs que se va a utilizar a continuación y el actual contendrá el par de HRIRs que utiliza actualmente el usuario. La solución se obtiene aplicando un efecto *crossfade* entre los módulos utilizando interpolaciones lineales muestra a muestra [11].

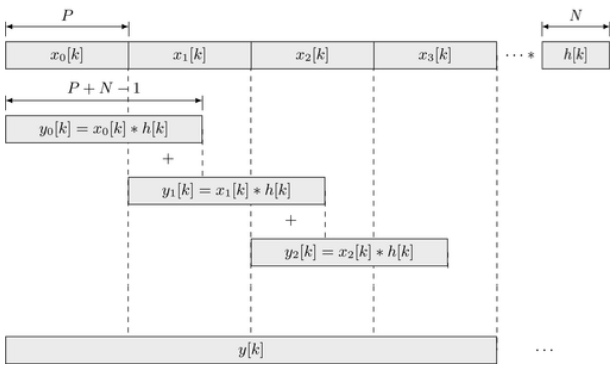

Figura 1. Convolución Segmentada [12]

B. *Ambiente 3D y Sistema de Interacción Físico*

En el ambiente virtual tridimensional se puede encontrar obstáculos que se pueden visualizar como mallas compuestas de triángulos (ver Fig.**2**).

El receptor es representado por una esfera que tiene una unidad espacial virtual de radio. Por otro lado, los emisores fueron representados como puntos en el espacio. El sistema, mediante el mouse y teclado, permite navegar dentro del ambiente moviendo una cámara virtual con perspectiva en primera persona. Para la creación de todos los gráficos computacionales se utilizó OpenGL.

La determinación de la dirección relativa de la fuente sonora con respecto al receptor biauricular determina un rayo que parte desde el centro del receptor hacia el centro del emisor. En seguida se analiza si existe colisión entre el rayo y algún triangulo componente de algún obstáculo de la escena utilizando el algoritmo de Möller–Trumbore [15].

Si existe colisión del rayo con algún triángulo, y su distancia de colisión es menor que la distancia del emisor al receptor, se verifica que existe oclusión por lo que no se podría escuchar la fuente. En ese caso, se desactiva el generador de audio 3D, caso contrario se activa el generador de audio 3D, tal como lo indica la Fig. **3**.

Al no existir oclusión, se transforma la dirección de llegada del rayo al sistema de coordenadas del receptor que es dependiente del movimiento de traslación y rotación provocado por el usuario usando el mouse y teclado. Para esto se utiliza matrices de rotación (Ecs. 5 y 6) alrededor de los ejes x y y [16].

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sen(\theta) \\ 0 & sen(\theta) & cos(\theta) \end{bmatrix} \quad (5)$$

$$R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sen(\theta) \\ 0 & 1 & 0 \\ -sen(\theta) & sen(\theta) & cos(\theta) \end{bmatrix} \quad (6)$$

Posteriormente, se transforma las coordenadas de arribo del rayo del sistema cartesiano a polar para que estas sean compatibles con la base de datos de HRIRs de Kemar [9]. Con estas coordenadas de arribo, se realiza la búsqueda del

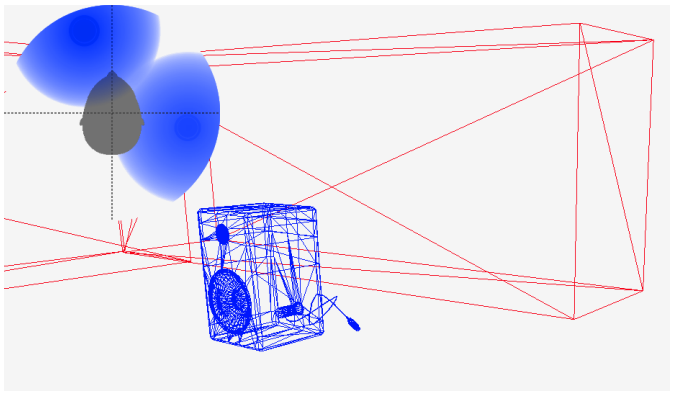

Figura 2. Interface Gráfica

par de HRIRs en la base de datos obteniendo el par de HRIRs más próximo a la dirección de incidencia del rayo sobre el receptor.

Para terminar el procesamiento se calcula un factor de ganancia, la cual se usará para recrear el efecto de reducción de intensidad del sonido debido a la distancia. Esta ganancia es inversamente proporcional al área de una esfera, que tiene por radio el módulo del vector existente entre la fuente y el receptor. Este factor de ganancia es compensado por otro factor (obtenido de manera experimental) para mejorar el realismo entre la distancia que se observa en el simulador y la que se escucha. Este procedimiento se efectúa cada vez que un nuevo cuadro de vídeo es generado en la escena, como se ilustra en el Alg. 1.

while *no finalice el programa* **do**
  Determinar un rayo desde el emisor hasta el centro del receptor;
  if *no existen colisiones* **or** *módulo del rayo* $<$ *que la distancia emisor-triángulo intersecado* **then**
    Trasladar rayo al sistema de referencia del receptor;
    Transformar a coordenadas polares;
    Buscar HRIR en la base de datos;
    Calcular ganancia dependiendo de la distancia del emisor al receptor;
  else if *Existe oclusión* **then**
    Bloquear sonido de la fuente;
end

Algorithm 1: Algoritmo de interacción física fuente-receptor

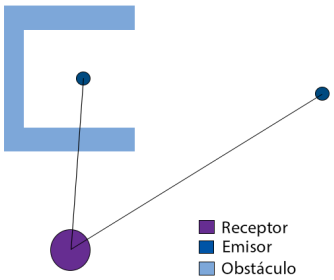

Figura 3. Sistema de Interacción Físico

C. *Procesamiento paralelo ambiente virtual*

Para el funcionamiento en tiempo real (con una latencia mínima), la aplicación que recrea el ambiente de manera virtual requiere que el audio y gráficos se ejecuten en diferentes hilos de ejecución de forma paralela. En el caso del audio, se ejecuta un hilo de alta prioridad, mientras que para los gráficos se utiliza un hilo de prioridad normal. En la Fig. **4** se muestra la arquitectura de los hilos sincronizados mediante mecanismos libres de bloqueos.

Un bloqueo es cualquier procedimiento que requiera esperar por un recurso (esperas causadas por el procesamiento de otros hilos o lectura de datos del disco duro, presencia de mutex o socket, entre otras). Cuando el audio es procesado por bloques (tiempo real), esperar por un recurso causa la pérdida de muestras del buffer de audio, lo cual se traduce en breves interrupciones causando ruido en la señal de sonido lo cual no es adecuado para una ejecución en tiempo real [6].

Para realizar una sincronización libre de bloqueos es conveniente el uso de variables atómicas para garantizar el acceso seguro a las mismas, evitando de esta forma que existan "data races". Para este caso se utilizaron como variables atómicas (de la librería standard de C++ [17]) al azimuth ϕ y la elevación θ, las cuales se obtienen del sistema de interacción 3D.

D. *Arquitectura servidor de audio*

La arquitectura escogida para el programa para las pruebas en el ambiente real fue una arquitectura cliente servidor, debido a que nos permite que el módulo de aurilización sea independiente del sistema detector de obstáculos tanto en el desarrollo del mismo como en su funcionamiento.

Un cliente es un sistema o programa que realiza peticiones de una o varias actividades a otro programa o sistemas, llamados servidores, que cumplen tareas específicas [18]. En este caso el cliente es el sistema detector de obstáculos que envía las coordenadas y distancia del obstáculo al servidor de audio para que el mismo procese una señal de audio y recreen la impresión de la presencia de una fuente sonora en la posición del obstáculo que es audible mediante auriculares.

Para la comunicación se utilizan sockets TCP que se comunican mediante un servidor local con IP 127.0.0.1, lo que

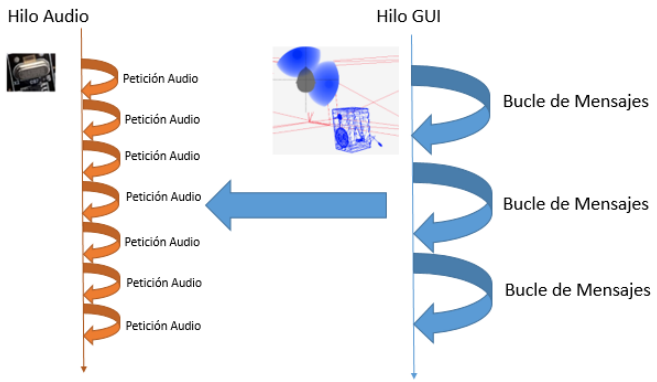

Figura 4. Arquitectura de Hilos de una Típica Aplicación de Audio

permite tener una latencia aproximada menor a 1 milisegundo en la comunicación en un computador Jetson TK1. El cliente cada vez que procesa una nueva posición en coordenadas polares envía la misma al servidor de audio para que genere un resultado audible. Al igual que en la aplicación para pruebas en un ambiente virtual es necesario que el socket de comunicación se ejecute en un hilo separado. Lo que genera la necesidad de sincronización con el hilo de procesamiento de audio, para esto se utilizaron variables atómicas. Para esta comunicación fue necesario definir un protocolo de intercambio de 13 bytes, cada byte es de tipo caracter de C++. Los tres primeros bytes representan la posición en azimuth del obstáculo que puede variar en 90 y -90 grados, seguidos por un byte de separación que es representado por una coma. A continuación, los siguientes tres bytes representan la posición en elevación del obstáculo que puede variar en -90 y 90 grados, se agrega una coma de separación y finalmente los últimos 5 bytes representan la distancia al obstáculo. Si no existe un obstáculo la distancia toma un valor negativo y es ignorada por el servidor de audio. Debido a que se utiliza la base de HRIR de KEMAR [9] si el azimuth fuera menor a -40 grados, se utilizará la HRIR con posición más baja en altura de la base de datos (-40 grados en elevación).

E. Sistema de Detección de Obstáculos

La estereoscopia, también denominada visión estero, permite el análisis y procesamiento de espacios tridimensionales en base a técnicas y mecanismos de adquisición de información bidimensional (imágenes) con el uso de dos o más cámaras. De esta forma este método simula la capacidad visual de los seres vivos al analizar las diferencias entre dos imágenes adquiridas. En el presente trabajo se hace uso de uno de los métodos para realizar reconstrucción tridimensional, el mismo que se basa en el uso de una cámara estereoscópica comercial llamada ZED que presenta una disposición alineada de sus dos cámaras [19].

El proceso de reconstrucción tridimensional a partir de visión binocular (estereoscopia con dos cámaras) comienza con la adquisición de las imágenes al mismo tiempo [19]). Una vez obtenidas las imágenes, se procede al análisis de correspondencias que se define como el proceso de encontrar las coordenadas de un píxel en dos diferentes imágenes que corresponden al mismo punto en el mundo real [20]. Los resultados de este proceso son distancias entre píxeles denominadas disparidades. A continuación, se realiza un proceso de triangulación de dichas disparidades para hallar una dimensión adicional. Con toda esta información se selecciona una de las imágenes obtenidas al inicio y cada píxel es empujado en una nueva dimensión para obtener una representación 3D del entorno. Todos estos procesos son realizados gracias a la calibración constante de las cámaras para determinar parámetros indispensables en las trasformaciones realizadas.

Para la detección de obstáculos la información 3D obtenida de la cámara estereoscópica es procesada mediante una librería conocida como Point Cloud Library (PCL) [21]. Se realiza una reducción de información para mejorar el desempeño del sistema a través procesamientos y filtros [22] como: Region of Interest (ROI), Voxel y Statistc Outliner Removal (SOR) [23]. Para la eliminación de puntos correspondientes al suelo se procesa el entorno reconstruido mediante el algoritmo RANSAC que permite la segmentación y eliminación de datos con características seleccionadas [24].

Una vez obtenido los obstáculos del entorno y los datos del suelo, se encuentra un conjunto de puntos más cercano, se eliminan datos atípicos (SOR) y se determina el centroide del mismo que es considerado la ubicación del obstáculo más cercano. Para el suavizado de la respuesta se utiliza un filtro de media móvil.

III. RESULTADOS

A. Prueba en Ambiente Virtual

Para probar la eficiencia de procesamiento y precisión de los resultados que produce esta aplicación en ambientes virtuales se desarrollaron 2 tipos de pruebas:

1. Prueba de navegación ciega.
2. Prueba de valoración subjetiva.

Para la primera prueba participaron dos sujetos sin entrenamiento previo. Esta validación consistía en realizar una navegación dentro del ambiente virtual por medio de los periféricos, sin tener una realimentación visual. Inicialmente, el sujeto era colocado en un punto fijo del ambiente virtual a una elevación de 1 unidad [1] y la fuente de sonido era ubicada alrededor del sujeto en una posición aleatoria con un azimut variable entre $0°$ y $360°$ a la altura del piso a 26 unidades de distancia del sujeto. El objetivo era llegar lo más cerca posible de la fuente sonora, apenas guiándose por el sonido que provenía de auriculares ecualizados. Una vez que el sujeto creía alcanzar su objetivo, este debía presionar una tecla para finalizar el procedimiento, el cual se repitió 50 veces.

Los resultados indican que se logró reducir la distancia inicial de separación en un $93{,}96\,\%$ en promedio. Esto a pesar de que las pruebas fueron realizadas con las HRIRs de la cabeza artificial KEMAR [9] y no las específicas de cada sujeto de prueba, lo cual puede generar distorsiones en la percepción [25]. Adicionalmente, la base de datos de HRIRs no cuentan con datos para elevaciones menores a $-40°$, lo cual genera mayores imprecisiones cuando el sujeto está cerca de la fuente.

Los resultados de las distancias finales que separan a la fuente sonora del sujeto se presentan en las Tabs. **1** y **2** para el sujeto 1 y 2 respectivamente. Se utilizan 6 intervalos para sintetizar la información. La segunda columna define el rango de distancia de los intervalos. Para cada intervalo se presenta la frecuencia de ocurrencia, fi (tercera columna), y la frecuencia acumulada, Fi (cuarta columna). Además, en la quinta columna se presenta el rango de error relativo obtenido.

Los datos muestran que el $62\,\%$ de los intentos el error no sobrepasa el $8\,\%$ de la distancia original. Apenas el $20\,\%$ de los intentos sobrepasa el $10\,\%$, sin embargo, ninguno de estos supera el $15\,\%$. El error relativo promedio fue de $6{,}04\,\%$. Cabe recalcar que los gráficos de frecuencia de ocurrencia de los intervalos, registran que existe una tendencia hacia abajo de los intervalos de peor desempeño (ver Figs. **5** y **6**).

[1] Unidad de distancia dentro del ambiente virtual.

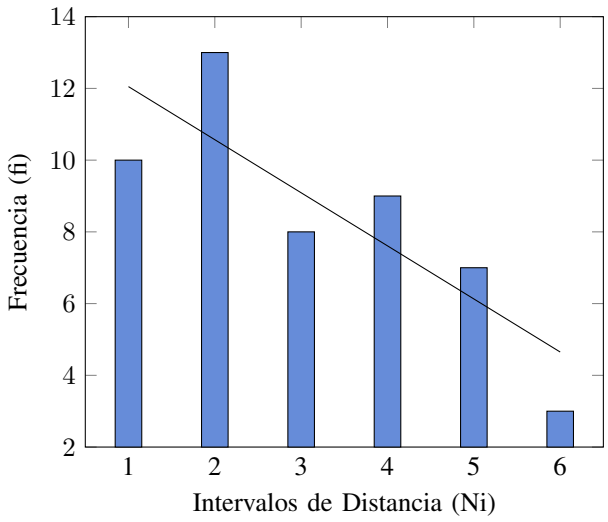

Figura 5. Resultados Sujeto 1

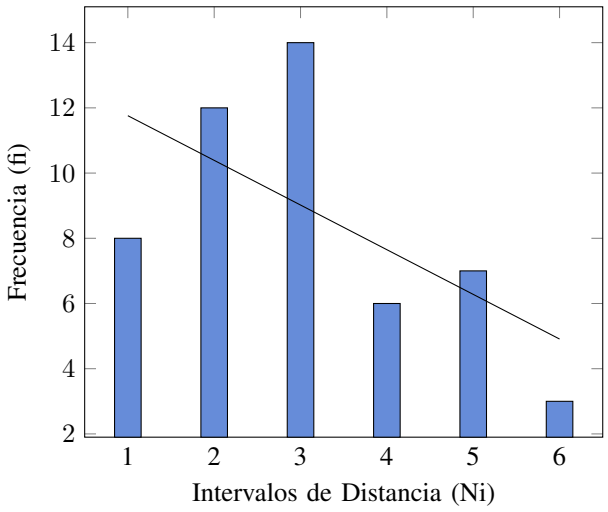

Figura 6. Resultados Sujeto 2

La segunda prueba consistió en una valoración subjetiva realizada a 70 sujetos durante la presentación de la aplicación durante el evento "Encuentros ciencia y tecnología EPN 2016 Ciudades Sostenibles en el Siglo XXI" en el marco del Hábitat III. En esta validación se pidió a los sujetos provistos de audífonos a que naveguen libremente en un ambiente virtual tridimensional donde se encontraban varias fuentes sonoras. Después, se les solicitó que llenen una encuesta cualitativa de dos preguntas de opción múltiple sobre la fidedignidad del sonido 3D generado por la aplicación.

La primera pregunta (ver Fig. **7**) estaba relacionada a la impresión del sujeto sobre si se consiguió generar correctamente sonidos tridimensionales. El 90 % respondió validando el efecto de la aplicación.

La segunda pregunta (ver Fig. **8**) buscaba información sobre cuánto costó percibir el efecto de inmersión acústica 3D en términos de manipulación de la aplicación. El 53 % de los encuestados reportó haber percibido el efecto de forma inmediata. Al siguiente 36 % le costó un poco acostumbrarse, mientras que el 11 % restante reportó haber tenido dificultades significativas.

B. Prueba en Ambiente Real

El equipo utilizado para las pruebas fue un computador Jetson TK1 con un procesador NVIDIA 4-Plus- Quad-Core ARM, una tarjeta de video NVIDIA Kepler con 192 núcleos CUDA y Ubuntu 14.04 como sistema operativo. Se utilizó un tamaño de bloque de 4096 muestras (el tamaño más pequeño sin que produzca ruido) y una taza de muestreo de 44100 muestras por segundo para el procesamiento del audio. Lo que nos da una latencia aproxima de 90 milisegundos en el audio que se obtiene dividiendo el tamaño del bloque para la taza de muestreo. Para la adquisición se utilizó una cámara estereoscópica Zed conectada a la Jetson TK1 que alimenta

Tabla 1
RESULTADO DISTANCIA SUJETO 1

Ni	Distancia	fi	Fi	Error
1	0,0998 a 0,8072	10	10	0 % al 3 %
2	0,8072 a 1,4146	13	23	3 % al 5 %
3	1,4146 a 2,0220	8	31	5 % al 8 %
4	2,0220 a 2,6294	9	40	8 % al 10 %
5	2,6294 a 3,2368	7	47	10 % al 12 %
6	3,2368 a 3,8441	3	50	12 % al 15 %

Tabla 2
RESULTADO DE DISTANCIA SUJETO 2

Ni	Distancia	fi	Fi	Error
1	0,0501 a 0,6194	8	8	0 % al 2 %
2	0,6194 a 1,1887	12	20	2 % al 5 %
3	1,1887 a 1,7580	14	34	5 % al 7 %
4	1,7580 a 2,3273	6	40	7 % al 9 %
5	2,3273 a 2,8966	7	47	9 % al 11 %
6	2,8966 a 3,4659	3	50	11 % al 13 %

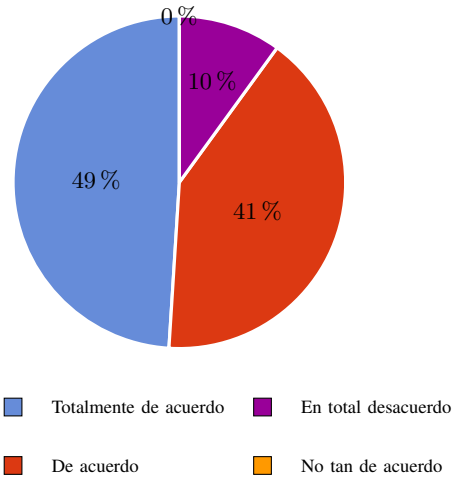

Figura 7. ¿Considera que el software simula correctamente sonido tridimensional?

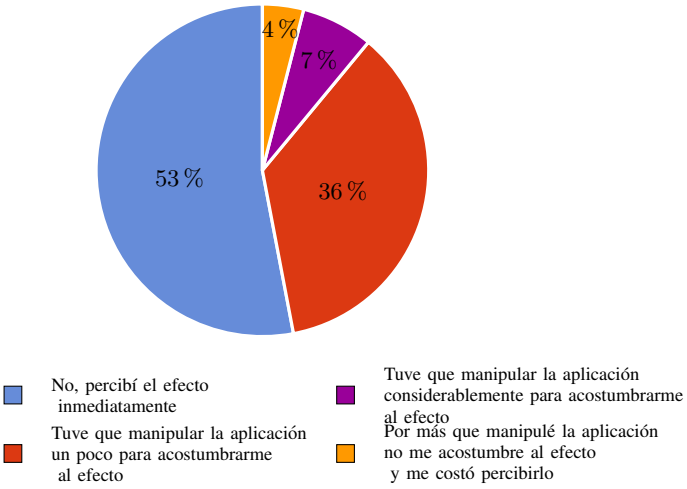

Figura 8. ¿Tuvo necesidad de acostumbrarse al sonido para percibir la sensacion 3D?

Figura 9. Sujeto 3 realizando la prueba

el sistema de detección de obstáculos. Dicho sistema provee de información de posición al servidor de audio cada 0.35 segundos en promedio (3 cuadros por segundo).

Las pruebas se realizaron en un espacio cerrado Fig. **9** con 6.60 m de largo por 2.60 m de ancho. Se dividió en espacios de 2.2 m de largo por 0.87 cm de ancho. Donde se ubicaban obstáculos de tamaño 1.76 m de altura y 60 cm de ancho, de manera aleatoria para cada una de sus 6 combinaciones posibles sin que un obstáculo se encuentre en la misma columna, ver Fig. **10**. Alrededor existía una barrera de 73 cm de alto y al final de la sala se encontraba una fuente de sonido con música a la que el sujeto debía llegar completamente a ciegas esquivando los obstáculos.

Se realizó una prueba preliminar sin una fuente sonora guía en la posición de llegada. Aquí se pudo notar que el sujeto se desorientó al no tener noción de su posición ni del lugar de llegada. Para las siguientes pruebas fue necesario añadir una fuente de sonido guía al final de la sala para que el sujeto tenga una referencia a la posición de llegada.

Los sujetos de prueba tenían una capacidad visual normal,

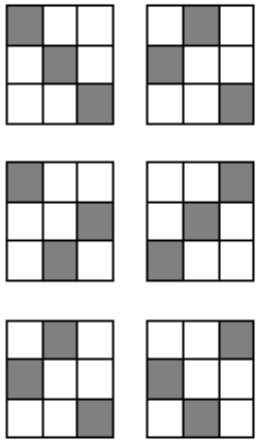

Figura 10. Posiciones aleatorias utilizadas

cada sujeto se entrenó 5 minutos para aprender la relación del volumen que percibía con la distancia de la fuente antes de realizar la prueba.

Todos los sujetos fueron capaces de percibir las señales de alerta producidos por el módulo de aurilización y el sistema de detección de obstáculos, aunque algunos lo percibían de mejor manera que otros. Esto se debe en parte a las diferencias anatómicas que causan distorsiones en la percepción, al poco entrenamiento y la falta de familiaridad con el dispositivo.

En promedio existe menos de una colisión con los obstáculos por prueba (0.53 col por prueba). Igualmente existe en promedio menos de una colisión con las barreras por intento, pero existían choques con la misma barrera en el mismo lugar debido a que la cámara estereoscópica falla con obstáculos a estas alturas.

Los resultados en promedio de los seis sujetos se muestran en la Tab. **3**. El tiempo promedio del recorrido fue 1.31 minutos con 0.53 colisiones (una cada dos intentos) con los obstáculos en promedio y 0.33 colisiones (una cada tres intentos) en promedio con las barreras. Como se puede observar existe una tendencia a que el tiempo disminuya Fig. **11**, pero con esto se incrementa la cantidad de colisiones sobre todo con los obstáculos ver Fig. **12** y Fig. **13**. Este incremento se debe a la latencia que posee la cámara que da percepciones erróneas sobre el posicionamiento de los obstáculos lo que causa interpretaciones erróneas del usuario, por lo tanto, colisiones con los obstáculos.

Las tres pruebas que no se finalizaron se debe a que el

Tabla 3
RESULTADOS DISTANCIA SUJETO 1

N.	Tiempo (min)	C. Barrera	C. Obstáculo	N. P. Terminadas
1	1,57	0,40	0,40	5
2	1,08	0,00	0,50	4
3	1,01	0,20	0,20	6
4	1,36	0,50	0,50	6
5	0,84	0,17	0,67	6
6	1,05	0,17	0,67	6

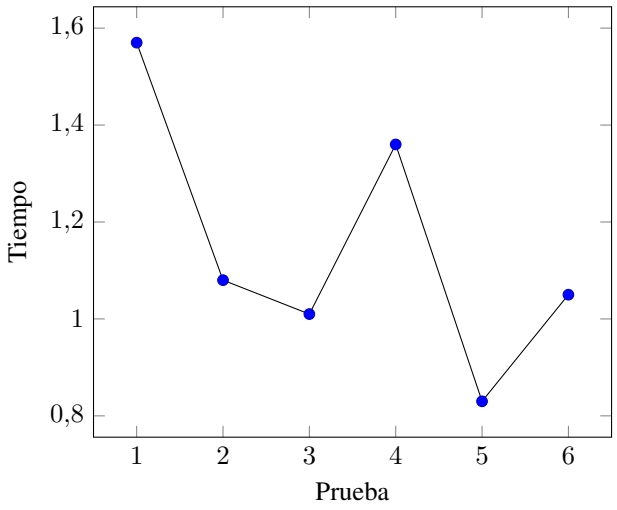

Figura 11. Tiempo promedio por prueba

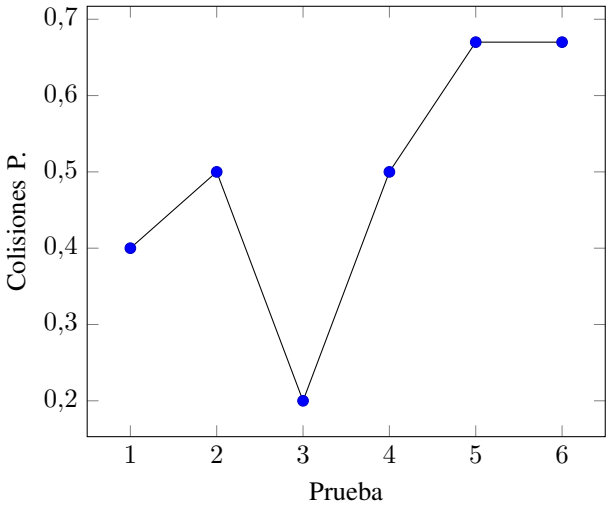

Figura 13. Colisiones promedio con obstáculos por prueba

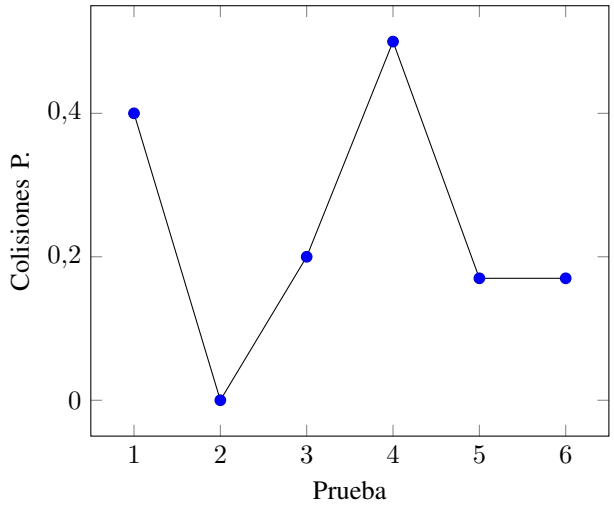

Figura 12. Colisiones promedio con barreras por prueba

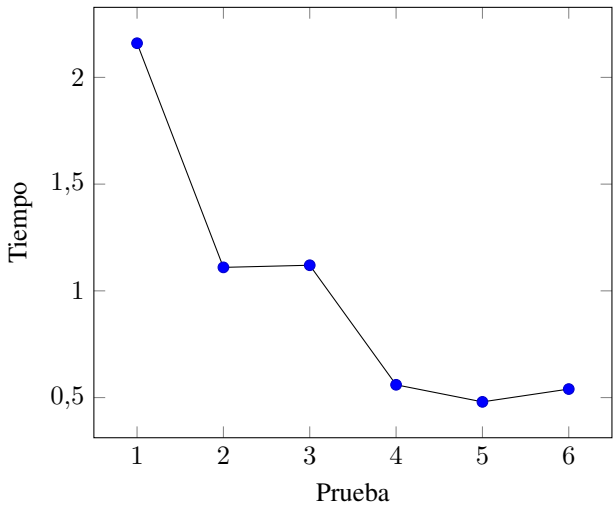

Figura 14. Tiempo sujeto 3

sujeto quedó atrapado entre la barrera y a una distancia menor a 50 cm por lo que escuchaba alertas sonoras en muchas direcciones. Esto se debe a que la cámara empieza a fallar en distancias menores a 50 cm y con obstáculos que se encuentren por debajo de la cintura del sujeto.

Si observamos las estadísticas del sujeto de prueba con mejor desempeño (Tab. **3**), podemos observar como disminuye el número de colisiones y a la vez reduce el tiempo. Como se muestra en la tabla 2 este sujeto redujo el número de colisiones y además el tiempo con cada repetición (ver Fig **14** y **15**). El sujeto finalizó las pruebas con un promedio de 59.7 segundos de tiempo y 0.33 colisiones por intento.

IV. Conclusiones y Recomendaciones

En este trabajo se consiguió desarrollar un módulo de aurilización para alertas sonoras de obstáculos y que funciona en tiempo real dentro de un ambiente, pudiendo ser este real o virtual.

Para el caso de las pruebas en el ambiente simulado, fue necesario configurar los comandos de traslación y rotación sobre su propio eje usando periféricos como el mouse y el teclado. Con esta herramienta de software fue posible realizar dos pruebas, una cuantitativa y otra cualitativa para poder validar el correcto funcionamiento del motor de aurilización. Los resultados cuantitativos demuestran que, a pesar de utilizar las HRIRs de la cabeza artificial KEMAR, los sujetos son capaces de reducir la distancia de separación con la fuente sonora en un 93,96 %. El análisis de frecuencia de los casos clasificados en intervalos de precisión muestra también una

Tabla 4
Resultados Distancia Sujeto 3

N.	Tiempo (min)	C. Barrera	C. Obstáculo	P. Terminada
1	2,16	0	1	si
2	1,11	0	0	si
3	1,12	0	1	si
4	0,56	0	0	si
5	0,48	0	0	si
6	0,54	0	0	si

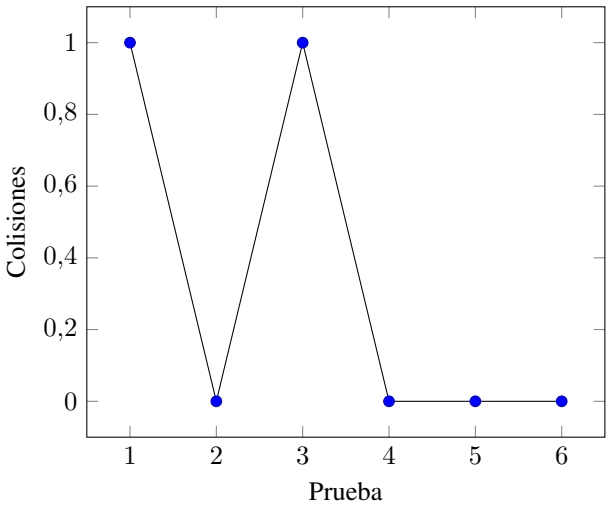

Figura 15. Colisiones sujeto 3

tendencia a la baja de los casos de más bajo desempeño. Esto indica que a medida que se repita el proceso, se produce un proceso de adaptación del sujeto a los sonidos 3D generados por la aplicación.

Por otro lado, los resultados de las pruebas cualitativas validan positivamente la generación del efecto sonoro 3D en el $90\,\%$ de los casos y que el mismo fue percibido de forma prácticamente intuitiva. En el caso del $10\,\%$ restante, se estima que existe un problema relacionado con falta de familiaridad con la interfaz de la aplicación. Es muy probable que esto también esté relacionado a impresiones producidas por no usar las HRIRs propias del sujeto. Este problema puede resolverse con un proceso de entrenamiento, tanto en el uso de la aplicación como en la utilización de HRIRs ajenas al sujeto.

Para el caso de las pruebas en el ambiente real, el sujeto con la mejor relación entre choques y tiempo tiene un promedio de 0.33 choques por prueba y tardó en promedio 59.7 segundos por prueba. Lo que demuestra que el sistema permite detectar obstáculos, aunque los sujetos hayan tenido un entrenamiento corto.

La latencia del sistema dificultó a los sujetos percibir los obstáculos de manera correcta, ya que debían permanecer sin movimiento hasta que el punto captado por el sensor se estabilice. Además, debido a que los sensores están a la altura del pecho, los obstáculos que se encuentran a la altura de la cintura como las barreras laterales son más difíciles de detectar. Por lo que se deben buscar mejores sensores o complementar los existen para reducir los fallos.

Finalmente debido al aumento de velocidad de los sujetos (ya que se sentían más confiados) la latencia del dispositivo causó más colisiones, por lo que es recomendable utilizar computadores que soporten una mayor cantidad de cómputo.

En conclusión, el motor de aurilización fue probado exitosamente (tanto en el ambiente virtual como real) con las pruebas descritas en este trabajo. Por tal motivo, este prototipo de software tiene el potencial de ser aplicado para la generación de sonidos 3D en dispositivos ETA comerciales, de manera que alerten de obstáculos a personas con discapacidad visual. Se prevé que sea necesario reducir aún más la complejidad computacional del motor de aurilización para poder procesar varias fuentes al mismo tiempo (para lidiar con varios obstáculos) o para poder simplificar la arquitectura del dispositivo. Otra alternativa sería reemplazar los actuales procedimientos de detección de obstáculos con técnicas alternativas de machine learning para acelerar su procesamiento.

AGRADECIMIENTOS

Los autores agradecen el apoyo financiero de la Escuela Politécnica Nacional para el desarrollo de este proyecto de investigación y específicamente de la Pontificia Universidad Católica del Ecuador en lo referente al motor de aurilización en ambientes virtuales.

BIBLIOGRAFÍA

[1] P. Lee and D. Stewart, "Virtual reality (vr): a billion dollar niche tmt predictions 2016." [Online]. Available: https://www2.deloitte.com/global/en/pages/technology-media-and-telecommunications/articles/tmt-pred16-media-virtual-reality-billion-dollar-niche.html

[2] K. Mendel, B.-I. Dalenbäck, and P. Svensson, "Auralization – an overview," *J. Audio Eng. Soc. 41*, p. 861, 1993.

[3] M. Vorländer, *Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Springer, 2008.

[4] J. F. Lucio Naranjo, R. Tenenbaum, L. A. Paz Arias, H. P. Morales Escobar, and I. J. Iniguez Jarrín, "3d sound applied to the design of assisted navigation devices for the visually impaired," *Latin-American Journal of Computing*, 2015.

[5] I. Lengua, D. Larisa, G. Peris, and B. Defez, "Dispositivo de navegación para personas invidentes basado en la tecnología time of flight." [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532013000300004

[6] R. Bencina, "Real-time audio programming 101: time waits for nothing." [Online]. Available: http://www.rossbencina.com/code/real-time-audio-programming-101-time-waits-for-nothing

[7] G. Alonso, L. Budde, and M. Zannier, "Síntesis de respuesta impulsiva de recintos a través del método de trazado de rayos," *UTN FRC - Depto Ing. Electrónica*, 2012.

[8] F. Pishdadian, "Filters, Reverberation & Convolution," 2017. [Online]. Available: http://www.cs.northwestern.edu/~pardo/courses/eecs352/lectures/MPM16-topic9-Filtering.pdf

[9] B. Gartner and K. Martin, "HRTF Measurements of a KEMAR Dummy-Head Microphone." [Online]. Available: http://sound.media.mit.edu/resources/KEMAR.html

[10] Ref. **?**, p. 28.

[11] T. Woźniak, "Implementing Binaural (HRTF) Panner Node with Web Audio API," April 2015. [Online]. Available: https://codeandsound.wordpress.com/2015/04/08/implementing-binaural-hrtf-panner-node-with-web-audio-api/

[12] S. Spors, "Segmented Convolution—DigitalSignalProcessing 0.0 documentation." [Online]. Available: http://dsp-nbsphinx.readthedocs.io/en/nbsphinx-experiment/nonrecursive_filters/segmented_convolution.html

[13] S. Smith, "FFT Convolution." [Online]. Available: http://www.dspguide.com/ch18/2.htm

[14] G. Wersnyi, "Effect of emulated head-tracking for reducing localization errors in virtual audio simulation," *IEEE Transactions On Audio, Speech, And Language Processing*, vol. 17, no. 2, pp. 247–252, 2009.

[15] T. Möller and B. Trumbore, "Fast, Minimum Storage Ray/Triangle Intersection," in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. New York, NY, USA: ACM, 2005. [Online]. Available: http://doi.acm.org/10.1145/1198555.1198746

[16] Ref. **?**, p. 98.

[17] T. C. R. Network, "Cplusplus reference ¡atomic¿." [Online]. Available: http://www.cplusplus.com/reference/atomic/

[18] G. Held, *Server Management*. CRC Press, 2000.

[19] F. Torres, P. Pomares, J.and Gil, and S. Puente, *Robots y Sistemas Sensoriales*. Prentice Hall, 2002.

[20] P. Corke, *Robotics, Vision and control*. Springer, 2013.

[21] T. C. R. Network, "About - point cloud library (pcl)." [Online]. Available: http://pointclouds.org/about/
[22] B. Li, X. Zhang, Munoz, X. J. P., X. J., Rong, and Y. Tian, "Assisting blind people to avoid obstacles: An wearable obstacle stereo feedback system based on 3d detection," *IEEE International Conference on Robotics and Biomimetics*, 2016.
[23] A. Garcia, "Towards a real-time 3d object recognition pipeline on mobile gpgpu computing platforms using low-cost rgb-d sensors," ser. CEUR Workshop Proceedings, 2015. [Online]. Available: https://doi.org/10.1017/CBO9781107415324.004
[24] A. Nguyen and B. Le, "3d point cloud segmentation: A survey," ser. IEEE Conference on Robotics, Automation and Mechatronics, 2013. [Online]. Available: https://doi.org/10.1109/RAM.2013.6758588
[25] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The cipic hrtf database." [Online]. Available: https://pdfs.semanticscholar.org/cee9/f63da2cafe7dd7b8bd0752bea57f38d4afc5.pdf

V. AUTORES

Alex Armendáriz Ingeniero en Sistemas y Computación por la Ingeniería en Sistemas de la Pontificia Universidad Católica del Ecuador (2017). Actualmente estudiante distancia de Point Blank London en la carrera de Ing. en sonido y producción musical, participó como ayudante de investigación en el área de simulación acústica en la Escuela Politécnica Nacional y la Pontificia Universidad Católica del Ecuador. Su principal campo de estudio está enfocado en aplicaciones de audio en tiempo real. El trabajo que actualmente realiza se encuentra enfocado en la implementación de sistemas de aurilización utilizando técnicas de procesamiento digital de señales, computación gráfica e inteligencia artificial.

José F. Lucio-Naranjo Ingeniero en Sistemas y Computación por la Pontificia Universidad Católica del Ecuador (2005). Máster y Ph.D. en Modelado Computacional por la Universidad del Estado de Río de Janeiro (2010 y 2014 respectivamente). Su investigación doctoral fue reconocida y apoyada por la Acoustical Society of América mediante ASA International Student Grant. Su principal campo de estudio está enfocado en técnicas de simulación numérica e inteligencia computacional aplicadas al modelado de la propagación acústica y a la generación de realidad virtual. Actúa como investigador y profesor titular de sistemas y computación en la Escuela Politécnica Nacional del Ecuador (EPN) y también como profesor e investigador a tiempo parcial en la Pontificia Universidad Católica del Ecuador (PUCE). También ha actuado como profesor en la Universidad del Estado de Río de Janeiro (UERJ), la Universidad Federal Fluminense, la Universidad de las Américas (UDLA) y la Universidad Central del Ecuador (UCE). Fue miembro de la Sociedad Brasileña de Acústica (SOBRAC).

Diego Francisco Navas Flores Nació en 1993 en Atuntaqui-Ecuador. Estudió su secundaria en la Unidad Educativa "La Salle". Se graduó como ingeniero en Electrónica y Control en la Escuela Politécnica Nacional en el 2018. Fue miembro de Club de Robótica de la EPN desde el año 2014. Trabaja como ayudante de investigación en la Escuela Politécnica Nacional.