

DESARROLLO DE UNA HERRAMIENTA MATLAB PARA LA REPRODUCCIÓN DE AUDIO ESPACIAL EN TIEMPO REAL

PACS: 43.60.-c

Maximo Cobos¹, Gabriel Moreno¹, Jesus Lopez-Ballester¹, Pablo Gutierrez-Parera², Irene Martin-Morato¹, Jaume Segura¹ y Ana M. Torres³

¹ SPAT - Universitat de Valencia, Avd. de la Universitat s/n, 46100 Burjassot, España

² iTeAM, Universitat Politecnica de Valencia, Cno. de Vera s/n, 46022 Valencia, España

³ IEEAC, Universidad de Castilla La Mancha, 16071 Cuenca, España

E-mail: Maximo.Cobos@uv.es

ABSTRACT

Spatial audio has been a topic of intensive research in the last decades. Although there are many tools available for developing real-time spatial sound systems, most of them work under audio-oriented frameworks. However, despite a significant number of signal processing researchers and engineers develop their algorithms in MATLAB, there is not currently any MATLAB-based tool for rapid spatial audio system prototyping and algorithm testing. This paper presents a tool for spatial audio research and education under this framework.

The presented tool provides the user with a friendly graphical user interface (GUI) that allows to move freely a number of sound sources in 3D and to develop specific functions to be used during their reproduction.

RESUMEN

El audio espacial ha sido un tema de intensa investigación en las últimas décadas. Aunque hay muchas herramientas disponibles para el desarrollo de sistemas de sonido espacial en tiempo real, la mayoría de ellas trabajan bajo entornos de audio orientado. Sin embargo, a pesar de que un importante número de investigadores e ingenieros de procesado de señales desarrollan sus algoritmos en MATLAB, no hay actualmente ninguna herramienta basada en MATLAB para el testeo de algoritmos y la creación rápida de prototipos de sistemas de audio espacial. Este artículo presenta una herramienta para la investigación y la educación en el campo del audio espacial.

a herramienta presentada proporciona al usuario una interfaz gráfica de usuario (GUI) que permite mover libremente una serie de fuentes de sonido en 3D y desarrollar funciones específicas que se utilizarán durante su reproducción.

1. INTRODUCCIÓN

Es un hecho bien conocido que audio espacial ha sido un tema de intensa investigación en las últimas décadas. Además de las técnicas estereofónicas y de sonido envolvente convencionales, se han propuesto una serie de diferentes enfoques para mejorar la reproducción de sonido espacial por altavoces y auriculares. En los últimos años, las técnicas de paneo avanzado como Vector Base Amplitud Panning (VBAP) [1] y métodos de síntesis de campo sonoro, como Wave Field Synthesis (WFS) [2] y Ambisonics [3] han atraído la mayor parte de la atención.

Hay muchas herramientas disponibles para el desarrollo de sistemas de sonido espacial en tiempo real, la mayoría de ellos trabajando bajo entornos de audio orientado como Pure Data [4], MAX MSP [5] o SuperCollider [6]. SoundScape Renderer (SSR) [7], que es un software de código abierto para la reproducción de audio espacial, ha demostrado ser una herramienta muy útil para la comunidad científica, permitiendo a los investigadores para llevar a cabo experimentos teniendo en cuenta los métodos de reproducción de audio espaciales más populares.

Sin embargo, a pesar de que un importante número de investigadores e ingenieros de procesado de señales desarrollan sus algoritmos bajo MATLAB, no hay actualmente ninguna herramienta basada en MATLAB para la creación rápida de prototipos de sistemas de audio espacial. MATLAB ofrece un excelente entorno de desarrollo para el procesado de señales, sin embargo, la grabación de audio en tiempo real y reproducción sobre MATLAB ha sido tradicionalmente un problema. Recientemente se ha incorporado el streaming de audio en tiempo real mediante el soporte de ASIO [8] (drivers reconocidos de baja latencia) a este software, proporcionando una buena oportunidad para probar algoritmos de audio espacial bajo un marco común.

Por otra parte, la programación orientada a objetos (OOP) permite el nivel de versatilidad y la abstracción necesaria para crear una herramienta general de reproducción de audio.

Este artículo presenta una herramienta basada en MATLAB para la educación e investigación de audio espacial. Se hace uso del "DSP System Toolbox" [9] de MATLAB, que proporciona las funcionalidades necesarias para la gestión de objetos de audio en tiempo real. La herramienta presentada proporciona al usuario una interfaz gráfica de usuario (GUI) que permite mover libremente un número de fuentes de sonido en 3D considerando cualquier configuración de altavoces. Los métodos de procesado de señales que se utilizan para reproducir una escena de sonido dada son independientes del resto de la herramienta. Por lo tanto, el usuario puede desarrollar funciones específicas que se utilizarán durante la reproducción de fuentes de sonido virtuales. La configuración inicial específica, entre otra información, la ubicación de los altavoces y la posición inicial de las fuentes virtuales, los archivos de audio asociados y el método de reproducción de audio seleccionado. La interfaz de la GUI se construye directamente desde la configuración inicial especificada cuando se inicia la herramienta. Muestra la configuración seleccionada tanto en vista en planta y perfil, permitiendo al usuario arrastrar y soltar las fuentes virtuales o insertar directamente una posición deseada mediante el ratón. Las fuentes pueden ser también silenciadas durante la reproducción.

Actualmente, la herramienta implementa algunas de las técnicas de reproducción de audio más populares: Binaural, VBAP y Ambisonics Amplitud Panning (AAP) [10]. Si bien pueden existir algunas limitaciones al utilizar MATLAB en lugar de otros entornos orientados a audio, esta herramienta ofrece una solución muy útil para la prueba rápida de algoritmos y el desarrollo de pruebas de concepto. Por otra parte, puede ser también muy útil en la educación para la práctica en conferencias y laboratorios, donde los estudiantes pueden mejorar sus habilidades de procesamiento de audio al escuchar directamente al resultado de sus funciones desarrolladas para audio espacial. Aunque la herramienta puede ser enriquecida con nuevos algoritmos y funcionalidades, la situación actual ya muestra un gran potencial para el avance de la investigación de audio espacial.

2. ARQUITECTURA DE SOFTWARE

Todo el código ha sido desarrollado bajo MATLAB mediante el uso de una interfaz gráfica de usuario y sus funcionalidades OOP. El uso de MATLAB en la comunidad de investigación es altamente extendida ya que ofrece una plataforma independiente que permite que la herramienta se ejecute bajo diferentes sistemas operativos. Los flujos de audio son gestionados por la herramienta "DSP System Toolbox", que permite la reproducción de audio en tiempo real y la grabación utilizando diferentes tipos de controladores de audio. En cualquier caso, los controladores ASIO son muy recomendables para tratar con múltiples canales de audio con baja latencia. Para una asignación adecuada de los canales de audio en esta herramienta, se necesita la versión 2013b de MATLAB. Además, muchas de las funciones de representación hacen uso de la herramienta "Signal Processing Toolbox". Hay que tener en cuenta que las dos herramientas se incluyen en la versión estándar. El proyecto está organizado en diferentes directorios que contienen los scripts y funciones necesarios. Todos los nombres de los archivos han sido cuidadosamente seleccionados para ser reconocidos fácilmente.

2.1. Niveles de software

La Figura 1 muestra una representación general de los diferentes niveles de software involucrados. En el nivel superior, la interfaz gráfica de usuario ofrece las funcionalidades de interacción del usuario. A continuación, DATA y HANDLES (que son gestionados por "guidata" y "guihandles") almacenan los parámetros necesarios durante la reproducción y los objetos gráficos, respectivamente. En el mismo nivel, tenemos objetos que son requeridos tanto para audio como para gestión gráfica, todos ellos con funcionalidades de POO.

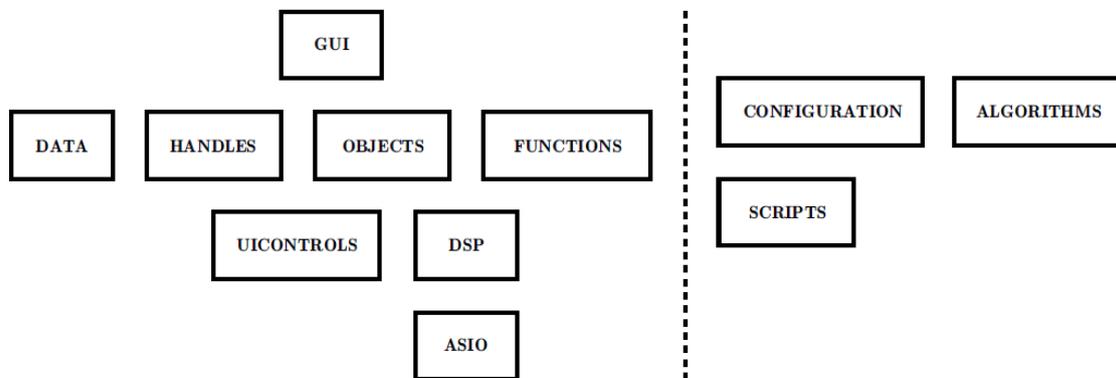


Figura 1. Niveles de software.

Además, las funciones necesarias para el funcionamiento de todos los elementos están incluidas en este nivel. En este nivel, nos encontramos con los UICONTROLS de MATLAB y la biblioteca DSP, que ambos son utilizados por los niveles superiores. En el nivel inferior, se encuentran los controladores de audio ASIO que se comunican con el hardware de audio. Aparte de estos niveles, CONFIGURATION y SCRIPTS se utilizan para cargar y guardar la configuración de la reproducción. Los algoritmos, (ALGORITHMS) son funciones independientes que el usuario puede modificar libremente o implementar nuevos, para añadir más funcionalidades a la herramienta.

2.2. Configuración de la Escena de Audio

Para iniciar la interfaz gráfica de usuario se necesita una estructura inicial de configuración de MATLAB. Esta estructura contiene todos los parámetros que describen la configuración de reproducción deseada, como la dimensión de la escena de audio, el método de representación, la asignación de canales, etc. Uno de los parámetros más importantes son las posiciones de los altavoces y las fuentes de audio. La estructura puede ser generada fácilmente por una función MATLAB que se puede modificar para adaptarse a las necesidades del usuario. Diferentes configuraciones de altavoces y fuentes de sonido se pueden cargar directamente gracias a esta función. El usuario puede generar sus propios scripts que contengan la posición inicial de las fuentes y los altavoces. Esta es una manera muy fácil de gestionar el sistema de configuración inicial de los usuarios estándar de MATLAB.

2.3. Renderizado

El procesado de las señales de audio correspondientes a las diferentes fuentes se realiza en tiempo real y es reproducido por los altavoces en consecuencia para el método de representación seleccionada. Cada vez que se inicia la herramienta, se selecciona un método de representación según lo especificado por la estructura de configuración inicial. Como siempre, esto puede ser modificado por el usuario sin cerrar el programa siempre que la reproducción de audio se detenga y la configuración de reproducción permita el método de representación seleccionado (por ejemplo, la reproducción binaural no se puede seleccionar si el número de altavoces es mayor que dos).

Uno de los aspectos más interesantes de esta herramienta es que el usuario puede desarrollar funciones adicionales correspondientes a nuevos métodos de representación e incorporarlas a este software sin esfuerzo o cambiando el resto del código, ya que la arquitectura ha sido creada atendiendo al máximo al patrón Modelo-Vista-Controlador (MVC) [1]. En nuestro entorno, el modelo, que está formado por los diferentes algoritmos de representación, está completamente separado de la interfaz gráfica y el núcleo de audio. Para añadir un nuevo método de representación, es suficiente seguir los parámetros de entrada-salida solicitada de la estructura por defecto y que se seleccione antes de la reproducción. Para mantener este concepto independiente de las principales funciones de gestión de audio, la salida del método de representación debe ser una matriz de dimensiones $M_s \times L$ donde M_s es el número de altavoces seleccionados que contribuyen al renderizado y L la longitud del filtro. Por otra parte, se debe proporcionar un vector de dimensiones $M_s \times 1$, especificando los índices de los altavoces activos para la fuente. Hay que tener en cuenta que para los algoritmos basados en paneo, L es siempre 1 (se utilizan coeficientes escalares de amplitud en lugar de respuestas impulsionales). Los coeficientes del filtro para cada altavoz y la fuente de sonido se almacenan en los datos de la GUI. De esta manera, la interfaz gráfica de usuario es totalmente independiente de la Método de renderizado seleccionado.

La Figura 2 muestra un esquema general de las etapas de procesado que sigue la herramienta. Las señales de las fuentes de entrada $x_k[n]$, $k = 1, \dots, K$, correspondientes a las K fuentes de la escena de audio, son procesadas por los filtros $H_{km}(\omega, \theta_k, \phi_k)$ en relación con cada uno de los altavoces M en el setup, donde θ_k , ϕ_k y r_k son las coordenadas esféricas, azimut, elevación y radio, de la fuente. Antes de filtrar, se aplica un factor de atenuación con la distancia $\alpha(r_k)$ aplicando la regla de atenuación $1/r$. Debemos destacar que el diagrama representado en la Figura 2 es una generalización del modo de funcionamiento de la herramienta en el sentido de que todos los altavoces se representan como altavoces contributivos. Sin embargo, para cada fuente de sonido, no todos los altavoces en la configuración están contribuyendo a la síntesis de dicha fuente, por lo que el filtrado se realiza internamente sólo para los altavoces seleccionados según lo especificado por el algoritmo de renderizado.

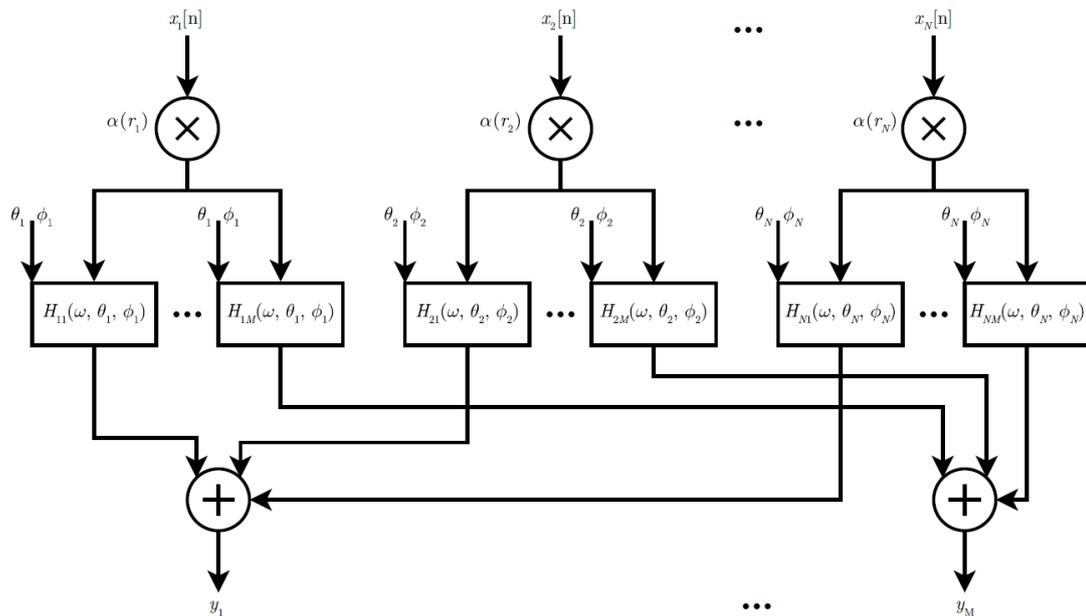


Figura 2. Flujo general de las señales.

2.4. Audio Streaming

Una de las bibliotecas clave de esta herramienta es la biblioteca de MATLAB “DSP System Toolbox”, especialmente el modelo “Signal Import and Export Model”. En esta toolbox, los objetos de audio de entrada (fuentes) se adquieren por medio de la clase `dsp.AudioFileReader`, mientras que los objetos de salida de audio (altavoces) utilizan la clase `dsp.AudioPlayer`. Se debe tener en cuenta que para sintetizar directamente una entrada de audio en vivo en lugar de leer desde un archivo de sonido grabado, será requerida la clase `dsp.AudioRecorded`.

Para evitar “clics” en el audio al cambiar los parámetros de renderizado, el sistema maneja tramas de audio consecutivas utilizando desvanecimiento cruzado o *cross-fading*. Se trata de un enfoque similar al seguido en el SSR [7]. Una trama dada se procesa con los parámetros actuales de representación en el instante de tiempo n y con los parámetros de renderizado anteriores en el $n-1$. Las salidas resultantes se cruzan, con pendientes de desvanecimiento en forma de coseno. Por otra parte, para evitar cambios muy bruscos de parámetros, un leve factor de memoria está incluido en el proceso, por lo que los parámetros de renderizado finales en un instante de tiempo dado son una mezcla de los parámetros anteriores y actuales. Dependiendo de este factor de memoria, puede cambiar el tiempo requerido para actualizar totalmente los parámetros en la salida de audio.

2.5. Administración de recursos

La herramienta ofrece la posibilidad de modificar los parámetros de procesamiento de señales y opciones de visualización con el fin de optimizar los recursos de la plataforma de ejecución. Este es un tema importante, ya que, siempre y cuando el número de fuentes y altavoces se aumenta, el coste computacional también se ve afectado sustancialmente, tanto en términos de procesamiento de señal como de visualización. Esto es especialmente importante en el último caso debido a las limitaciones de MATLAB para el manejo de muchos objetos gráficos en tiempo real.

2.6. Inicialización de la GUI

La función principal que inicia la herramienta iniciar la herramienta también es responsable de cargar todos los objetos gráficos (y no gráficos) del sistema según lo especificado por la estructura de configuración inicial. Una vez que la configuración de escena se ha cargado, los objetos gráficos que representan los altavoces se colocan en la interfaz gráfica de usuario en dos vistas: una vista de la planta (proyección xy) y una vista de perfil (proyección yz).

El sistema de coordenadas seguido en el posicionamiento de los diferentes elementos debe tenerse en cuenta cuidadosamente. El centro del sistema de coordenadas es siempre la cabeza del oyente, con el aumento de x en la dirección de la oreja derecha, y el aumento en la dirección delantera y z creciente hacia el techo.

Sin embargo, los ángulos de azimut y elevación no siguen su correspondencia habitual a las coordenadas cartesianas, ya que, por razones de simplicidad y de acuerdo a la literatura de audio espacial, la dirección del azimut cero, $\theta = 0$ está en el frente del oyente y aumenta en sentido antihorario. El ángulo de elevación cero está en el plano xy y aumenta hacia el eje z . La Figura 3 muestra la correspondencia de las coordenadas esféricas y cartesianas. El radio r es siempre la distancia al centro del sistema de coordenadas.

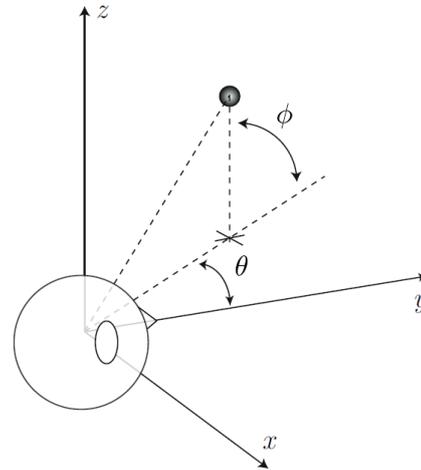


Figura 3. Sistema de coordenadas utilizado.

La herramienta presentada tiene la capacidad de crear escenas de sonido, tanto en 2 como en 3 dimensiones. Sin embargo, debe tenerse en cuenta que, dependiendo del método de representación seleccionado, sólo podrán ser representadas escenas de dos dimensiones. Esto se comprueba en la inicialización, que muestra una advertencia / error si la configuración de origen especificada no es compatible con el método de representación.

3. MÓDULOS DE RENDERIZADO.

Esta sección describe los diferentes métodos de renderizado que se implementan actualmente en la herramienta presentada.

3.1. Sonido Binaural.

Para sintetizar escenas de sonido binaural, se debe especificar en el conjunto de datos una función de transferencia relativa a la cabeza, Head-Related Transfer Function (HRTF) [12] en lo sucesivo. Para este fin, el archivo de configuración debe almacenar los nombres de los archivos que contienen las respuestas impulsionales relativas a la cabeza, Head-Related Impulse (HRIRs), y sus direcciones asociadas. Obviamente, este método de representación sólo utiliza dos altavoces (auriculares), pero permite la reproducción de una escena 3D. Los objetos de filtrado se crean teniendo en cuenta las indicaciones de las fuentes y las instrucciones disponibles en el conjunto de datos HRTF. La herramienta selecciona el HRIR más cercano a la posición de la fuente en el conjunto de datos y filtra la señal de la fuente en consecuencia.

Actualmente, el sistema no interpola las respuestas, pero se realiza un proceso fade-in / fade-out aplicado a los flujos de audio que proporciona cierta interpolación. Como un ejemplo, la herramienta sí que implementa la síntesis mediante el uso del conjunto de datos KEMAR del MIT [14]. Teniendo en cuenta el proceso seguido para incorporar este conjunto de datos, cualquier otro conjunto de datos disponibles se puede incorporar fácilmente. Hay que tener en cuenta también que la mayoría de conjuntos de datos de HRTF no incluyen información de la distancia, de modo que sólo se aplica a la señal de entrada un factor de atenuación de acuerdo con su distancia al oyente.

3.2. Vector Base Amplitude Panning.

VBAP es un método basado en el paneo de amplitud según el posicionamiento de una fuente virtual 3D. Se puede entender como una generalización de panoramización estéreo a 3D considerando tripletes de altavoces. Básicamente, después de seleccionar el triplete de altavoces que contiene la dirección de la fuente, el vector que apunta a la fuente se proyecta sobre las direcciones de los altavoces que forman el triplete. Los coeficientes resultantes son los que se utilizan para realizar el paneo de amplitud. Una vez más, un factor de atenuación se tiene en cuenta para simular la distancia a la fuente virtual.

3.3. Ambisonics Amplitude Panning.

Ambisonics es una técnica de espacialización de sonido basado en la descomposición armónica esférica del campo de sonido. Esta herramienta implementa AAP, que realiza un panning simultáneo de la fuente mediante el uso de todos los altavoces disponibles en la configuración de la reproducción. Para implementar esta técnica correctamente, todos los altavoces en la configuración deben tener la misma separación angular en un círculo con el centro situado en la cabeza del oyente (si este no es el caso, la herramienta notificará una advertencia). La prestación es similar a la utilizada en SSR que utiliza la función de desplazamiento que se describe en [13].

4. INTERFAZ GRÁFICA DE USUARIO

La interfaz gráfica de usuario (ver Figura 4) representa el entorno que permite la interacción del usuario. Por lo tanto, se ha prestado especial atención a todos los aspectos relacionados con el diseño de esta interfaz. Como resultado, la interfaz gráfica de usuario ofrece un aspecto muy simple, con cajas de entrada que muestran las ubicaciones de origen y que permiten modificar las posiciones actuales en cualquier momento. De hecho, la interfaz es tan simple que cualquier usuario inexperto puede interactuar fácilmente con las diferentes fuentes en tiempo real desde el primer contacto. El usuario puede mover las fuentes, ya sea arrastrando y soltando o utilizando los cuadros de texto de entrada. La entrada se puede proporcionar tanto en coordenadas cartesianas como esféricas (véase la Sección 2.6 para más detalles).

La función de arrastrar y soltar es una de las principales ventajas de la herramienta, ya que, además de ser más intuitiva, por lo general es el método preferido para la interacción del usuario. Sin embargo, este tipo de control no es tan eficiente implementado en MATLAB y consume una gran cantidad de recursos, por lo tanto se ha tenido especial cuidado en la aplicación de tal función. Los objetos gráficos están relacionados entre sí de modo que, si uno de ellos sufre un cambio, este afecta de forma instantánea los otros en tiempo real. Esto permite al usuario conocer con exactitud la posición real de cada fuente en cualquier momento.

Un aspecto relevante a destacar de esta interfaz gráfica es que se sigue un enfoque programático completo en lugar de utilizar la herramienta “GUIDe” habitual de MATLAB. Como resultado, la interfaz gráfica de usuario es construida totalmente al iniciar la herramienta según lo especificado por la estructura de configuración inicial, ubicando los objetos gráficos según la configuración de la escena y reproducción de sonido previsto.

Los altavoces son representados por círculos negros marcados con su número de canal de audio correspondiente. El cambio de las posiciones de altavoces cuando la herramienta está en funcionamiento no es posible. Esto es porque el cambio de ubicaciones de altavoces durante la reproducción no es una característica deseable (ni común). Por otra parte, manteniendo las ubicaciones de altavoces sin cambios reducimos el coste computacional, ya que la herramienta no necesita para comprobar los cambios de posición.

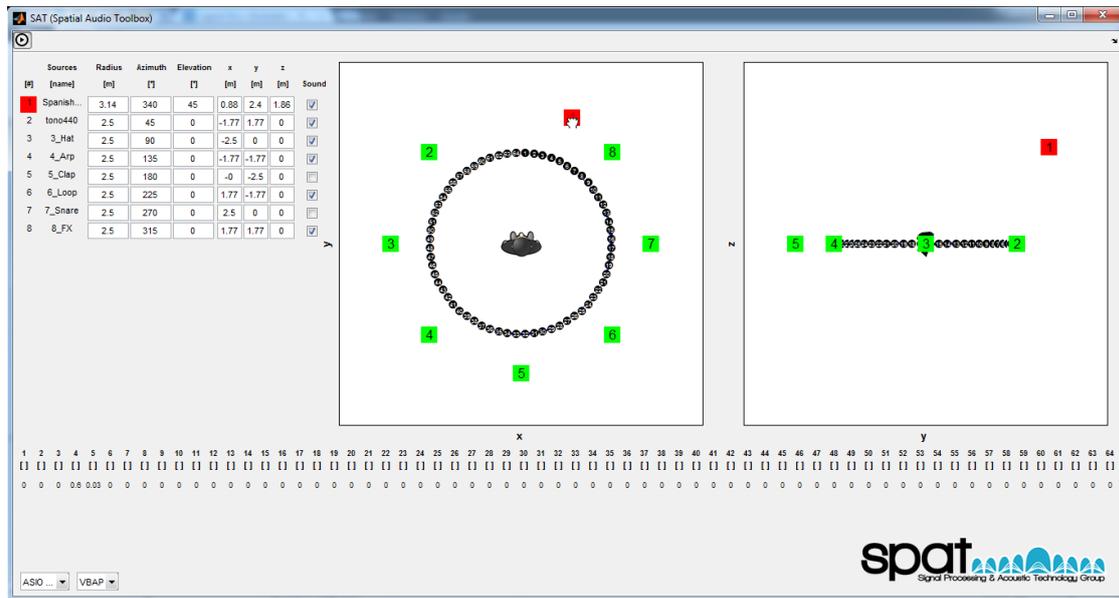


Figura 4. Captura de pantalla de la herramienta.

Las fuentes virtuales están representadas por cuadrados verdes marcadas con su índice de fuente del sonido correspondiente. Cuando el puntero del ratón está en contacto con cualquiera de las fuentes, toma la forma de una mano abierta, y cambia a una mano cerrada cuando se hace clic, cambiando el color de la fuente a rojo. Entonces, el usuario puede arrastrar el ratón mientras mantiene presionado el botón para mover la fuente. El color cambia de nuevo a verde cuando se suelta el botón del ratón.

Para iniciar la reproducción de sonido de la escena, el usuario sólo tiene que pulsar el botón de reproducción situado en la barra de tareas. El audio continuará reproduciéndose a menos que el usuario pulsa de nuevo el mismo botón. Hay que tener en cuenta que el método de representación y el controlador de audio seleccionado no se pueden modificar mientras la reproducción está encendida. Además, se proporciona un botón de verificación para cada fuente para permitir al usuario especificar cuáles de las fuentes de sonido deben ser reproducidas en cualquier momento (incluso durante la reproducción).

Una cuestión muy importante es que los parámetros de representación relacionada con cada fuente sólo se actualizan cuando sea necesario (cuando hay cualquier cambio en la ubicación de origen). Como resultado, se obtiene un gran ahorro computacional.

5. CONCLUSIONES Y TRABAJO FUTURO

Se ha presentado una herramienta de reproducción de audio espacial basada en MATLAB. La herramienta ha sido desarrollada mediante el uso de una interfaz gráfica de usuario programática y funcionalidades POO de MATLAB. La motivación para el desarrollo de esta herramienta reside en el hecho de que el uso de MATLAB está muy extendido en la comunidad de investigación y ofrece una plataforma independiente que permite que la herramienta se ejecute bajo diferentes sistemas operativos. Los algoritmos de renderizado utilizados por la herramienta presentada son independientes de la interfaz gráfica de usuario y los procesos de streaming de audio subyacentes, lo que permite a los investigadores y profesionales de la educación centrarse en los aspectos de procesamiento de señales en lugar de en la interacción de datos de usuario o la gestión de los controladores de audio. Actualmente, la herramienta implementa algoritmos básicos de representación como representación binaural, VBAP y AAP.

La estructura de la herramienta permite a un usuario inexperto incorporar nuevas funcionalidades sin mucho esfuerzo. Además, la estructura independiente entre los diferentes elementos en el sistema, hace que sea muy fácil modificar imágenes, archivos de audio, dimensiones gráficas, etc.

Actualmente, la herramienta está todavía en desarrollo, con especial atención a la aplicación de métodos Wave Field Synthesis (WFS) o Higher Order Ambisonics (HOA). Por otra parte, se pretende adaptar el código para leer archivos para la configuración de la escena de audio siguiendo XML Audio Scene Description Format (ASDF) de forma similar al SoundScape Renderer [7].

6. REFERENCIAS

1. **Pulkki, V.** (1997). Virtual Sound Source Positioning Using Vector Base Amplitude Panning. *Journal of the AES*, 45(6), 456-466.
2. **Spors, S., Rabenstein, R., and Ahrens, J.** (2008, May). The theory of wave field synthesis revisited. In 124th AES Convention, Amsterdam, The Netherlands (pp. 17-20).
3. **Gerzon, M. A.** (1985). Ambisonics in multichannel broadcasting and video. *Journal of the AES*, 33(11), 859-871.
4. **Puckette, M.** (1996). Pure Data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, 37-41.
5. **Puckette, M. S., Ucsd, M. S. P., and Apel, T.** (1998). Real-time audio analysis tools for Pd and MSP.
6. **McCartney, J.** (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, 26(4), 61-68.
7. **Ahrens, J., Geier, M., and Spors, S.** (2008, May). The SoundScape Renderer: A Unified Spatial Audio Reproduction Framework for Arbitrary Rendering Methods. In 124th AES Convention, Amsterdam, The Netherlands.
8. Steinberg, ASIO.
http://www.steinberg.net/nc/en/company/developers/sdk_download_portal.html
9. Mathworks. DSP System Toolbox. <http://mathworks.com/help/dsp/>
10. **Ahrens, J., and Spors, S.** (2007, October). Rendering of Virtual Sound Sources with Arbitrary Directivity in Higher Order Ambisonics. In 123th AES Convention, New York, USA.
11. **Bucanek, J.** (2009). Model-View-Controller Pattern. *Learn Objective-C for Java Developers*, 353-402.

12. **Cheng, C. I., and Wake_eld, G. H.** (1999, September). Introduction to Head-Related Transfer Functions (HRTFs): Representations of HRTFs in Time, Frequency, and Space. In 107th AES Convention, New York, USA.
13. **Neukom, M.** (2007, October). Ambisonic Panning. In 123th AES Convention, New York, USA.
14. **Gardner, B., and Martin, K.** (1994). HRFT Measurements of a KEMAR Dummy-head Microphone.