

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 645 432**

51 Int. Cl.:

G10L 19/24 (2013.01)

G10L 19/008 (2013.01)

G10L 19/018 (2013.01)

H04S 3/02 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.09.2014 PCT/US2014/057611**

87 Fecha y número de publicación internacional: **02.04.2015 WO15048387**

96 Fecha de presentación y número de la solicitud europea: **26.09.2014 E 14781027 (9)**

97 Fecha y número de publicación de la concesión europea: **13.09.2017 EP 3050055**

54 Título: **Renderización de audio multicanal mediante la utilización de matrices interpoladas**

30 Prioridad:

27.09.2013 US 201361883890 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
05.12.2017

73 Titular/es:

**DOLBY LABORATORIES LICENSING
CORPORATION (100.0%)
1275 Market Street
San Francisco, CA 94103, US**

72 Inventor/es:

**LAW, MALCOLM J.;
MELKOTE, VINAY;
WILSON, RHONDA;
PLAIN, SIMON y
JASPAR, ANDY**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 645 432 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Renderización de audio multicanal mediante la utilización de matrices interpoladas.

Campo técnico

5 La invención se refiere al procesamiento de señales de audio y más en concreto a la renderización de programas de audio multicanal (por ejemplo, flujos de bits indicativos de programas de audio basados en objetos que incluyen al menos un canal de objeto de audio y al menos un canal de altavoz) mediante la utilización de matrices interpoladas, y a la codificación y la decodificación de los programas. En algunas realizaciones, un decodificador lleva a cabo la interpolación en un conjunto de matrices primitivas semilla para determinar las matrices interpoladas para su uso en la renderización de canales del programa. Algunas realizaciones generan, decodifican y/o renderizan datos de audio en el formato conocido como Dolby TrueHD.

Antecedentes

Dolby y Dolby TrueHD son marcas registradas de Dolby Laboratories Licensing Corporation.

15 La complejidad, y el coste financiero y computacional, de la renderización de programas de audio aumenta en función del número de canales que se van a renderizar. Durante la renderización y la reproducción de programas de audio basados en objetos, el contenido de audio tiene un número de canales (por ejemplo, canales de objeto y canales de altavoz) que es típicamente mucho mayor (por ejemplo, en un orden de magnitud) que el número que se da durante la renderización y la reproducción de programas basados en canales de altavoz convencionales. Típicamente también, el sistema de altavoces utilizado para la reproducción incluye un número mucho mayor de altavoces que el número empleado para la reproducción de los programas basados en canales de altavoz convencionales.

Aunque las realizaciones de la invención son útiles para la renderización de los canales de cualquier programa de audio multicanal, muchas realizaciones de la invención son especialmente útiles para la renderización de los canales de programas de audio basados en objetos que tienen un número elevado de canales.

25 Es conocido el empleo de sistemas de reproducción (por ejemplo, en los cines) para renderizar programas de audio basados en objetos. Los programas de audio basados en objetos pueden ser indicativos de muchos objetos de audio diferentes correspondientes a imágenes en una pantalla, diálogo, ruidos y efectos sonoros que provienen de diferentes lugares en (o relativos a) la pantalla, así como la música de fondo y efectos del ambiente (que pueden ser indicados por canales de altavoz del programa) para crear la experiencia auditiva general prevista. La reproducción precisa de dichos programas requiere que los sonidos se reproduzcan de una manera que corresponda en la mayor medida posible a lo que se pretende por el creador de contenido con respecto al tamaño, posición, intensidad, movimiento y profundidad del objeto de audio.

35 Durante la generación de programas de audio basados en objetos, se supone típicamente que los altavoces que se van a emplear para la renderización están situados en ubicaciones arbitrarias en el entorno de reproducción; no necesariamente en una disposición predeterminada en un plano (nominalmente) horizontal o en cualquier otra disposición predeterminada conocida en el momento de la generación del programa. Típicamente, los metadatos incluidos en el programa indican los parámetros de renderización para la renderización de al menos un objeto del programa en una ubicación espacial evidente o a lo largo de una trayectoria (en un volumen tridimensional), por ejemplo, utilizando un arreglo tridimensional de altavoces. Por ejemplo, un canal de objeto del programa puede tener metadatos correspondientes que indican una trayectoria tridimensional de posiciones espaciales evidentes en las que el objeto (indicado por el canal de objeto) ha de renderizarse. La trayectoria puede incluir una secuencia de ubicaciones "en el suelo" (en el plano de un subconjunto de altavoces que se supone que están situados en el suelo, o en otro plano horizontal, del entorno de reproducción), y una secuencia de ubicaciones "por encima del suelo" (cada una determinada moviendo un subconjunto de los altavoces que se supone que están situados en al menos otro plano horizontal del entorno de reproducción).

45 Los programas de audio basados en objetos representan una mejora significativa en muchos aspectos en comparación con los programas de audio basados en canales de altavoz tradicionales, dado que el audio basado en canales de altavoz es más limitado con respecto a la reproducción espacial de objetos de audio específicos que el audio basado en canales de objeto. Los programas de audio basados en canales de altavoz constan de canales de altavoz solamente (no canales de objeto) y cada canal de altavoz determina típicamente una señal de altavoz para un altavoz individual específico en un entorno de escucha.

55 Se han propuesto diversos métodos y sistemas para la generación y la renderización de programas de audio basados en objetos. Durante la generación de un programa de audio basado en objetos, se supone típicamente que un número arbitrario de altavoces serán empleados para la reproducción del programa y que los altavoces que se van a emplear para la reproducción estarán situados en ubicaciones arbitrarias en el entorno de reproducción; no

necesariamente en un plano (nominalmente) horizontal o en cualquier otra disposición predeterminada conocida en el momento de la generación del programa. Típicamente, los metadatos relacionados con objetos incluidos en el programa indican los parámetros de renderización para la renderización de al menos un objeto del programa en una ubicación espacial evidente o a lo largo de una trayectoria (en un volumen tridimensional), por ejemplo, utilizando un arreglo tridimensional de altavoces. Por ejemplo, un canal de objeto del programa puede tener metadatos correspondientes que indican una trayectoria tridimensional de posiciones espaciales evidentes en las que el objeto (indicado por el canal de objeto) ha de renderizarse. La trayectoria puede incluir una secuencia de ubicaciones "en el suelo" (en el plano de un subconjunto de altavoces que se supone que están situados en el suelo, o en otro plano horizontal, del entorno de reproducción), y una secuencia de ubicaciones "por encima del suelo" (cada una determinada moviendo un subconjunto de los altavoces que se supone que están situados en al menos otro plano horizontal del entorno de reproducción). Ejemplos de renderización de programas de audio basados en objetos se describen, por ejemplo, en la Solicitud Internacional PCT N.º PCT/US2001/028783, publicada bajo la Publicación Internacional N.º WO 2011/119401 A2 el 29 de septiembre de 2011 y cedida al cesionario de la presente solicitud.

Un programa de audio basado en objetos puede incluir canales "de base". Un canal de base puede ser un canal de objeto indicativo de un objeto cuya posición no cambia durante el intervalo de tiempo pertinente (y por lo tanto se renderiza típicamente utilizando un conjunto de altavoces de sistema de reproducción que tienen ubicaciones de altavoz estáticas) o puede ser un canal de altavoz (que se va a renderizar por un altavoz específico de un sistema de reproducción). Los canales de base no tienen metadatos de posición variables en el tiempo correspondientes (aunque se puede considerar que tienen metadatos de posición invariables en el tiempo). Pueden ser indicativos de elementos de audio que están dispersos en el espacio, por ejemplo, audio indicativo del ambiente.

La reproducción de un programa de audio basado en objetos en una configuración de altavoz tradicional (por ejemplo, un sistema de reproducción 7.1) se consigue mediante la renderización de los canales del programa (incluidos los canales de objeto) a un conjunto de señales de altavoz. En realizaciones típicas de la invención, el proceso de renderización de los canales de objeto (a veces denominados en la presente memoria "objetos") y de otros canales de un programa de audio basado en objetos (o canales de un programa de audio de otro tipo) comprende en gran medida (o únicamente) una conversión de metadatos espaciales (para los canales que se van a renderizar) en cada instante de tiempo en una matriz de ganancia correspondiente (denominada en la presente memoria "matriz de renderización") que representa cuánto contribuye cada uno de los canales (por ejemplo, canales de objeto y canales de altavoz) a una mezcla de contenido de audio (en el instante) indicada por la señal de altavoz para un altavoz particular (es decir, el peso relativo de cada uno de los canales del programa en la mezcla indicada por la señal de altavoz).

Un "canal de objeto" de un programa de audio basado en objetos es indicativo de una secuencia de muestras indicativas de un objeto de audio y el programa incluye típicamente una secuencia de valores de metadatos de posición espacial indicativos de la posición de objeto o la trayectoria para cada canal de objeto. En realizaciones típicas de la invención, se utilizan secuencias de valores de metadatos de posición correspondientes a canales de objeto de un programa para determinar una matriz $A(t)$ de $M \times N$ indicativa de una especificación de ganancia variable en el tiempo para el programa.

La renderización de "N" canales (por ejemplo, canales de objeto, o canales de objeto y canales de altavoz) de un programa de audio a "M" altavoces (señales de altavoz) en el tiempo "t" del programa se puede representar por la multiplicación de un vector $x(t)$ de longitud "N", compuesto por una muestra de audio en el tiempo "t" de cada canal, por una matriz $A(t)$ de $M \times N$ determinada a partir de los metadatos de posición asociados (y opcionalmente otros metadatos correspondientes al contenido de audio que se va a renderizar, por ejemplo, las ganancias de objeto) en el tiempo "t". Los valores resultantes (por ejemplo, ganancias o niveles) de las señales de altavoz en el tiempo t se pueden representar como un vector $y(t)$, como en la siguiente ecuación (1):

$$\begin{matrix}
 \begin{bmatrix} y_0(t) \\ y_1(t) \\ \vdots \\ y_{M-1}(t) \end{bmatrix} \\
 \mathbf{y}(t)
 \end{matrix}
 =
 \begin{matrix}
 \begin{bmatrix} a_{00}(t) & a_{01}(t) & a_{02}(t) & \ddots & a_{0,N-1}(t) \\ a_{10}(t) & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ a_{M-1,0}(t) & \ddots & \ddots & \ddots & a_{M-1,N-1}(t) \end{bmatrix} \\
 \mathbf{A}(t)
 \end{matrix}
 \begin{matrix}
 \begin{bmatrix} x_0(t) \\ x_1(t) \\ x_2(t) \\ \vdots \\ x_{N-1}(t) \end{bmatrix} \\
 \mathbf{x}(t)
 \end{matrix}
 \quad (1)$$

Aunque la ecuación (1) describe la renderización de N canales de un programa de audio (por ejemplo, un programa de audio basado en objetos o una versión codificada de un programa de audio basado en objetos) a M canales de salida (por ejemplo, M señales de altavoz), también representa un conjunto genérico de escenarios en los que un conjunto de N muestras de audio se convierten en un conjunto de M valores (por ejemplo, M muestras) mediante operaciones lineales. Por ejemplo, $A(t)$ podría ser una matriz estática, "A", cuyos coeficientes no varían con valores diferentes del tiempo "t". Como otro ejemplo, $A(t)$ (que podría ser una matriz estática, A) podría representar una

mezcla descendente convencional de un conjunto de canales de altavoz $\mathbf{x}(t)$ a un conjunto más reducido de canales de altavoz $\mathbf{y}(t)$ (o $\mathbf{x}(t)$ podría ser un conjunto de canales de audio que describen una escena espacial en un formato Ambisonics), y la conversión a señales de altavoz $\mathbf{y}(t)$ se podría establecer como la multiplicación por la matriz de mezcla descendente \mathbf{A} . Incluso en una aplicación que emplea una matriz de mezcla descendente nominalmente estática, la transformación lineal real (multiplicación de matriz) aplicada puede ser dinámica con el fin de asegurar la protección de recorte de la mezcla descendente (es decir, una transformación estática \mathbf{A} se puede convertir en una transformación variable en el tiempo $\mathbf{A}(t)$, para asegurar la protección de recorte).

Un sistema de renderización de programa de audio (por ejemplo, un decodificador que implementa dicho sistema) puede recibir metadatos que determinan las matrices de renderización $\mathbf{A}(t)$ (o puede recibir las propias matrices) solo de forma intermitente y no en cada instante " t " durante un programa. Por ejemplo, esto podría ser debido a cualquiera de una variedad de razones, por ejemplo, una resolución de tiempo baja del sistema que emite realmente los metadatos o la necesidad de limitar la velocidad de bits de la transmisión del programa. Los inventores han reconocido que puede ser conveniente que un sistema de renderización interpole entre las matrices de renderización $\mathbf{A}(t_1)$ y $\mathbf{A}(t_2)$, en instantes de tiempo " t_1 " y " t_2 " durante un programa, respectivamente, para obtener una matriz de renderización $\mathbf{A}(t_3)$ para un instante de tiempo intermedio " t_3 ". La interpolación asegura que la posición percibida de los objetos en las señales de altavoz renderizadas varía suavemente con el tiempo y puede eliminar artefactos no deseables tales como el ruido de cremallera que provienen de actualizaciones de matriz discontinuas (constante a intervalos). La interpolación puede ser lineal (o no lineal), y típicamente debería asegurar una trayectoria continua en el tiempo desde $\mathbf{A}(t_1)$ hasta $\mathbf{A}(t_2)$.

El Dolby TrueHD es un formato de códec de audio convencional que admite una transmisión sin pérdidas y escalable de señales de audio. El audio fuente se codifica en una jerarquía de subflujos de canales, y un subconjunto seleccionado de los subflujos (en lugar de todos los subflujos) puede ser recuperado del flujo de bits y decodificado, con el fin de obtener una presentación dimensional (mezcla descendente) inferior de la escena espacial. Cuando se decodifican todos los subflujos, el audio resultante es idéntico al audio fuente (la codificación, seguida por la decodificación, es sin pérdidas).

En una versión disponible comercialmente del TrueHD, el audio fuente es típicamente una mezcla de canales 7.1 que se codifica en una secuencia de 3 subflujos, que incluye un primer subflujo que puede ser decodificado para determinar una mezcla descendente de 2 canales del audio original de canales 7.1. Los primeros 2 subflujos se pueden decodificar para determinar una mezcla descendente de canales 5.1 del audio original. Los tres subflujos pueden ser decodificados para determinar el audio de canales 7.1 original. Los detalles técnicos del Dolby TrueHD, y la tecnología del Empaquetado sin Pérdidas de Meridian (MLP, Meridian Lossless Packing) en la que se basa, son bien conocidos. Los aspectos del TrueHD y la tecnología de MLP se describen en la Patente de EE. UU. 6,611,212, expedida el 26 de agosto de 2003 y cedida a Dolby Laboratories Licensing Corp., y el artículo de Gerzon *et al.*, titulado "The MLP Lossless Compression System for PCM Audio", J. AES, vol. 52, n.º 3, páginas 243-260 (marzo de 2004).

El TrueHD admite la especificación de las matrices de mezcla descendente. En su uso típico, el creador de contenido de un programa de audio de canales 7.1 especifica una matriz estática para efectuar la mezcla descendente del programa de canales 7.1 hasta una mezcla de canales 5.1, y otra matriz estática para efectuar la mezcla descendente de la mezcla descendente de canales 5.1 hasta una mezcla descendente de 2 canales. Cada matriz de mezcla descendente estática se puede convertir en una secuencia de matrices de mezcla descendente (cada matriz en la secuencia para efectuar la mezcla descendente de un intervalo diferente en el programa) con el fin de conseguir la protección de recorte. Sin embargo, cada matriz en la secuencia se transmite (o los metadatos que determinan cada matriz en la secuencia se transmiten) al decodificador, y el decodificador no lleva a cabo la interpolación en ninguna matriz de mezcla descendente especificada previamente para determinar una matriz posterior en una secuencia de matrices de mezcla descendente para un programa.

La Figura 1 es un diagrama esquemático de los elementos de un sistema TrueHD convencional, en el que el codificador (30) y el decodificador (32) están configurados para implementar operaciones de matrización en muestras de audio. En el sistema de la Figura 1, el codificador 30 está configurado para codificar un programa de audio de 8 canales (por ejemplo, un conjunto tradicional de señales de altavoz 7.1) como un flujo de bits codificado que incluye 2 subflujos, y el decodificador 32 está configurado para decodificar el flujo de bits codificado para renderizar el programa de 8 canales original (sin pérdidas) o una mezcla descendente de 2 canales del programa de 8 canales original. El codificador 30 está acoplado y configurado para generar el flujo de bits codificado y para afirmar el flujo de bits codificado al sistema de entrega 31.

El sistema de entrega 31 está acoplado y configurado para entregar (por ejemplo, almacenando y/o transmitiendo) el flujo de bits codificado al decodificador 32. En algunas realizaciones, el sistema 31 implementa la entrega de (por ejemplo, transmite) un programa de audio multicanal codificado por un sistema de difusión o una red (por ejemplo, internet) al decodificador 32. En algunas realizaciones, el sistema 31 almacena un programa de audio multicanal codificado en un medio de almacenamiento (por ejemplo, un disco o conjunto de discos), y el decodificador 32 está configurado para leer el programa del medio de almacenamiento.

El bloque denominado "Asign Ca Inv 1" en el codificador 30 está configurado para llevar a cabo la permutación de canales (equivalente a la multiplicación por una matriz de permutación) en los canales del programa de entrada. Los canales permutados luego experimentan la codificación en la etapa 33, que emite 8 canales de señal codificados. Los canales de señal codificados pueden (pero no necesitan) corresponder a los canales de altavoz de reproducción. Los canales de señal codificados a veces se denominan canales "internos" dado que un decodificador (y/o sistema de renderización) típicamente decodifica y renderiza el contenido de los canales de señal codificados para recuperar el audio de entrada, de modo que los canales de señal codificados sean "internos" al sistema de codificación/decodificación. La codificación llevada a cabo en la etapa 33 es equivalente a la multiplicación de cada conjunto de muestras de los canales permutados por una matriz de codificación (implementada como una cascada

de multiplicaciones de matriz de $n+1$, identificada como $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$, que se va a describir más adelante en mayor detalle).

El subsistema de determinación de matrices 34 está configurado para generar datos indicativos de los coeficientes de dos conjuntos de matrices de salida (un conjunto correspondiente a cada uno de los 2 subflujos de los canales codificados). Un conjunto de matrices de salida consta de dos matrices, P_0^2, P_1^2 , cada una de las cuales es una matriz primitiva (definida más abajo) de dimensiones 2×2 , y es para la renderización de un primer subflujo (un subflujo de mezcla descendente) que comprende 2 de los canales de audio codificados del flujo de bits codificado (para renderizar una mezcla descendente de 2 canales del audio de entrada de 8 canales). El otro conjunto de matrices de salida consta de las matrices de renderización, P_0, P_1, \dots, P_n , cada una de las cuales es una matriz primitiva, y es para la renderización de un segundo subflujo que comprende los 8 canales de audio codificados del flujo de bits codificado (para la recuperación sin pérdidas del programa de audio de entrada de 8 canales). Una

cascada de las matrices, P_0^2, P_1^2 , junto con las matrices $P_0^{-1}, P_1^{-1}, \dots, P_n^{-1}$, aplicada al audio en el codificador, es igual a la especificación de matriz de mezcla descendente que transforma los 8 canales de audio de entrada en la mezcla descendente de 2 canales, y una cascada de las matrices, P_0, P_1, \dots, P_n , renderiza los 8 canales codificados del flujo de bits codificado de vuelta a los 8 canales de entrada originales.

Los coeficientes (de cada una de las matrices) que se emiten desde el subsistema 34 al subsistema de empaquetado 35 son metadatos que indican la ganancia relativa o absoluta de cada canal que se va a incluir en una mezcla correspondiente de canales del programa. Los coeficientes de cada matriz de renderización (para un instante de tiempo durante el programa) representan cuánto debería contribuir cada uno de los canales de una mezcla a la mezcla de contenido de audio (en el instante correspondiente de la mezcla renderizada) indicada por la señal de altavoz para un altavoz de sistema de reproducción particular.

Los 8 canales de audio codificados (emitidos desde la etapa de codificación 33), los coeficientes de matriz de salida (generados por el subsistema 34) y típicamente también los datos adicionales se afirman al subsistema de empaquetado 35, que los reúne en el flujo de bits codificado que luego se afirma al sistema de entrega 31.

El flujo de bits codificado incluye los datos indicativos de los 8 canales de audio codificados, los 2 conjuntos de matrices de salida (un conjunto correspondiente a cada uno de los 2 subflujos de los canales codificados) y típicamente también los datos adicionales (por ejemplo, metadatos relativos al contenido de audio).

El subsistema de análisis sintáctico 36 del decodificador 32 está configurado para aceptar (leer o recibir) el flujo de bits codificado del sistema de entrega 31 y para analizar sintácticamente el flujo de bits codificado. El subsistema 36 es operativo para afirmar los subflujos del flujo de bits codificado, que incluye un "primer" subflujo que comprende

solo 2 de los canales codificados del flujo de bits codificado, y las matrices de salida (P_0^2, P_1^2) correspondientes al primer subflujo, a la etapa de multiplicación de matriz 38 (para el procesamiento que da lugar a una presentación de mezcla descendente de 2 canales de contenido del programa de entrada de 8 canales original). El subsistema 36 es también operativo para afirmar los subflujos del flujo de bits codificado (el "segundo" subflujo que comprende los 8 canales codificados del flujo de bits codificado) y las matrices de salida correspondientes (P_0, P_1, \dots, P_n) a la etapa de multiplicación de matriz 37 para el procesamiento que da lugar a la renderización sin pérdidas el programa de 8 canales original.

Más específicamente, la etapa 38 multiplica 2 muestras de audio de los 2 canales del primer subflujo por una cascada de las matrices P_0^2, P_1^2 , y cada conjunto resultante de 2 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 0" para producir cada par de muestras de la mezcla descendente de 2 canales requerida de los 8 canales de audio originales. La cascada de operaciones de matriz llevadas a cabo en el codificador 30 y el decodificador 32 es equivalente a la aplicación de una especificación de matriz de mezcla descendente que transforma los 8 canales de audio de entrada en la mezcla descendente de 2 canales.

La etapa 37 multiplica cada vector de 8 muestras de audio (una de cada del conjunto completo de 8 canales del flujo de bits codificado) por una cascada de las matrices P_0, P_1, \dots, P_n , y cada conjunto resultante de 8 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 1" para producir cada conjunto de 8 muestras del programa de 8 canales original recuperado sin pérdidas. Con el fin de que el audio de 8 canales de salida sea exactamente igual que el audio de 8 canales de entrada (para conseguir la característica "sin pérdidas" del sistema), las operaciones de matrización llevadas a cabo en el codificador 30 deberían ser exactamente (incluidos los efectos de cuantificación) las inversas de las operaciones de matrización llevadas a cabo en el decodificador 32 en el (segundo) subflujo sin pérdidas del flujo de bits codificado (es decir, la multiplicación por la cascada de las matrices P_0, P_1, \dots, P_n). De este modo, en la Figura 1, las operaciones de matrización en la etapa 33 del codificador 30 se identifican como una cascada de las matrices inversas de las matrices P_0, P_1, \dots, P_n , en la secuencia inversa

aplicada en la etapa 37 del decodificador 32, concretamente: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

El decodificador 32 aplica la inversa de la permutación de canales aplicada por el codificador 30 (es decir, la matriz de permutación representada por el elemento "Asign Ca 1" del decodificador 32 es la inversa de aquella representada por el elemento "Asign Ca Inv 1" del codificador 30).

Dada una especificación de matriz de mezcla descendente (por ejemplo, la especificación de una matriz estática A que tiene las dimensiones de 2×8), un objetivo de una implementación de codificador TrueHD convencional del codificador 30 consiste en diseñar matrices de salida (por ejemplo, P_0, P_1, \dots, P_n y P_0^2, P_1^2 de la Figura 1) y matrices de entrada $(P_n^{-1}, \dots, P_1^{-1}, P_0^{-1})$ y asignaciones de canales de salida (y entrada) de modo que:

1. el flujo de bits codificado sea jerárquico (es decir, en el ejemplo, los primeros 2 canales codificados sean suficientes para derivar la presentación de mezcla descendente de 2 canales, y el conjunto completo de 8 canales codificados es suficiente para recuperar el programa de 8 canales original); y
 2. las matrices para el flujo superior (P_0, P_1, \dots, P_n en el ejemplo) son exactamente invertibles de modo que el audio de entrada sea exactamente recuperable por el decodificador.
- Los sistemas informáticos típicos funcionan con precisión finita y la inversión de una matriz invertible arbitraria exactamente podría requerir una precisión muy alta. El TrueHD resuelve este problema obligando a las matrices de salida y a las matrices de entrada (es decir, P_0, P_1, \dots, P_n y $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$) a ser matrices cuadradas del tipo conocido como "matrices primitivas".

Una matriz primitiva P de dimensiones $N \times N$ es de la forma:

$$P = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \ddots \\ \alpha_0 & \alpha_1 & \alpha_2 & \ddots & \alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Una matriz primitiva es siempre una matriz cuadrada. Una matriz primitiva de dimensiones $N \times N$ es idéntica a la matriz de identidad de dimensiones $N \times N$ a excepción de una hilera (no trivial) (es decir, la hilera que comprende los elementos $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ en el ejemplo). En todas las demás hileras, los elementos fuera de la diagonal son ceros y el elemento compartido con la diagonal tiene un valor absoluto de 1 (es decir, +1 o -1). Para simplificar el lenguaje en esta descripción, los dibujos y las descripciones siempre supondrán que una matriz primitiva tiene elementos diagonales que son iguales a +1 con la posible excepción del elemento diagonal en la hilera no trivial. Sin embargo, señalamos que esto es sin pérdida de generalidad, y las ideas presentadas en esta descripción se refieren a la clase general de matrices primitivas en las que los elementos diagonales pueden ser +1 o -1.

Cuando una matriz primitiva, P , opera en (es decir, multiplica) un vector $x(t)$, el resultado es el producto $Px(t)$, que es otro vector N dimensional que es exactamente igual que $x(t)$ en todos los elementos excepto uno. De este modo, cada matriz primitiva se puede asociar con un canal único que manipula (o en el que opera).

Utilizaremos el término "matriz primitiva unitaria" en la presente memoria para designar una matriz primitiva en la que el elemento compartido con la diagonal (por la hilera no trivial de la matriz primitiva) tiene un valor absoluto de 1 (es decir, o +1 o -1). De este modo, la diagonal de una matriz primitiva unitaria consta de todos unos positivos, +1 o todos unos negativos, -1, o algunos unos positivos y algunos unos negativos. Una matriz primitiva solo altera un canal de un conjunto (vector) de muestras de canales de programa de audio, y una matriz primitiva unitaria es

también invertible sin pérdidas debido a los valores unitarios en la diagonal. De nuevo, para simplificar la exposición en la presente memoria, utilizaremos el término "matriz primitiva unitaria" para hacer referencia a una matriz primitiva cuya hilera no trivial tiene un elemento diagonal de +1. Sin embargo, todas las referencias a las matrices primitivas unitarias en la presente memoria, incluidas en las reivindicaciones, tienen por objeto cubrir el caso más genérico en el que una matriz primitiva unitaria puede tener una hilera no trivial cuyo elemento compartido con la diagonal es +1 o -1.

Si $\alpha_2 = 1$ (dando lugar a una matriz primitiva unitaria que tiene una diagonal que consta de unos positivos) en el ejemplo anterior de la matriz primitiva, P, se ve que la inversa de P es exactamente:

$$P^{-1} = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ 0 & 1 & 0 & \ddots & \ddots \\ -\alpha_0 & -\alpha_1 & 1 & \ddots & -\alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Es cierto en general que la inversa de una matriz primitiva unitaria se determina simplemente invirtiendo (multiplicando por -1) cada uno de sus coeficientes α no triviales que no se encuentra a lo largo de la diagonal.

Si las matrices P_0, P_1, \dots, P_n empleadas en el decodificador 32 de la Figura 1 son matrices primitivas unitarias (que

tienen diagonales unitarias), la secuencia de operaciones de matrización $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$ en el codificador 30 y P_0, P_1, \dots, P_n en el decodificador 32 se puede implementar mediante circuitos de precisión finita del tipo mostrado en las Figuras 2A y 2B. La Figura 2A es una circuitería convencional de un codificador para llevar a cabo una matrización sin pérdidas mediante matrices primitivas implementadas con aritmética de precisión finita. La Figura 2B es una circuitería convencional de un decodificador para llevar a cabo una matrización sin pérdidas mediante matrices primitivas implementadas con aritmética de precisión finita. Los detalles de las implementaciones típicas de la circuitería de la Figura 2A y la Figura 2B (y variaciones a las mismas) se describen en la Patente de EE. UU. 6,611,212 citada anteriormente, expedida el 26 de agosto de 2003.

En la Figura 2A (que representa la circuitería para la codificación de un programa de audio de 4 canales que comprende los canales S1, S2, S3 y S4), una primera matriz primitiva P_0^{-1} (que tiene una hilera de 4 coeficientes α distintos de cero) opera en cada muestra del canal S1 (para generar el canal codificado S1') mezclando la muestra pertinente del canal S1 con las muestras correspondientes (que tienen lugar en el mismo tiempo, t) de los canales S2, S3 y S4. Una segunda matriz primitiva P_1^{-1} (que también tiene una hilera de 4 coeficientes α distintos de cero) opera en cada muestra del canal S2 (para generar una muestra correspondiente del canal codificado S2') mezclando la muestra pertinente del canal S2 con las muestras correspondientes de los canales S1', S3 y S4. Más específicamente, la muestra del canal S2 se multiplica por el inverso de un coeficiente α_1 (identificado como "coef[1,2]") de la matriz P_0^{-1} , la muestra del canal S3 se multiplica por el inverso de un coeficiente α_2 (identificado como "coef[1,3]") de la matriz P_0^{-1} , y la muestra del canal S4 se multiplica por el inverso de un coeficiente α_3 (identificado como "coef[1,4]") de la matriz P_0^{-1} , los productos se suman y después se cuantifican, y la suma cuantificada se resta entonces de la muestra correspondiente del canal S 1. De forma similar, la muestra del canal S 1 se multiplica por el inverso de un coeficiente α_0 (identificado como "coef[2,1]") de la matriz P_1^{-1} , la muestra del canal S3 se multiplica por el inverso de un coeficiente α_2 (identificado como "coef[2,3]") de la matriz P_1^{-1} , y la muestra del canal S4 se multiplica por el inverso de un coeficiente α_3 (identificado como "coef[2,4]") de la matriz P_1^{-1} , los productos se suman y después se cuantifican, y la suma cuantificada se resta entonces de la muestra correspondiente del canal S2. La etapa de cuantificación Q1 de la matriz P_0^{-1} cuantifica la salida del elemento de suma que suma los productos de las multiplicaciones (por coeficientes α distintos de cero de la matriz P_0^{-1} , que son valores típicamente fraccionarios) para generar el valor cuantificado que se resta de la muestra del canal S 1 para generar la muestra correspondiente del canal codificado S1'. La etapa de cuantificación Q2 de la matriz P_1^{-1} cuantifica la salida del elemento de suma que suma los productos de las multiplicaciones (por coeficientes α distintos de cero de la matriz P_1^{-1} , que son valores típicamente fraccionarios) para generar el valor cuantificado que se resta de la muestra del canal S2 para generar la muestra correspondiente del canal codificado S2'. En una implementación típica (por ejemplo, para llevar a cabo la codificación de TrueHD), cada muestra de cada uno de los canales S1, S2, S3 y S4 comprende 24 bits (como se indica en la Figura 2A), y la salida de cada elemento de multiplicación comprende 38 bits (como también se indica en la Figura 2A), y cada una de las etapas de cuantificación Q1 y Q2 emite un valor cuantificado de 24 bits en respuesta a cada valor de 38 bits que se introduce en las mismas.

Naturalmente, para codificar los canales S3 y S4, 2 matrices primitivas adicionales podrían estar dispuestas en cascada con las 2 matrices primitivas (P_0^{-1} y P_1^{-1}) indicadas en la Figura 2A.

En la Figura 2B (que representa la circuitería para la decodificación del programa codificado de 4 canales generado por el codificador de la Figura 2A), una matriz primitiva P_1 (que tiene una hilera de 4 coeficientes α distintos de cero, y que es la inversa de la matriz P_1^{-1}) opera en cada muestra del canal codificado $S2'$ (para generar una muestra correspondiente del canal decodificado $S2$) mezclando muestras de los canales $S1'$, $S3$ y $S4$ con la muestra pertinente del canal $S2'$. Una segunda matriz primitiva P_0 (que también tiene una hilera de 4 coeficientes α distintos de cero y que es la inversa de la matriz P_0^{-1}) opera en cada muestra del canal codificado $S1'$ (para generar una muestra correspondiente del canal decodificado $S1$) mezclando las muestras de los canales $S2$, $S3$ y $S4$ con la muestra pertinente del canal $S1'$. Más específicamente, la muestra del canal $S1'$ se multiplica por un coeficiente α_0 (identificado como "coef[2,1]") de la matriz P_1 , la muestra del canal $S3$ se multiplica por un coeficiente α_2 (identificado como "coef[2,3]") de la matriz P_1 , y la muestra del canal $S4$ se multiplica por un coeficiente α_3 (identificado como "coef[2,4]") de la matriz P_1 , los productos se suman y después se cuantifican, y la suma cuantificada se añade entonces a la muestra correspondiente del canal $S1'$. De forma similar, la muestra del canal $S2'$ se multiplica por un coeficiente α_1 (identificado como "coef[1,2]") de la matriz P_0 , la muestra del canal $S3$ se multiplica por un coeficiente α_2 (identificado como "coef[1,3]") de la matriz P_0 , y la muestra del canal $S4$ se multiplica por un coeficiente α_3 (identificado como "coef[1,4]") de la matriz P_0 , los productos se suman y después se cuantifican, y la suma cuantificada se añade entonces a la muestra correspondiente del canal $S1'$. La etapa de cuantificación $Q2$ de la matriz P_1 cuantifica la salida del elemento de suma que suma los productos de las multiplicaciones (por coeficientes α distintos de cero de la matriz P_1 , que son valores típicamente fraccionarios) para generar el valor cuantificado que se suma a la muestra del canal $S2'$ para generar la muestra correspondiente del canal $S2$ decodificado. La etapa de cuantificación $Q1$ de la matriz P_0 cuantifica la salida del elemento de suma que suma los productos de las multiplicaciones (por coeficientes α distintos de cero de la matriz P_0 , que son valores típicamente fraccionarios) para generar el valor cuantificado que se suma a la muestra del canal $S1'$ para generar la muestra correspondiente del canal $S1$ decodificado. En una implementación típica (por ejemplo, para llevar a cabo la decodificación de TrueHD), cada muestra de cada uno de los canales $S1'$, $S2'$, $S3$ y $S4$ comprende 24 bits (como se indica en la Figura 2B), y la salida de cada elemento de multiplicación comprende 38 bits (como también se indica en la Figura 2B), y cada una de las etapas de cuantificación $Q1$ y $Q2$ emite un valor cuantificado de 24 bits en respuesta a cada valor de 38 bits que se introduce en las mismas.

Naturalmente, para decodificar los canales $S3$ y $S4$, 2 matrices primitivas adicionales podrían estar dispuestas en cascada con las 2 matrices primitivas (P_0 y P_1) indicadas en la Figura 2B.

Una secuencia de matrices primitivas, por ejemplo, la secuencia de matrices de $N \times N$ primitivas P_0, P_1, \dots, P_n implementada por el decodificador de la Figura 1, que opera en un vector (N muestras, cada una de las cuales es una muestra de un canal diferente de un primer conjunto de N canales) puede implementar cualquier transformación lineal de las N muestras en un nuevo conjunto de N muestras (por ejemplo, puede implementar la transformación lineal llevada a cabo en un tiempo t multiplicando las muestras de N canales de un programa de audio basado en objetos por cualquier implementación de $N \times N$ de la matriz $A(t)$ de la ecuación (1) durante la renderización de los canales en N señales de altavoz, en la que la transformación se consigue manipulando un canal cada vez). De este modo, la multiplicación de un conjunto de N muestras de audio por una secuencia de matrices primitivas de $N \times N$ representa un conjunto genérico de escenarios en los que el conjunto de N muestras se convierten en otro conjunto (de N muestras) mediante operaciones lineales.

Con referencia de nuevo a una implementación del TrueHD del decodificador 32 de la Figura 1, con el fin de mantener la uniformidad de la arquitectura de decodificador en el TrueHD, las matrices de salida del subflujo de mezcla descendente (P_0^2, P_1^2 en la Figura 1) se implementan también como matrices primitivas aunque no necesitan ser invertibles (o tener una diagonal unitaria) dado que no están asociadas con la obtención de la característica sin pérdidas.

Las matrices primitivas de entrada y salida empleadas en un codificador TrueHD y un decodificador dependen de cada especificación de mezcla descendente particular que se va a implementar. La función de un decodificador TrueHD consiste en aplicar la cascada adecuada de matrices primitivas al flujo de bits de audio codificado recibido. De este modo, el decodificador TrueHD de la Figura 1 decodifica los 8 canales del flujo de bits codificado (entregado por el sistema D), y genera una mezcla descendente de 2 canales aplicando una cascada de 2 matrices primitivas de salida P_0^2, P_1^2 a un subconjunto de los canales del flujo de bits decodificado. Una implementación del TrueHD del decodificador 32 de la Figura 1 también es operativa para decodificar los 8 canales del flujo de bits codificado (entregado por el sistema D) para recuperar sin pérdidas el programa de 8 canales original aplicando una cascada de 8 matrices primitivas de salida P_0, P_1, \dots, P_n a los canales del flujo de bits codificado.

Un decodificador TrueHD no tiene el audio original (que se introdujo al codificador) para cotejar para determinar si su reproducción es sin pérdidas (o de cualquier otra manera que se desee por el codificador en el caso de una mezcla descendente). Sin embargo, el flujo de bits codificado contiene una "palabra de comprobación" (o comprobación sin pérdidas) que se compara con una palabra similar derivada en el decodificador a partir del audio reproducido para determinar si la reproducción es fiel.

Si un programa de audio basado en objetos (por ejemplo, que comprende más de 8 canales) fuera codificado por un codificador TrueHD convencional, el codificador podría generar subflujos de mezcla descendente que llevan presentaciones compatibles con dispositivos de reproducción heredados (por ejemplo, presentaciones que se podrían decodificar a señales de altavoz mezcladas de forma descendente para su reproducción en un canal 7.1 o un canal 5.1 tradicionales u otra configuración de altavoz tradicional) y un subflujo superior (indicativo de todos los canales del programa de entrada). Un decodificador TrueHD podría recuperar el programa de audio basado en objetos original sin pérdidas para su renderización por un sistema de reproducción. Cada especificación de matriz de renderización empleada por el codificador en este caso (es decir, para la generación del subflujo superior y de cada subflujo de mezcla descendente), y de este modo cada matriz de salida determinada por el codificador, podría ser una matriz de renderización variable en el tiempo, $A(t)$, que transforma linealmente las muestras de los canales del programa (por ejemplo, para generar una mezcla descendente de canal 7.1 o canal 5.1). Sin embargo, dicha matriz $A(t)$ variaría típicamente de forma rápida en el tiempo a medida que los objetos se mueven en la escena espacial, y la velocidad de bits y las limitaciones de procesamiento de un sistema TrueHD convencional (u otro sistema de decodificación convencional) obligaría típicamente al sistema a ser capaz de acomodar como máximo de acomodar una aproximación de constante a intervalos a tal especificación de matriz continuamente (y rápidamente) variable (con una tasa de actualización de matriz más alta conseguida a costa de una velocidad de bits aumentada para la transmisión del programa codificado). Con el fin de admitir la renderización de programas de audio multicanal basados en objetos (y otros programas de audio multicanal) con señales de altavoz indicativas de una mezcla rápidamente variable del contenido de los canales de los programas, los inventores han reconocido que es conveniente mejorar los sistemas convencionales para acomodar la matriz interpolada, en la que las actualizaciones de matriz de renderización son infrecuentes y se especifica una trayectoria deseada (es decir, una secuencia deseada de las mezclas del contenido de los canales del programa) entre las actualizaciones paramétricamente.

Breve descripción de la invención

En una clase de realizaciones, la invención es un método según la reivindicación 1.

En algunas realizaciones, el método incluye una etapa de generación de contenido de audio codificado mediante la realización de operaciones de matriz en las muestras de los N canales del programa (por ejemplo, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en las que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y la secuencia de cascadas de matrices incluye una primera cascada de matrices inversas que es una cascada de las inversas de las matrices primitivas de la primera cascada).

En algunas realizaciones, cada una de las matrices primitivas es una matriz primitiva unitaria. En algunas realizaciones en las que $N = M$, el método también incluye una etapa de recuperación sin pérdidas de los N canales del programa mediante el procesamiento del flujo de bits codificado, incluida mediante la realización de la interpolación para determinar la secuencia de cascadas de matrices primitivas actualizadas de $N \times N$, a partir de los valores de interpolación, la primera cascada de matrices primitivas, y la función de interpolación. El flujo de bits codificado puede ser indicativo de (es decir, puede incluir datos indicativos de) la función de interpolación, o la función de interpolación se puede proporcionar de otro modo al decodificador.

En algunas realizaciones en las que $N = M$, el método también incluye las etapas de: entrega del flujo de bits codificado a un decodificador configurado para implementar la función de interpolación, y el procesamiento del flujo de bits codificado en el decodificador para recuperar sin pérdidas los N canales del programa, incluida mediante la realización de la interpolación para determinar la secuencia de cascadas de matrices primitivas actualizadas de $N \times N$, a partir de los valores de interpolación, la primera cascada de matrices primitivas, y la función de interpolación.

En algunas realizaciones, el programa es un programa de audio basado en objetos que incluye al menos un canal de objeto y datos de posición indicativos de una trayectoria de al menos un objeto. La mezcla variable en el tiempo, $A(t)$, puede ser determinada a partir de los datos de posición (o a partir de datos que incluyen los datos de posición).

En algunas realizaciones, la primera cascada de matrices primitivas es una matriz primitiva semilla, y los valores de interpolación son indicativos de una matriz delta semilla para la matriz primitiva semilla.

En algunas realizaciones, una mezcla descendente variable en el tiempo, $A_2(t)$, de contenido de audio o contenido codificado del programa a M1 canales de altavoz también ha sido especificada durante el intervalo de tiempo, en el que M1 es un número entero menor que M, y el método incluye las etapas de:

determinación de una segunda cascada de matrices primitivas de $M1 \times M1$ que, cuando se aplica a las muestras de M1 canales del contenido de audio o contenido codificado, implementa una mezcla descendente de contenido de audio del programa a los M1 canales de altavoz, en la que la mezcla descendente es acorde con la mezcla variable en el tiempo, $A_2(t)$, en el sentido de que la mezcla descendente es al menos sustancialmente igual a $A_2(t)$;

determinación de valores de interpolación adicionales que, con la segunda cascada de matrices primitivas de $M1 \times M1$ y una segunda función de interpolación definida durante el subintervalo, son indicativos de una secuencia de

5 cascadas de matrices primitivas de $M1 \times M1$ actualizadas, de tal manera que cada una de las cascadas de matrices primitivas de $M1 \times M1$ actualizadas, cuando se aplican a las muestras de los $M1$ canales del contenido de audio o el contenido codificado, implementa una mezcla descendente actualizada, asociada con un tiempo diferente en el subintervalo, de contenido de audio del programa a los $M1$ canales de altavoz, en los que cada una de dichas mezclas descendentes actualizadas es acorde con la mezcla variable en el tiempo, $A_2(t)$, y en los que el flujo de bits codificado es indicativo de los valores de interpolación adicionales y la segunda cascada de matrices primitivas de $M1 \times M1$. El flujo de bits codificado puede ser indicativo de (es decir, puede incluir datos indicativos de) la segunda función de interpolación, o la segunda función de interpolación se puede proporcionar de otro modo al decodificador. La mezcla descendente variable en el tiempo, $A_2(t)$, es una mezcla descendente de contenido de audio o contenido codificado del programa en el sentido de que es una mezcla descendente de contenido de audio del programa original, o del contenido de audio codificado del flujo de bits codificado, o de una versión parcialmente decodificada del contenido de audio codificado del flujo de bits codificado, o de audio codificado de otra manera (por ejemplo, decodificado parcialmente) indicativo de contenido de audio del programa. La variación de tiempo en la especificación de mezcla descendente $A_2(t)$ puede ser debida (al menos en parte) al aumento de o la liberación de la protección de recorte de la mezcla descendente especificada.

En una segunda clase de realizaciones, la invención es un método según la reivindicación 5.

20 En algunas realizaciones, el contenido de audio codificado ha sido generado llevando a cabo operaciones de matriz en muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en las que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y la secuencia de cascadas de matrices incluye una primera cascada de matrices inversas que es una cascada de las inversas de las matrices primitivas de la primera cascada.

25 Los canales del programa de audio que se recuperan (por ejemplo, se recuperan sin pérdidas) de acuerdo con estas realizaciones a partir del flujo de bits codificado pueden ser una mezcla descendente de contenido de audio de un programa de audio de entrada de X canales (en el que X es un número entero cualquiera y N es menor que X) que se ha generado a partir del programa de audio de entrada de X canales mediante la realización de operaciones de matriz en el programa de audio de entrada de X canales, determinando de esta manera el contenido de audio codificado del flujo de bits codificado.

En algunas realizaciones en la segunda clase, cada una de las matrices primitivas es una matriz primitiva unitaria.

30 En algunas realizaciones en la segunda clase, una mezcla descendente variable en el tiempo, $A_2(t)$, del programa de N canal a $M1$ canales de altavoz ha sido especificada durante el intervalo de tiempo, y una mezcla descendente variable en el tiempo, $A_2(t)$, de contenido de audio o contenido codificado del programa a M canales de altavoz también ha sido especificada durante el intervalo de tiempo. El método incluye las etapas de:

recepción de una segunda cascada de matrices primitivas de $M1 \times M1$ y un segundo conjunto de valores de interpolación;

35 aplicación de la segunda cascada de matrices primitivas de $M1 \times M1$ a las muestras de $M1$ canales del contenido de audio codificado para implementar una mezcla descendente del programa de N canales a $M1$ canales de altavoz, en los que la mezcla descendente es acorde con la mezcla variable en el tiempo, $A_2(t)$, en el sentido de que la mezcla descendente es al menos sustancialmente igual a $A_2(t)$;

40 aplicación del segundo conjunto de valores de interpolación, la segunda cascada de matrices primitivas de $M1 \times M1$ y una segunda función de interpolación definida durante el subintervalo para obtener una secuencia de cascadas de matrices primitivas de $M1 \times M1$ actualizadas; y

45 aplicación de las matrices primitivas de $M1 \times M1$ actualizadas a las muestras de los $M1$ canales del contenido codificado para implementar al menos una mezcla descendente actualizada del programa de N canales, asociada con un tiempo diferente en el subintervalo, en la que cada una de dichas mezclas descendentes actualizada es acorde con la mezcla variable en el tiempo, $A_2(t)$.

50 En algunas realizaciones, la invención es un método para la renderización de un programa de audio multicanal, que incluye las etapas de suministro de un conjunto de matrices semilla (por ejemplo, una única matriz semilla o un conjunto de al menos 2 matrices semilla, correspondientes a un tiempo durante el programa de audio) a un decodificador, y de realización de la interpolación en el conjunto de matrices semilla (que está asociado con un tiempo durante el programa de audio) para determinar un conjunto de matrices de renderización interpoladas (una única matriz de renderización interpolada o un conjunto de al menos 2 matrices de renderización interpoladas, correspondientes a un tiempo posterior durante el programa de audio) para su utilización en la renderización de los canales del programa.

55 En algunas realizaciones, una matriz primitiva semilla y una matriz delta semilla (o un conjunto de matrices primitivas semilla y matrices delta semilla) son entregadas de vez en cuando (por ejemplo, de manera infrecuente) al

5 decodificador. El decodificador actualiza cada matriz primitiva semilla (correspondiente a un tiempo, t_1) generando una matriz primitiva interpolada (para un tiempo, t , posterior a t_1) de acuerdo con una realización de la invención a partir de la matriz primitiva semilla y una matriz delta semilla correspondiente, y una función de interpolación $f(t)$. Los datos indicativos de la función de interpolación pueden ser entregados con las matrices semilla o la función de interpolación puede ser predeterminada (es decir, conocida de antemano tanto por el codificador como por el decodificador). Alternativamente, una matriz primitiva semilla (o un conjunto de matrices primitivas semilla) se entrega de vez en cuando (por ejemplo, de manera infrecuente) al decodificador. El decodificador actualiza cada matriz primitiva semilla (correspondiente a un tiempo, t_1) generando una matriz primitiva interpolada (para un tiempo, t , posterior a t_1) de acuerdo con una realización de la invención a partir de la matriz primitiva semilla y una función de interpolación $f(t)$, es decir, no necesariamente utilizando una matriz delta semilla que corresponde a la matriz primitiva semilla. Los datos indicativos de la función de interpolación pueden ser entregados con la matriz (o matrices) primitiva(s) semilla o la función puede ser predeterminada (es decir, conocida de antemano tanto por el codificador como por el decodificador).

15 En realizaciones típicas, cada matriz primitiva es una matriz primitiva unitaria. En este caso, la inversa de la matriz primitiva se determina simplemente invirtiendo (multiplicando por -1) cada uno de sus coeficientes no triviales (cada uno de sus coeficientes α). Esto hace posible que las inversas de las matrices primitivas (que son aplicadas por el codificador para codificar el flujo de bits) se determinen de manera más eficiente, y permite la utilización del procesamiento de precisión finita (por ejemplo, circuitos de precisión finita) para implementar las multiplicaciones de matriz requeridas en el codificador y el decodificador.

20 Los aspectos de la invención incluyen un sistema o dispositivo (por ejemplo, un codificador o un decodificador) configurado (por ejemplo, programado) para implementar cualquier realización del método inventivo, un sistema o dispositivo que incluye una memoria intermedia que almacena (por ejemplo, de una manera no transitoria) al menos una trama u otro segmento de un programa de audio codificado generado por cualquier realización del método inventivo o las etapas del mismo, y un medio legible por ordenador (por ejemplo, un disco) que almacena el código (por ejemplo, de una manera no transitoria) para la implementación de cualquier realización del método inventivo o las etapas del mismo. Por ejemplo, el sistema inventivo puede ser o incluir un procesador de fines generales, un procesador de señales digitales o un microprocesador programables, programados con software o firmware y/o configurados de otro modo para llevar a cabo cualquiera de una variedad de operaciones en los datos, incluida una realización del método inventivo o las etapas del mismo. Tal procesador para fines generales puede ser o incluir un sistema informático que incluye un dispositivo de entrada, una memoria y circuitería de procesamiento programada (y/o configurada de otro modo) para llevar a cabo una realización del método inventivo (o las etapas del mismo) en respuesta a los datos afirmados a los mismos.

Breve descripción de los dibujos

35 La Figura 1 es un diagrama de bloques de los elementos de un sistema convencional que incluye un codificador, un subsistema de entrega y un decodificador.

La Figura 2A es un diagrama de una circuitería de codificador convencional para la realización de operaciones de matrización sin pérdidas mediante matrices primitivas implementadas con aritmética de precisión finita.

La Figura 2B es un diagrama de una circuitería de decodificador convencional para la realización de operaciones de matrización sin pérdidas mediante matrices primitivas implementadas con aritmética de precisión finita.

40 La Figura 3 es un diagrama de bloques de la circuitería empleada en una realización de la invención para aplicar una matriz primitiva de 4×4 (implementada con aritmética de precisión finita) a 4 canales de un programa de audio. La matriz primitiva es una matriz primitiva semilla, una de cuyas hileras no triviales comprende los elementos α_0 , α_1 , α_2 y α_3 .

45 La Figura 4 es un diagrama de bloques de la circuitería empleada en una realización de la invención para aplicar una matriz primitiva de 3×3 (implementada con aritmética de precisión finita) a 3 canales de un programa de audio. La matriz primitiva es una matriz primitiva interpolada, generada a partir de una matriz primitiva semilla $P_k(t_1)$ una de cuyas hileras no triviales comprende los elementos α_0 , α_1 y α_2 , y una matriz delta semilla $\Delta_k(t_1)$ cuya hilera no trivial comprende los elementos δ_0 , δ_1 , ..., δ_{N-1} , y una función de interpolación $f(t)$.

50 La Figura 5 es un diagrama de bloques de una realización del sistema inventivo que incluye una realización del codificador inventivo, un subsistema de entrega, y una realización del codificador inventivo.

La Figura 6 es un diagrama de bloques de otra realización del sistema inventivo que incluye una realización del codificador inventivo, un subsistema de entrega, y una realización del decodificador inventivo.

La Figura 7 es un gráfico de la suma de los errores cuadrados entre una especificación obtenida y una especificación real en diferentes instantes del tiempo, t , mediante la utilización de matrices primitivas interpoladas (la

curva denominada "Matrización interpolada") y con matrices primitivas (no interpoladas) de constante a intervalos (la curva denominada "Matrización no interpolada").

Notación y nomenclatura

5 A lo largo de esta descripción, incluidas en las reivindicaciones, la expresión de llevar a cabo una operación "en" una señal o datos (por ejemplo, un filtrado, un ajuste a escala, una transformación o la aplicación de una ganancia a, la señal o los datos) se utiliza en un sentido amplio para designar la realización de la operación directamente en la señal o los datos, o en una versión procesada de la señal o los datos (por ejemplo, en una versión de la señal que ha experimentado un filtrado preliminar o un preprocesamiento antes de la realización de la operación en la misma).

10 A lo largo de esta descripción, incluidas en las reivindicaciones, la expresión "sistema" se utiliza en un sentido amplio para designar un dispositivo, un sistema o un subsistema. Por ejemplo, un subsistema que implementa un decodificador se puede denominar "sistema de decodificador", y un sistema que incluye dicho subsistema (por ejemplo, un sistema que genera Y señales de salida en respuesta a múltiples entradas, en las que el subsistema genera M de las entradas y las otras entradas de Y - M se reciben de una fuente externa) también se puede denominar "sistema de decodificador".

15 A lo largo de esta descripción, incluidas en las reivindicaciones, el término "procesador" se utiliza en un sentido amplio para designar un sistema o dispositivo programable o configurable de otro modo (por ejemplo, con software o firmware) para llevar a cabo operaciones en los datos (por ejemplo, audio o video u otros datos de imágenes). Ejemplos de procesadores incluyen una matriz de puertas programable por campo (u otro circuito integrado o conjunto de chips configurables), un procesador de señales digitales programado y/o configurado de otro modo para
20 llevar a cabo un procesamiento encauzado en el audio u otros datos de sonido, un procesador o un ordenador de fines generales programables y un chip de microprocesador o un conjunto de chips programables.

A lo largo de esta descripción, incluidas en las reivindicaciones, la expresión "metadatos" hace referencia a datos separados y diferentes de los datos de audio correspondientes (contenido de audio de un flujo de bits que también incluye metadatos). Los metadatos están asociados con datos de audio, e indican al menos una característica de los
25 datos de audio (por ejemplo, qué tipo(s) de procesamiento ya se ha(n) llevado a cabo, o debería(n) llevarse a cabo, en los datos de audio, o la trayectoria de un objeto indicada por los datos de audio). La asociación de los metadatos con los datos de audio es síncrona en el tiempo. De este modo, los metadatos actuales (recibidos o actualizados más recientemente) pueden indicar que los datos de audio correspondientes contemporáneamente tienen una característica indicada y/o comprende los resultados de un tipo indicado de procesamiento de datos de audio.

30 A lo largo de esta descripción, incluidas en las reivindicaciones, el término "se acopla" o "acoplado" se utiliza para hacer referencia a una conexión directa o indirecta. De este modo, si un primer dispositivo se acopla a un segundo dispositivo, dicha conexión puede ser mediante una conexión directa o mediante una conexión indirecta mediante otros dispositivos y conexiones.

A lo largo de esta descripción, incluidas en las reivindicaciones, las siguientes expresiones tienen las siguientes
35 definiciones:

altavoz: se utiliza para designar cualquier transductor emisor de sonido. Esta definición incluye los altavoces implementados como múltiples transductores (por ejemplo, el altavoz de graves y el altavoz de agudos);

señal de altavoz: una señal de audio que se va a aplicar directamente a un altavoz, o una señal de audio que se ha de aplicar a un amplificador y un altavoz en serie;

40 canal (o "canal de audio"): una señal de audio monofónica. Dicha señal se puede renderizar típicamente de tal manera que sea equivalente a la aplicación de la señal directamente a un altavoz en una posición deseada o nominal. La posición deseada puede ser estática, como es típicamente el caso con los altavoces físicos, o dinámica;

45 programa de audio: un conjunto de uno o más canales de audio (al menos un canal de altavoz y/o al menos un canal de objeto) y opcionalmente también los metadatos asociados (por ejemplo, metadatos que describen una presentación de audio espacial deseada);

50 canal de altavoz (o "canal de señal de altavoz"): un canal de audio que está asociado con un altavoz mencionado (en una posición deseada o nominal), o con una zona de altavoz mencionada dentro de una configuración de altavoz definida. Un canal de altavoz se renderiza de tal manera que sea equivalente a la aplicación de la señal de audio directamente al altavoz mencionado (en la posición deseada o nominal) o a un altavoz en la zona de altavoz mencionada;

canal de objeto: un canal de audio indicativo del sonido emitido por una fuente de audio (a veces denominado "objeto" de audio). Típicamente, un canal de objeto determina una descripción de fuente de audio paramétrica (por ejemplo, los metadatos indicativos de la descripción de fuente de audio paramétrica están incluidos en o

proporcionados con el canal de objeto). La descripción de fuente puede determinar el sonido emitido por la fuente (como una función del tiempo), la posición evidente (por ejemplo, coordenadas espaciales en 3D) de la fuente como una función del tiempo, y opcionalmente al menos un parámetro adicional (por ejemplo, tamaño o anchura evidentes de la fuente) que caracteriza la fuente; y

- 5 programa de audio basado en objetos: un programa de audio que comprende un conjunto de uno o más canales de objeto (y opcionalmente también que comprende al menos un canal de altavoz) y opcionalmente también metadatos asociados (por ejemplo, metadatos indicativos de una trayectoria de un objeto de audio que emite sonido indicado por un canal de objeto, o metadatos indicativos de otro modo de una presentación de audio espacial deseada de sonido indicado por un canal de objeto, o metadatos indicativos de una identificación de al menos un objeto de audio
10 que es una fuente del sonido indicado por un canal de objeto).

Descripción detallada de las realizaciones de la invención

Se describirán ejemplos de las realizaciones de la invención con referencia a las Figuras 3, 4, 5 y 6.

La Figura 5 es un diagrama de bloques de una realización del sistema de procesamiento de datos de audio inventivo que incluye el codificador 40 (una realización del codificador inventivo), el subsistema de entrega 41 (que puede ser idéntico al subsistema de entrega 31 de la Figura 1) y el decodificador 42 (una realización del decodificador inventivo), acoplados entre sí como se muestra. Aunque el subsistema 42 se denomina en la presente memoria "decodificador", se debería entender que puede ser implementado como un sistema de reproducción que incluye un subsistema de decodificación (configurado para analizar sintácticamente y decodificar un flujo de bits indicativo de un programa de audio multicanal codificado) y otros subsistemas configurados para implementar la renderización y al menos algunas etapas de la reproducción de la salida del subsistema de decodificación. Algunas realizaciones de la invención son decodificadores que no están configurados para llevar a cabo la renderización y/o la reproducción (y que se utilizarían típicamente con un sistema de renderización y/o reproducción separado). Algunas realizaciones de la invención son sistemas de reproducción (por ejemplo, un sistema de reproducción que incluye un subsistema de decodificación y otros subsistemas configurados para implementar la renderización y al menos algunas etapas de la reproducción de la salida del subsistema de decodificación.
15
20
25

En el sistema de la Figura 5, el codificador 40 está configurado para codificar un programa de audio de 8 canales (por ejemplo, un conjunto tradicional de señales de altavoz 7.1) como un flujo de bits codificado que incluye 2 subflujos, y el decodificador 42 está configurado para decodificar el flujo de bits codificado para renderizar o el programa de 8 canales original (sin pérdidas) o una mezcla descendente de 2 canales del programa de 8 canales original. El codificador 40 está acoplado y configurado para generar el flujo de bits codificado y para afirmar el flujo de bits codificado al sistema de entrega 41.
30

El sistema de entrega 41 está acoplado y configurado para entregar (por ejemplo, almacenando y/o transmitiendo) el flujo de bits codificado al decodificador 42. En algunas realizaciones, el sistema 41 implementa la entrega de (por ejemplo, transmite) un programa de audio multicanal codificado por un sistema de difusión o una red (por ejemplo, internet) al decodificador 42. En algunas realizaciones, el sistema 41 almacena un programa de audio multicanal codificado en un medio de almacenamiento (por ejemplo, un disco o un conjunto de discos), y el decodificador 42 está configurado para leer el programa del medio de almacenamiento.
35

El bloque denominado "Asign Ca Inv" 1 en el codificador 40 está configurado para llevar a cabo la permutación de canales (equivalente a la multiplicación por una matriz de permutación) en los canales del programa de entrada. Los canales permutados luego experimentan la codificación en la etapa 43, que emite 8 canales de señal codificados. Los canales de señal codificados pueden (pero no necesitan) corresponder a canales de altavoz de reproducción. Los canales de señal codificados se denominan a veces canales "internos" dado que un decodificador (y/o sistema de renderización) típicamente decodifica y renderiza el contenido de los canales de señal codificados para recuperar el audio de entrada, de modo que los canales de señal codificados sean "internos" al sistema de codificación/decodificación. La codificación llevada a cabo en la etapa 43 es equivalente a la multiplicación de cada conjunto de muestras de los canales permutados por una matriz de codificación, implementada como una cascada
40
45

de multiplicaciones de matriz, identificada como $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

Aunque n puede ser igual a 7 en la realización ejemplar, en la realización y en las variaciones de la misma el programa de audio de entrada comprende un número arbitrario (N o X) de canales, en los que N (o X) es cualquier número entero mayor que 1, y n en la Figura 5 puede ser n = N-1 (o n = X-1 u otro valor). En tales realizaciones alternativas, el codificador está configurado para codificar el programa de audio multicanal como un flujo de bits codificado que incluye cierto número de subflujos, y el decodificador está configurado para decodificar el flujo de bits codificado para renderizar o el programa multicanal original (sin pérdidas) o una o más mezclas descendentes del programa multicanal original. Por ejemplo, la etapa de codificación (correspondiente a la etapa 43) de dicha realización alternativa puede aplicar una cascada de matrices primitivas de NxN a las muestras de los canales del programa, para generar N canales de señal codificados que se pueden convertir en una primera mezcla de M
50
55

canales de salida, en la que la primera mezcla es acorde con una mezcla variable en el tiempo $A(t)$, especificada durante un intervalo, en el sentido de que la primera mezcla es al menos sustancialmente igual a $A(t_1)$, en el que t_1 es un tiempo en el intervalo. El decodificador puede crear los M canales de salida aplicando una cascada de matrices primitivas de $N \times N$ recibida como parte del contenido de audio codificado. El codificador en tal realización alternativa también puede generar una segunda cascada de matrices primitivas de $M_1 \times M_1$ (en la que M_1 es un número entero menor que N), que también se incluye en el contenido de audio codificado. Un decodificador puede aplicar la segunda cascada en M1 canales de señal codificados para implementar una mezcla descendente del programa de N canales a M1 canales de altavoz, en la que la mezcla descendente es acorde con otra mezcla variable en el tiempo, $A_2(t)$, en el sentido de que la mezcla descendente es al menos sustancialmente igual a $A_2(t_1)$. El codificador en dicha realización alternativa también generaría valores de interpolación (de acuerdo con cualquier realización de la presente invención) e incluiría los valores de interpolación en el flujo de bits codificado emitido desde el codificador, para su utilización por un decodificador para decodificar y renderizar el contenido del flujo de bits codificado de acuerdo con la mezcla variable en el tiempo, $A(t)$, y/o para decodificar y renderizar una mezcla descendente del contenido del flujo de bits codificado de acuerdo con la mezcla variable en el tiempo, $A_2(t)$.

La descripción de la Figura 5 a veces hará referencia a la entrada de señal multicanal al codificador inventivo como una señal de entrada de 8 canales para la especificidad, pero la descripción (con variaciones triviales evidentes para aquellos con experiencia ordinaria) también se aplica al caso general reemplazando las referencias a una señal de entrada de 8 canales con referencias a una señal de entrada de N canales, reemplazando las referencias a cascadas de matrices primitivas de 8 canales (o 2 canales) con referencias a las matrices primitivas de M canales (o M1 canales), y reemplazando las referencias a la recuperación sin pérdidas de una señal de entrada de 8 canales por las referencias a la recuperación sin pérdidas de una señal de audio de M canales (en la que la señal de audio de M canales se ha determinado llevando a cabo operaciones de matriz para aplicar una mezcla variable en el tiempo, $A(t)$, a una señal de audio de entrada de N canales para determinar M canales de señal codificados).

Con referencia a la etapa de codificador 43 de la Figura 5, cada matriz P_n^{-1} , ..., P_1^{-1} y P_0^{-1} (y de este modo la cascada aplicada por la etapa 43) se determina en el subsistema 44 y se actualiza de vez en cuando (típicamente de manera infrecuente) de acuerdo con una mezcla variable en el tiempo especificada de los N (donde $N = 8$) canales del programa a los N canales de señal codificados que se ha especificado durante el intervalo de tiempo.

El subsistema de determinación de matrices 44 está configurado para generar datos indicativos de los coeficientes de dos conjuntos de matrices de salida (un conjunto correspondiente a cada uno de los 2 subflujos de los canales codificados). Cada conjunto de matrices de salida se actualiza de vez en cuando, de modo que los coeficientes también se actualicen de vez en cuando. Un conjunto de matrices de salida consta de 2 matrices de renderización, $P_0^2(t)$, $P_1^2(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones 2×2 , y es para la renderización de un primer subflujo (un subflujo de mezcla descendente) que comprende 2 de los canales de audio codificados del flujo de bits codificado (para renderizar una mezcla descendente de 2 canales del audio de entrada de 8 canales). El otro conjunto de matrices de salida consta de 8 matrices de renderización, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones 8×8 , y es para la renderización de un segundo subflujo que comprende los 8 canales de audio codificados del flujo de bits codificado (para la recuperación sin pérdidas del programa de audio de entrada de 8 canales). Para cada tiempo, t, una cascada de las matrices de renderización, $P_0^2(t)$, $P_1^2(t)$, se puede interpretar como una matriz de renderización para los canales del primer subflujo que renderiza la mezcla descendente de 2 canales a partir de los 2 canales de señal codificados en el primer subflujo, y de forma similar una cascada de las matrices de renderización, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, se puede interpretar como una matriz de renderización para los canales del segundo subflujo.

Los coeficientes (de cada matriz de renderización) que se emiten desde el subsistema 44 al subsistema de empaquetado 45 son metadatos que indican la ganancia relativa o absoluta de cada canal que se va a incluir en una mezcla correspondiente de canales del programa. Los coeficientes de cada matriz de renderización (para un instante de tiempo durante el programa) representan cuánto debería contribuir cada uno de los canales de una mezcla a la mezcla del contenido de audio (en el instante correspondiente de la mezcla renderizada) indicado por la señal de altavoz para un altavoz de sistema de reproducción particular.

Los 8 canales de audio codificados (emitidos desde la etapa de codificación 43), los coeficientes de matriz de salida (generados por el subsistema 44), y típicamente también los datos adicionales se afirman al subsistema de empaquetado 45, que los reúne en el flujo de bits codificado que luego se afirma al sistema de entrega 41.

El flujo de bits codificado incluye datos indicativos de los 8 canales de audio codificados, los 2 conjuntos de matrices de salida variables en el tiempo (un conjunto correspondiente a cada uno de los dos subflujos de los canales codificados), y típicamente también los datos adicionales (por ejemplo, metadatos relativos al contenido de audio).

En funcionamiento, el codificador 40 (y las realizaciones alternativas del codificador inventivo, por ejemplo, el codificador 100 de la Figura 6) codifica un programa de audio de N canales cuyas muestras corresponden a un intervalo de tiempo, en el que el intervalo de tiempo incluye un subintervalo desde un tiempo t_1 hasta un tiempo t_2 .

Cuando una mezcla variable en el tiempo, $A(t)$, de N canales de señal codificados a M canales de salida ha sido especificada durante el intervalo de tiempo, el codificador lleva a cabo las etapas de:

- 5 determinación de una primera cascada de matrices primitivas de $N \times N$ (por ejemplo, las matrices $P_0(t_1)$, $P_1(t_1)$, ..., $P_n(t_1)$, para el tiempo t_1) que, cuando se aplica a las muestras de los N canales de señal codificados, implementa una primera mezcla del contenido de audio de los N canales de señal codificados a los M canales de salida, en la que la primera mezcla es acorde con la mezcla variable en el tiempo, $A(t)$, en el sentido de que la primera mezcla es al menos sustancialmente igual a $A(t_1)$;
- 10 generación de contenido de audio codificado (por ejemplo, la salida de la etapa 43 del codificador 40 o la salida de la etapa 103 del codificador 100) llevando a cabo las operaciones de matriz en las muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en las que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y la secuencia de cascadas de matrices incluye una primera cascada de matrices inversas que es una cascada de las inversas de las matrices primitivas de la primera cascada;
- 15 determinación de los valores de interpolación (por ejemplo, los valores de interpolación incluidos en la salida de la etapa 44 del codificador 40 o en la salida de la etapa 103 del codificador 100) que, con la primera cascada de matrices primitivas (por ejemplo, incluidas en la salida de la etapa 44 o la etapa 103) y una función de interpolación definida durante el subintervalo son indicativos de una secuencia de las cascadas de matrices primitivas actualizadas de $N \times N$, de tal manera que cada una de las cascadas de las matrices primitivas actualizadas, cuando se aplica a las muestras de los N canales de señal codificados, implementa una mezcla actualizada, asociada con un tiempo diferente en el subintervalo, de los N canales de señal codificados a los M canales de salida, en los que cada una de dichas mezclas actualizadas es acorde con la mezcla variable en el tiempo, $A(t)$. De forma preferible, pero no necesariamente (en todas las realizaciones), cada mezcla actualizada es acorde con la mezcla variable en el tiempo en el sentido de que la mezcla actualizada asociada con cualquier tiempo t_3 en el subintervalo es al menos sustancialmente igual a $A(t_3)$; y
- 20 generación de un flujo de bits codificado (por ejemplo, la salida de la etapa 45 del codificador 40 o la salida de la etapa 104 del codificador 100) que es indicativo del contenido de audio codificado, los valores de interpolación y la primera cascada de matrices primitivas.

Con referencia a la etapa 44 de la Figura 5, cada conjunto de matrices de salida (el conjunto P_0^2, P_1^2 , o el conjunto P_0, P_1, \dots, P_n) se actualiza de vez en cuando. El primer conjunto de matrices P_0^2, P_1^2 que se emite (en un primer momento, t_1) es una matriz semilla (implementada como una cascada de matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de 2 canales de la salida codificada de la etapa 43, correspondiente al primer tiempo). El primer conjunto de matrices P_0, P_1, \dots, P_n que se emite (en el primer tiempo, t_1) es también una matriz semilla (implementada como una cascada de matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de los 8 canales de la salida codificada de la etapa 43 correspondiente al primer tiempo). Cada conjunto actualizado de las matrices P_0^2, P_1^2 que se emite desde la etapa 44 es una matriz semilla actualizada (implementada como una cascada de matrices primitivas unitarias, que también se puede denominar una cascada de matrices primitivas semilla unitarias) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de los 2 canales de la salida codificada de la etapa 43, correspondientes al tiempo de actualización). Cada conjunto actualizado de las matrices P_0, P_1, \dots, P_n que se emite desde la etapa 43 es también una matriz semilla (implementada como una cascada de matrices primitivas unitarias, que también se puede denominar una cascada de las matrices primitivas semilla unitarias) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de los 8 canales de la salida codificada de la etapa 43 correspondiente al primer tiempo).

La etapa de salida 44 también emite los valores de interpolación, que (con una función de interpolación para cada matriz semilla) permiten al decodificador 42 generar las versiones interpoladas de las matrices semilla (correspondientes a los tiempos después del primer tiempo, t_1 , y entre los tiempos de actualización). Los valores de interpolación (que pueden incluir datos indicativos de cada función de interpolación) son incluidos por la etapa 45 en el flujo de bits codificado emitido desde el codificador 40. Describiremos ejemplos de dichos valores de interpolación más adelante (los valores de interpolación pueden incluir una matriz delta para cada matriz semilla).

Con referencia al decodificador 42 de la Figura 5, el subsistema de análisis sintáctico 46 (del decodificador 42) está configurado para aceptar (leer o recibir) el flujo de bits codificado del sistema de entrega 41 y para analizar sintácticamente el flujo de bits codificado. El subsistema 46 es operativo para afirmar los subflujos del flujo de bits codificado (que incluye un "primer" subflujo que comprende solo 2 canales codificados del flujo de bits codificado), y

las matrices de salida (P_0^2, P_1^2) correspondientes al primer subflujo, a la etapa de multiplicación de matriz 48 (para el procesamiento que da lugar a una presentación de mezcla descendente de 2 canales del contenido del programa de entrada de 8 canales original). El subsistema 46 es también operativo para afirmar los subflujos del flujo de bits codificado (un "segundo" subflujo que comprende los 8 canales codificados del flujo de bits codificado), y las matrices de salida correspondientes (P_0, P_1, \dots, P_n) a la etapa de multiplicación de matriz 47 para el procesamiento que da lugar a la reproducción sin pérdidas del programa de 8 canales original.

El subsistema de análisis sintáctico 46 (y el subsistema de análisis sintáctico 105 en la Figura 6) puede incluir (y/o implementar) herramientas de codificación y decodificación sin pérdidas adicionales (por ejemplo, codificación LPC, codificación de Huffman, etcétera).

La etapa de interpolación 60 está acoplada para recibir cada matriz semilla para el segundo subflujo (es decir, el conjunto inicial de matrices primitivas, P_0, P_1, \dots, P_n , para el tiempo t_1 , y cada conjunto actualizado de las matrices primitivas, P_0, P_1, \dots, P_n) incluidas en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de versiones interpoladas de cada matriz semilla. La etapa 60 está acoplada y configurada para pasar a través (a la etapa 47) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 47) las versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla, o entre los tiempos de actualización de matriz semilla posteriores).

La etapa de interpolación 61 está acoplada para recibir cada matriz semilla para el primer subflujo (es decir, el conjunto inicial de matrices primitivas, P_0^2 y P_1^2 , para el tiempo t_1 , y cada conjunto actualizado de matrices primitivas, P_0^2 y P_1^2) incluido en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de versiones interpoladas de cada una de dichas matrices semilla. La etapa 61 está acoplada y configurada para pasar a través (a la etapa 48) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 48) las versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla, o entre los tiempos de actualización de matriz semilla posteriores).

La etapa 48 multiplica 2 muestras de audio de los 2 canales (del flujo de bits codificado) que corresponden a los canales del primer subflujo por la cascada actualizada más recientemente de las matrices P_0^2 y P_1^2 (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0^2 y P_1^2 generadas por la etapa 61), y cada conjunto resultante de 2 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 0" para producir cada par de las muestras de la mezcla descendente de 2 canales requerida de los 8 canales de audio originales. La cascada de operaciones de matrización llevadas a cabo en el codificador 40 y el decodificador 42 es equivalente a la aplicación de una especificación de matriz de mezcla descendente que transforma los 8 canales de audio de entrada a la mezcla descendente de 2 canales.

La etapa 47 multiplica cada vector de 8 muestras de audio (una de cada una del conjunto completo de los 8 canales del flujo de bits codificado) por la cascada actualizada más recientemente de las matrices P_0, P_1, \dots, P_n (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0, P_1, \dots, P_n generada por la etapa 60) y cada conjunto resultante de 8 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque denominado "Asign Ca 1" para producir cada conjunto de 8 muestras del programa de 8 canales original recuperado sin pérdidas. Con el fin de que el audio de 8 canales de salida sea exactamente igual que el audio de 8 canales de entrada (para conseguir la característica "sin pérdidas" del sistema), las operaciones de matrización llevadas a cabo en el codificador 40 deberían ser exactamente (incluidos los efectos de cuantificación) las inversas de las operaciones de matrización llevadas a cabo en el decodificador 42 en el segundo subflujo del flujo de bits codificado (es decir, cada multiplicación en la etapa 47 del decodificador 42 por una cascada de las matrices P_0, P_1, \dots, P_n). De este modo, en la Figura 5, las operaciones de matrización en la etapa 43 del codificador 40 se identifican como una cascada de las matrices inversas de las matrices P_0, P_1, \dots, P_n , en la secuencia inversa aplicada en la etapa 47 del decodificador 42, concretamente: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

De este modo, la etapa 47 (con la etapa de permutación, Asign Ca 1) es un subsistema de multiplicación de matriz acoplado y configurado para aplicar de manera secuencial cada cascada de matrices primitivas emitidas desde la etapa de interpolación 60 al contenido de audio codificado extraído del flujo de bits codificado, para recuperar sin pérdidas los N canales de al menos un segmento del programa de audio multicanal que fue codificado por el codificador 40.

La etapa de permutación Asign Ca 1 del decodificador 42 aplica a la salida de la etapa 47 la inversa de la permutación de canales aplicada por el codificador 40 (es decir, la matriz de permutación representada por la etapa

"Asign Ca 1" del decodificador 42 es la inversa de aquella representada por el elemento "Asign Ca Inv 1" del codificador 40).

En las variaciones a los subsistemas 40 y 42 del sistema mostrado en la Figura 5, se omiten uno o más de los elementos o se incluyen unidades de procesamiento de datos de audio adicionales.

5 En las variaciones a la realización descrita del decodificador 42, el decodificador inventivo está configurado para llevar a cabo la recuperación sin pérdidas de los N canales del contenido de audio codificado a partir de un flujo de bits codificado indicativo de N canales de señal codificados, en los que los propios N canales de contenido de audio son una mezcla descendente del contenido de audio de un programa de audio de entrada de X canales (en el que X es un número entero cualquiera y N es menor que X), generada llevando a cabo operaciones de matriz en el
10 programa de audio de entrada de X canales para aplicar una mezcla variable en el tiempo a los X canales del programa de audio de entrada, determinando de esta manera los N canales del contenido de audio codificado del flujo de bits codificado. En dichas variaciones, el decodificador lleva a cabo la interpolación en las matrices de NxN primitivas proporcionadas con (por ejemplo, incluidas en) el flujo de bits codificado.

15 En una clase de realizaciones, la invención es un método para la renderización de un programa de audio multicanal, incluida mediante la realización de una transformación lineal (multiplicación de matriz) en las muestras de los canales del programa (por ejemplo, para generar una mezcla descendente del contenido del programa). La transformación lineal es dependiente del tiempo en el sentido de que la transformación lineal que se va a llevar a cabo en un tiempo durante el programa (es decir, en las muestras de los canales correspondientes a ese tiempo) difiere de la transformación lineal que se va a llevar a cabo en otro tiempo durante el programa. En algunas
20 realizaciones, el método emplea al menos una matriz semilla (que se puede implementar como una cascada de matrices primitivas unitarias) que determina la transformación lineal que se va a llevar a cabo en un primer tiempo durante el programa (es decir, en las muestras de los canales correspondientes al primer tiempo), e implementa la interpolación para determinar al menos una versión interpolada de la matriz semilla que determina la transformación lineal que se va a llevar a cabo en un segundo tiempo durante el programa. En realizaciones típicas, el método es
25 llevado a cabo por un decodificador (por ejemplo, el decodificador 40 de la Figura 5 o el decodificador 102 de la Figura 6) que está incluido en, o asociado con, un sistema de reproducción. Típicamente, el decodificador está configurado para llevar a cabo la recuperación sin pérdidas del contenido de audio de un flujo de bits de audio codificado indicativo del programa, y la matriz semilla (y cada versión interpolada de la matriz semilla) se implementa como una cascada de matrices primitivas (por ejemplo, matrices primitivas unitarias).

30 Típicamente, las actualizaciones de matriz de renderización (actualizaciones de la matriz semilla) tienen lugar de manera infrecuente (por ejemplo, una secuencia de las versiones actualizadas de la matriz semilla se incluye en el flujo de bits de audio codificado entregado al decodificador, pero hay intervalos de tiempo largos entre los segmentos del programa correspondientes a unos consecutivos de dichas versiones actualizadas), y una trayectoria de renderización deseada (por ejemplo, una secuencia deseada de las mezclas del contenido de los canales del
35 programa) entre las actualizaciones de matriz semilla se especifica paramétricamente (por ejemplo, por metadatos incluidos en el flujo de bits de audio codificado entregado al decodificador).

Cada matriz semilla (de una secuencia de matrices semilla actualizadas) se designará como $A(t_j)$, o $P_k(t_j)$ si es una matriz primitiva, en la que t_j es el tiempo (en el programa) correspondiente a la matriz semilla (es decir, el tiempo correspondiente a la matriz semilla "j^a"). Donde la matriz semilla se implementa como una cascada de las matrices primitivas, $P_k(t_j)$, el índice k indica la posición en la cascada de cada matriz primitiva. Típicamente, la matriz "k^a", $P_k(t_j)$, en una cascada de las matrices primitivas opera en el canal "k^o".

Cuando la transformación lineal (por ejemplo, la especificación de mezcla descendente), $A(t)$, es rápidamente variable, un codificador (por ejemplo, un codificador convencional) necesitaría transmitir las matrices semilla actualizadas frecuentemente con el fin de conseguir una aproximación cercana de $A(t)$.

45 Considérese una secuencia de las matrices primitivas $P_k(t1), P_k(t2), P_k(t3), \dots$, que operan en el mismo canal k pero en diferentes instantes de tiempo $t1, t2, t3, \dots$. En lugar de enviar las matrices primitivas actualizadas en cada uno de estos instantes, un método inventivo de realización envía, en el tiempo $t1$ (es decir, incluye en un flujo de bits codificado en una posición correspondiente al tiempo $t1$) una matriz primitiva semilla $P_k(t1)$, y una matriz delta semilla $\Delta_k(t1)$ que define la tasa de variación de los coeficientes de matriz. Por ejemplo, la matriz primitiva semilla y
50 la matriz delta semilla pueden tener la forma:

$$P_k(t1) = \begin{bmatrix} 1 & 0 & \ddots & \ddots & 0 \\ \ddots & 1 & \ddots & \ddots & \ddots \\ \alpha_0 & \cdots & \alpha_k & \ddots & \alpha_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Delta_k(t1) = \begin{bmatrix} 0 & 0 & \ddots & \ddots & 0 \\ \ddots & 0 & \ddots & \ddots & \ddots \\ \delta_0 & \cdots & \delta_k & \ddots & \delta_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5 Dado que $P_k(t1)$ es una matriz primitiva, es idéntica a la matriz de identidad de dimensiones $N \times N$ a excepción de una hilera (no trivial) (es decir, la hilera que comprende los elementos $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ en el ejemplo). En el ejemplo, la matriz $\Delta_k(t1)$ comprende ceros, a excepción de una hilera (no trivial) (es decir, la hilera que comprende los elementos $\delta_0, \delta_1, \dots, \delta_{N-1}$ en el ejemplo). El elemento α_k designa el 1 de los elementos $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{N-1}$ que tiene lugar en la diagonal de $P_k(t1)$, y el elemento δ_k designa el 1 de los elementos $\delta_0, \delta_1, \dots, \delta_{N-1}$ que tiene lugar en la diagonal de $\Delta_k(t1)$.

De este modo, la matriz primitiva en un instante de tiempo t (que tiene lugar después del tiempo $t1$) se interpola (por ejemplo, mediante las etapas 60 o 61 del decodificador 42 o las etapas 110, 111, 112 o 113 del decodificador 102) como: $P_k(t) = P_k(t1) + f(t)\Delta_k(t1)$,

10 donde $f(t)$ es el factor de interpolación para el tiempo t , y $f(t1) = 0$. Por ejemplo, si se desea la interpolación lineal, la función $f(t)$ puede ser de la forma $f(t) = a^*(t-t1)$, donde a es una constante. Si la interpolación se implementa en un decodificador, el decodificador debe estar configurado para conocer la función $f(t)$. Por ejemplo, los metadatos que determinan la función $f(t)$ pueden ser entregados al decodificador con el flujo de bits de audio codificado que se va a decodificar y a renderizar.

15 Mientras que lo anterior describe un caso general de la interpolación de matrices primitivas, con el elemento α_k igual a 1, $P_k(t1)$ es una matriz primitiva unitaria que es susceptible de inversión sin pérdidas. Sin embargo, con el fin de mantener la característica sin pérdidas en cada instante de tiempo, necesitaríamos establecer también $\delta_k = 0$, de modo que la matriz primitiva en cada instante sea susceptible de inversión sin pérdidas.

20 Nótese que $P_k(t)\mathbf{x}(t) = P_k(t1)\mathbf{x}(t) + f(t)(\Delta_k(t1)\mathbf{x}(t))$. De este modo, en lugar de actualizar la matriz primitiva semilla en cada instante de tiempo t , se podría calcular de manera equivalente 2 conjuntos intermedios de canales $P_k(t1)\mathbf{x}(t)$ y $\Delta_k(t1)\mathbf{x}(t)$, y combinarlos con el factor de interpolación $f(t)$. Esta aproximación es de manera típica menos costosa computacionalmente en comparación con la aproximación de actualizar la matriz primitiva cada instante en la que cada coeficiente delta tiene que ser multiplicado por el factor de interpolación.

25 Otra aproximación equivalente consiste en dividir $f(t)$ en un número entero r y una fracción $f(t)-r$, y luego conseguir la aplicación requerida de la matriz primitiva interpolada como:

$$P_k(t)\mathbf{x}(t) = (P_k(t1) + r\Delta_k(t1))\mathbf{x}(t) + (f(t) - r)(\Delta_k(t1)\mathbf{x}(t)). \quad (2)$$

Esta última aproximación (que utiliza la ecuación (2)) sería de este modo una mezcla de las 2 aproximaciones tratadas previamente.

30 En el TrueHD, 0,833 ms (40 muestras a 48 kHz) de valor de audio se definen como una unidad de acceso. Si la matriz delta Δ_k se define como la tasa de variación de la matriz primitiva P_k por unidad de acceso, y si definimos $f(t) = (t-t1)/T$, donde T es la longitud de la unidad de acceso, entonces r en la ecuación (2) aumenta en 1 cada unidad de acceso, y $f(t)-r$ es simplemente una función de la desviación de una muestra dentro de una unidad de acceso. De este modo, el valor fraccionario $f(t)-r$ no necesita necesariamente ser calculado y se puede derivar simplemente a partir de una tabla de consulta indexada mediante las desviaciones dentro de una unidad de acceso. En el fin de cada unidad de acceso, $P_k(t1) + r\Delta_k(t1)$ se actualiza por la suma de $\Delta_k(t1)$. En general T no necesita corresponder a una unidad de acceso y en lugar de eso puede ser cualquier segmentación fija de la señal, por ejemplo, podría ser un bloque de longitud de 8 muestras.

35 Una simplificación adicional, si bien una aproximación, consistiría en ignorar la parte fraccionaria $f(t)-r$ por completo y actualizar periódicamente $P_k(t1) + r\Delta_k(t1)$. Esto produce esencialmente una actualización de matriz de constante a intervalos, pero sin el requisito de transmitir matrices primitivas con frecuencia.

40 La Figura 3 es un diagrama de bloques de la circuitería empleada en una realización de la invención para aplicar una matriz primitiva de 4×4 (implementada con aritmética de precisión finita) a 4 canales de un programa de audio. La matriz primitiva es una matriz primitiva semilla, una de cuyas hileras no triviales comprende los elementos $\alpha_0, \alpha_1, \alpha_2$ y α_3 . Se contempla que 4 de dichas matrices primitivas, cada una para la transformación de muestras de uno diferente de los 4 canales, se dispondrían en cascada para transformar las muestras de los 4 canales. Dicha circuitería se podría utilizar cuando las matrices primitivas se actualizan primero mediante interpolación, y las matrices primitivas actualizadas se aplican en los datos de audio.

45 La Figura 4 es un diagrama de bloques de la circuitería empleada en una realización de la invención para aplicar una matriz primitiva de 3×3 (implementada con aritmética de precisión finita) a 3 canales de un programa de audio. La matriz primitiva es una matriz primitiva interpolada, generada de acuerdo con una realización de la invención a partir de una matriz primitiva semilla $P_k(t1)$ una de cuyas hileras no triviales comprende los elementos α_0, α_1 y α_2 , y una matriz delta semilla $\Delta_k(t1)$ cuya hilera no trivial comprende los elementos δ_0, δ_1 y δ_2 , y una función de interpolación $f(t)$. De este modo, la matriz primitiva en un instante de tiempo t (que tiene lugar después del tiempo $t1$) se interpola como: $P_k(t) = P_k(t1) + f(t)\Delta_k(t1)$, donde $f(t)$ es un factor de interpolación para el tiempo t (el valor de la función de

interpolación $f(t)$ en el tiempo t), y $f(t_1) = 0$. Se contempla que 3 de dichas matrices primitivas, cada una de ellas para la transformación de las muestras de un canal diferente de los 3 canales, se dispondrían en cascada para transformar las muestras de los 3 canales. Dicha circuitería se podría utilizar cuando una matriz primitiva semilla o parcialmente actualizada se aplica en los datos de audio y la matriz delta se aplica en los datos de audio y las dos se combinan entre sí utilizando el factor de interpolación.

La circuitería de la Figura 3 está configurada para aplicar la matriz primitiva semilla a 4 canales de programa de audio S1, S2, S3 y S4 (es decir, para multiplicar las muestras de los canales por la matriz). Más específicamente, una muestra del canal S1 se multiplica por el coeficiente α_0 (identificado como "coef[p,0]_m") de la matriz, una muestra del canal S2 se multiplica por el coeficiente α_1 (identificado como "coef[p,1]_m") de la matriz, una muestra del canal S3 se multiplica por el coeficiente α_2 (identificado como "coef[p,2]_m") de la matriz, y una muestra del canal S4 se multiplica por el coeficiente α_3 (identificado como "coef[p,3]_m") de la matriz. Los productos se suman en el elemento de suma 10, y cada salida de suma del elemento 10 se cuantifica después en la etapa de cuantificación Qss para generar el valor cuantificado que es la versión transformada (incluida en el canal S2') de la muestra del canal S2. En una implementación típica, cada muestra de cada uno de los canales S1, S2, S3 y S4 comprende 24 bits (como se indica en la Figura 3), y la salida de cada elemento de multiplicación comprende 38 bits (como también se indica en la Figura 3), y la etapa de cuantificación Qss emite un valor cuantificado de 24 bits en respuesta a cada valor de 38 bits que se introduce en la misma.

La circuitería de la Figura 4 está configurada para aplicar la matriz primitiva interpolada a los 3 canales de programa de audio C1, C2 y C3 (es decir, para multiplicar las muestras de los canales por la matriz). Más específicamente, una muestra del canal C1 se multiplica por el coeficiente α_0 (identificado como "coef[p,0]_m") de la matriz primitiva semilla, una muestra del canal C2 se multiplica por el coeficiente α_1 (identificado como "coef[p,1]_m") de la matriz primitiva semilla, y una muestra del canal S3 se multiplica por el coeficiente α_2 (identificado como "coef[p,2]_m") de la matriz primitiva semilla. Los productos se suman en el elemento de suma 12, y cada salida de suma del elemento 12 se añade después (en la etapa 14) a la salida de valor correspondiente de la etapa de factor de interpolación 13. La salida de valor de la etapa 14 se cuantifica en la etapa de cuantificación Qss para generar el valor cuantificado que es la versión transformada (incluida en el canal C3') de la muestra del canal C3.

La misma muestra del canal C1 se multiplica por el coeficiente δ_0 (identificado como "coef[p,0]_delta") de la matriz delta semilla, la muestra del canal C2 se multiplica por el coeficiente δ_1 (identificado como "coef[p,1]_delta") de la matriz delta semilla, y la muestra del canal S3 se multiplica por el coeficiente δ_2 (identificado como "coef[p,2]_delta") de la matriz delta semilla. Los productos se suman en el elemento de suma 11, y cada salida de suma del elemento 11 se cuantifica entonces en la etapa de cuantificación Qfin para generar un valor cuantificado que se multiplica después (en la etapa de factor de interpolación 13) por el valor actual de la función de interpolación, $f(t)$.

En una implementación típica de la Figura 4, cada muestra de cada uno de los canales C1, C2 y C3 comprende 32 bits (como se indica en la Figura 4), y la salida de cada uno de los elementos de suma 11, 12 y 14 comprende 50 bits (como también se indica en la Figura 4), y cada una de las etapas de cuantificación Qfin y Qss emite un valor cuantificado de 32 bits en respuesta a cada valor de 50 bits que se introduce en la misma.

Por ejemplo, una variación al circuito de la Figura 4 podría transformar un vector de las muestras de x canales de audio, en las que $x = 2, 4, 8$ o N canales. Una cascada de x de dichas variaciones al circuito de la Figura 4 podría llevar a cabo la multiplicación de matriz de dichos x canales por una matriz semilla de $x \times x$ (o una versión interpolada de dicha matriz semilla). Por ejemplo, dicha cascada de x de dichas variaciones al circuito de la Figura 4 podría implementar las etapas 60 y 47 del decodificador 42 (en las que $x = 8$), o las etapas 61 y 48 del decodificador 42 (en las que $x = 2$), o las etapas 113 y 109 del decodificador 102 (en las que $x = N$), o las etapas 112 y 108 del decodificador 102 (en las que $x = 8$), o las etapas 111 y 107 del decodificador 102 (en las que $x = 6$), o las etapas 110 y 106 del decodificador 102 (en las que $x = 2$).

En la realización de la Figura 4, la matriz primitiva semilla y la matriz delta semilla se aplican en paralelo a cada conjunto (vector) de las muestras de entrada (donde cada uno de dichos vectores incluye una muestra de cada uno de los canales de entrada).

Con referencia a la Figura 6, describimos a continuación una realización de la invención en la que el programa de audio que se va a decodificar es un programa de audio basado en objetos de N canales. El sistema de la Figura 6 incluye el codificador 100 (una realización del codificador inventivo), el subsistema de entrega 31 y el decodificador 102 (una realización del decodificador inventivo), acoplados entre sí como se muestra. Aunque el subsistema 102 se denomina en la presente memoria "decodificador", debería entenderse que se puede implementar como un sistema de reproducción que incluye un subsistema de decodificación (configurado para analizar sintácticamente y decodificar un flujo de bits indicativo de un programa de audio multicanal codificado) y otros subsistemas configurados para implementar la renderización y al menos algunas etapas de la reproducción de la salida del subsistema de decodificación. Algunas realizaciones de la invención son decodificadores que no están configurados para llevar a cabo la renderización y/o la reproducción (y que se utilizarían típicamente con un sistema de renderización y/o reproducción separado). Algunas realizaciones de la invención son sistemas de reproducción (por

ejemplo, un sistema de reproducción que incluye un subsistema de decodificación y otros subsistemas configurados para implementar la renderización y al menos algunas etapas de la reproducción de la salida del subsistema de decodificación.

5 En el sistema de la Figura 6, el codificador 100 está configurado para codificar el programa de audio basado en objetos de N canales como un flujo de bits codificado que incluye 4 subflujos, y el decodificador 102 está configurado para decodificar el flujo de bits codificado para renderizar o el programa de N canales original (sin pérdidas) o una mezcla descendente de 8 canales del programa de N canales original, o una mezcla descendente de 6 canales del programa de N canales original, o una mezcla descendente de 2 canales del programa de N canales original. El codificador 100 está acoplado y configurado para generar el flujo de bits codificado y para afirmar el flujo de bits
10 codificado al sistema de entrega 31.

El sistema de entrega 31 está acoplado y configurado para entregar (por ejemplo, almacenando y/o transmitiendo) el flujo de bits codificado al decodificador 102. En algunas realizaciones, el sistema 31 implementa la entrega de (por ejemplo, transmite) un programa de audio multicanal codificado por un sistema de difusión o una red (por ejemplo, internet) al decodificador 102. En algunas realizaciones, el sistema 31 almacena un programa de audio multicanal codificado en un medio de almacenamiento (por ejemplo, un disco o un conjunto de discos), y el decodificador 102
15 está configurado para leer el programa del medio de almacenamiento.

El bloque denominado "Asign Ca Inv3" en el codificador 100 está configurado para llevar a cabo la permutación de canales (equivalente a la multiplicación por una matriz de permutación) en los canales del programa de entrada. Los canales permutados luego experimentan la codificación en la etapa 101, que emite N canales de señal codificados.
20 Los canales de señal codificados pueden (pero no necesitan) corresponder a los canales de altavoz de reproducción. Los canales de señal codificados a veces se denominan canales "internos" dado que un decodificador (y/o un sistema de renderización) típicamente decodifica y renderiza el contenido de los canales de señal codificados para recuperar el audio de entrada, de modo que los canales de señal codificados sean "internos" al sistema de codificación/decodificación. La codificación llevada a cabo en la etapa 101 es equivalente a la multiplicación de cada conjunto de muestras de los canales permutados por una matriz de codificación (implementada como una cascada
25 de las multiplicaciones de matriz, identificada como $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

Cada matriz $P_n^{-1}, \dots, P_1^{-1}$ y P_0^{-1} (y de este modo la cascada aplicada por la etapa 101) se determina en el subsistema 103 y se actualiza de vez en cuando (típicamente de manera infrecuente) de acuerdo con una mezcla variable en el tiempo especificada de los N canales del programa a N canales de señal codificados que se ha especificado durante
30 el intervalo de tiempo.

En las variaciones a la realización ejemplar de la Figura 6, el programa de audio de entrada comprende un número arbitrario (N o X, donde X es mayor que N) de canales. En dichas variaciones, los N canales de programa de audio multicanal que son indicados por el flujo de bits codificado emitido desde el codificador, que se puede recuperar sin pérdidas por el decodificador, pueden ser N canales de contenido de audio que han sido generados a partir del
35 programa de audio de entrada de X canales llevando a cabo operaciones de matriz en el programa de audio de entrada de X canales para aplicar una mezcla variable en el tiempo a los X canales del programa de audio de entrada, determinando de esta manera el contenido de audio codificado del flujo de bits codificado.

El subsistema de determinación de matrices 103 de la Figura 6 está configurado para generar datos indicativos de los coeficientes de 4 conjuntos de matrices de salida (donde un conjunto corresponde a cada uno de los 4 subflujos de los canales codificados). Cada conjunto de matrices de salida se actualiza de vez en cuando, de modo que los coeficientes se actualicen también de vez en cuando.
40

Un conjunto de matrices de salida consta de 2 matrices de renderización, $P_0^2(t), P_1^2(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones 2×2 , y es para la renderización de un primer subflujo (un subflujo de mezcla descendente) que comprende 2 de los canales de audio codificados del flujo de bits codificado (para renderizar una mezcla descendente de 2 canales del audio de entrada). Otro conjunto de matrices de salida puede constar de nada menos que 6 matrices de renderización, $P_0^6(t), P_1^6(t), P_2^6(t), P_3^6(t), P_4^6(t)$ y $P_5^6(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones 6×6 , y es para la renderización de un segundo subflujo (un subflujo de mezcla descendente) que comprende 6 de los canales de audio codificados del flujo de bits codificado (para renderizar una mezcla descendente de 6 canales del audio de entrada). Otro conjunto de matrices de salida consta de nada menos que 8 matrices de renderización, $P_0^8(t), P_1^8(t), \dots, P_7^8(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones 8×8 , y es para la renderización de un tercer subflujo (un subflujo de mezcla descendente) que comprende 8 de los canales de audio codificados del flujo de bits codificado (para renderizar una mezcla descendente de 8 canales del audio de entrada).
45
50

El otro conjunto de matrices de salida consta de N matrices de renderización, $P_0(t), P_1(t), \dots, P_n(t)$, cada una de las cuales es una matriz primitiva (preferiblemente una matriz primitiva unitaria) de dimensiones $N \times N$, y es para la
55

5 renderización de un cuarto subflujo que comprende todos los canales de audio codificados del flujo de bits codificado (para la recuperación sin pérdidas del programa de audio de entrada de N canales). Para cada tiempo, t, una cascada de las matrices de renderización, $P_0^2(t)$, $P_1^2(t)$, se puede interpretar como una matriz de renderización para los canales del primer subflujo, una cascada de las matrices de renderización, $P_0^6(t)$, $P_1^6(t)$, ..., $P_5^6(t)$, también se puede interpretar como una matriz de renderización para los canales del segundo subflujo, una cascada de las matrices de renderización, $P_0^8(t)$, $P_1^8(t)$, ..., $P_7^8(t)$, también se puede interpretar como una matriz de renderización para los canales del tercer subflujo, y una cascada de las matrices de renderización, $P_0(t)$, $P_1(t)$, ..., $P_n(t)$, es equivalente a una matriz de renderización para los canales del cuarto subflujo.

10 Los coeficientes (de cada matriz de renderización) que se emiten desde el subsistema 103 al subsistema de empaquetado 104 son metadatos que indican la ganancia relativa o absoluta de cada canal que se va a incluir en una mezcla correspondiente de canales del programa. Los coeficientes de cada matriz de renderización (para un instante de tiempo durante el programa) representan cuánto debería contribuir cada uno de los canales de una mezcla a la mezcla de contenido de audio (en el instante correspondiente de la mezcla renderizada) indicada por la señal de altavoz para un altavoz de sistema de reproducción particular.

15 Los N canales de audio codificados (emitidos desde la etapa de codificación 101), los coeficientes de matriz de salida (generados por el subsistema 103), y típicamente también los datos adicionales (por ejemplo, para su inclusión como metadatos en el flujo de bits codificado) se afirman al subsistema de empaquetado 104, que los reúne en el flujo de bits codificado que se afirma luego al sistema de entrega 31.

20 El flujo de bits codificado incluye datos indicativos de los N canales de audio codificados, los 4 conjuntos de matrices de salida variables en el tiempo (un conjunto correspondiente a cada uno de los 4 subflujos de los canales codificados), y típicamente también datos adicionales (por ejemplo, metadatos relativos al contenido de audio).

La etapa 103 del codificador 100 actualiza cada conjunto de matrices de salida (por ejemplo, el conjunto P_0^2, P_1^2 , o el conjunto P_0, P_1, \dots, P_n) de vez en cuando. El primer conjunto de las matrices P_0^2, P_1^2 que se emite (en un primer tiempo, t1) es una matriz semilla (implementada como una cascada de matrices primitivas, por ejemplo, matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de los 2 canales de la salida codificada de la etapa 101, correspondientes al primer tiempo). El primer conjunto de las matrices $P_0^6(t)$, $P_1^6(t)$, ..., $P_n^6(t)$, que se emite (en el tiempo t1) es una matriz semilla (implementada como una cascada de matrices primitivas, por ejemplo, matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de los 6 canales de la salida codificada de la etapa 101, correspondientes al primer tiempo). El primer conjunto de las matrices $P_0^8(t)$, $P_1^8(t)$, ..., $P_n^8(t)$, que se emite (en el tiempo t1) es una matriz semilla (implementada como una cascada de matrices primitivas, por ejemplo, matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de los 8 canales de la salida codificada de la etapa 101, correspondientes al primer tiempo). El primer conjunto de las matrices P_0, P_1, \dots, P_n que se emite (en el tiempo t1) es una matriz semilla (implementada como una cascada de matrices primitivas unitarias) que determina una transformación lineal que se va a llevar a cabo en el primer tiempo durante el programa (es decir, en las muestras de todos los canales de la salida codificada de la etapa 101 correspondientes al primer tiempo).

40 Cada conjunto actualizado de las matrices P_0^2, P_1^2 que se emite desde la etapa 103 es una matriz semilla actualizada (implementada como una cascada de matrices primitivas, que también se puede denominar una cascada de matrices primitivas semilla) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de los 2 canales de la salida codificada de la etapa 101, correspondientes al tiempo de actualización). Cada conjunto actualizado de las matrices $P_0^6(t)$, $P_1^6(t)$, ..., $P_n^6(t)$, que se emite desde la etapa 103 es una matriz semilla actualizada (implementada como una cascada de matrices primitivas, que también se puede denominar una cascada de matrices primitivas semilla) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de 6 canales de la salida codificada de la etapa 101, correspondientes al tiempo de actualización). Cada conjunto actualizado de las matrices $P_0^8(t)$, $P_1^8(t)$, ..., $P_n^8(t)$ que se emite desde la etapa 103 es una matriz semilla actualizada (implementada como una cascada de matrices primitivas, que también se puede denominar una cascada de matrices primitivas semilla) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de los 2 canales de la salida codificada de la etapa 101, correspondientes al tiempo de actualización). Cada conjunto actualizado de las matrices P_0, P_1, \dots, P_n que se emite desde la etapa 103 es también una matriz semilla (implementada como una cascada de matrices primitivas unitarias, que también se puede denominar una cascada de matrices primitivas semilla unitarias) que determina una transformación lineal que se va a llevar a cabo en el tiempo de actualización durante el programa (es decir, en las muestras de todos los canales de la salida codificada de la etapa 101 correspondientes al primer tiempo).

La etapa de salida 103 también está configurada para emitir valores de interpolación, que (con una función de interpolación para cada matriz semilla) permiten al decodificador 102 generar las versiones interpoladas de las matrices semilla (correspondientes a los tiempos después del primer tiempo, t_1 , y entre los tiempos de actualización). Los valores de interpolación (que pueden incluir datos indicativos de cada función de interpolación) son incluidos por la etapa 104 en el flujo de bits codificado emitido desde el codificador 100. Los ejemplos de dichos valores de interpolación se describen en otro lugar en la presente memoria (los valores de interpolación pueden incluir una matriz delta para cada matriz semilla).

Con referencia al decodificador 102 de la Figura 6, el subsistema de análisis sintáctico 105 está configurado para aceptar (leer o recibir) el flujo de bits codificado desde el sistema de entrega 31 y para analizar sintácticamente el flujo de bits codificado. El subsistema 105 es operativo para afirmar un primer subflujo que comprende solo 2 canales codificados del flujo de bits codificado, las matrices de salida (P_0, P_1, \dots, P_n) correspondientes al cuarto

subflujo (superior), y las matrices de salida (P_0^2, P_1^2) correspondientes al primer subflujo, a la etapa de multiplicación de matriz 106 (para el procesamiento que da lugar a una presentación de mezcla descendente de 2 canales de contenido del programa de entrada de N canales original). El subsistema 105 es operativo para afirmar el segundo subflujo del flujo de bits codificado que comprende 6 canales codificados del flujo de bits codificado, y las matrices de salida ($P_0^6(t), P_1^6(t), \dots, P_n^6(t)$) correspondientes al segundo subflujo, a la etapa de multiplicación de matriz 107 (para el procesamiento que da lugar a una presentación de mezcla descendente de 6 canales del contenido del programa de entrada de N canales original). El subsistema 105 es operativo para afirmar un tercer subflujo del flujo de bits codificado que comprende 8 canales codificados del flujo de bits codificado, y las matrices de salida ($P_0^8(t), P_1^8(t), \dots, P_n^8(t)$) correspondientes al tercer subflujo, a la etapa de multiplicación de matriz 108 (para el procesamiento que da lugar a una presentación de mezcla descendente de 8 canales del contenido del programa de entrada de N canales original). El subsistema 105 también es operativo para afirmar el cuarto subflujo (superior) del flujo de bits codificado (que comprende todos los canales codificados del flujo de bits codificado), y las matrices de salida correspondientes (P_0, P_1, \dots, P_n) a la etapa de multiplicación de matriz 109 para el procesamiento que da lugar a la reproducción sin pérdidas del programa de N canales original.

La etapa de interpolación 113 está acoplada para recibir cada matriz semilla para el cuarto subflujo (es decir, el conjunto inicial de las matrices primitivas, P_0, P_1, \dots, P_n , para el tiempo t_1 , y cada conjunto actualizado de las matrices primitivas, P_0, P_1, \dots, P_n) incluido en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de las versiones interpoladas de cada matriz semilla. La etapa 113 está acoplada y configurada para pasar a través (a la etapa 109) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 109) las versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla o entre los tiempos de actualización de matriz semilla posteriores).

La etapa de interpolación 112 está acoplada para recibir cada matriz semilla para el tercer subflujo (es decir, el conjunto inicial de las matrices primitivas, $P_0^8, P_1^8, \dots, P_n^8$, para el tiempo t_1 , y cada conjunto actualizado de las matrices primitivas, $P_0^8, P_1^8, \dots, P_n^8$) incluido en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de las versiones interpoladas de cada una de dichas matrices semilla. La etapa 112 está acoplada y configurada para pasar a través (a la etapa 108) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 108) las versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla o entre los tiempos de actualización de matriz semilla posteriores).

La etapa de interpolación 111 está acoplada para recibir cada matriz semilla para el segundo subflujo (es decir, el conjunto inicial de las matrices primitivas, $P_0^6, P_1^6, \dots, P_n^6$, para el tiempo t_1 , y cada conjunto actualizado de las matrices primitivas, $P_0^6, P_1^6, \dots, P_n^6$) incluido en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de versiones interpoladas de cada una de dichas matrices semilla. La etapa 111 está acoplada y configurada para pasar a través (a la etapa 107) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 107) versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla, o entre los tiempos de actualización de matriz semilla posteriores).

La etapa de interpolación 110 está acoplada para recibir cada matriz semilla para el primer subflujo (es decir, el conjunto inicial de las matrices primitivas, P_0^2 y P_1^2 , para el tiempo t_1 , y cada conjunto actualizado de las matrices primitivas, P_0^2 y P_1^2) incluido en el flujo de bits codificado, y los valores de interpolación (también incluidos en el flujo de bits codificado) para la generación de versiones interpoladas de cada una de dichas matrices semilla. La etapa 110 está acoplada y configurada para pasar a través (a la etapa 106) cada una de dichas matrices semilla, y para generar (y afirmar a la etapa 106) versiones interpoladas de cada una de dichas matrices semilla (cada versión interpolada correspondiente a un tiempo después del primer tiempo, t_1 , y antes del primer tiempo de actualización de matriz semilla, o entre los tiempos de actualización de matriz semilla posteriores).

5 La etapa 106 multiplica cada vector de 2 muestras de audio de los 2 canales codificados del primer subflujo por la cascada actualizada más recientemente de las matrices P_0^2 y P_1^2 (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0^2 y P_1^2 generadas por la etapa 110), y cada conjunto resultante de 2 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 0" para producir cada par de muestras de la mezcla descendente de 2 canales requerida de los N canales de audio originales. La cascada de operaciones de matrización llevadas a cabo en el codificador 40 y el decodificador 102 es equivalente a la aplicación de una especificación de matriz de mezcla descendente que transforma los N canales de audio de entrada en la mezcla descendente de 2 canales.

10 La etapa 107 multiplica cada vector de 6 muestras de audio de los 6 canales codificados del segundo subflujo por la cascada actualizada más recientemente de las matrices P_0^6, \dots, P_n^6 (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0^6, \dots, P_n^6 generada por la etapa 111), y cada conjunto resultante de 6 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 1" para producir cada conjunto de muestras de la mezcla descendente de 6 canales requerida de los N canales de audio originales. La cascada de operaciones de matrización llevadas a cabo en el codificador 100 y el decodificador 102 es equivalente a la aplicación de una especificación de matriz de mezcla descendente que transforma los N canales de audio de entrada en la mezcla descendente de 6 canales.

20 La etapa 108 multiplica cada vector de 8 muestras de audio de los 8 canales codificados (del tercer subflujo) por la cascada actualizada más recientemente de las matrices P_0^8, \dots, P_n^8 (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0^8, \dots, P_n^8 generada por la etapa 112), y cada conjunto resultante de 8 muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 2" para producir cada par de muestras de la mezcla descendente de 8 canales requerida de los N canales de audio originales. La cascada de operaciones de matrización llevadas a cabo en el codificador 100 y el decodificador 102 es equivalente a la aplicación de una especificación de matriz de mezcla descendente que transforma los N canales de audio de entrada en la mezcla descendente de 8 canales.

30 La etapa 109 multiplica cada vector de N muestras de audio (una de cada uno del conjunto completo de N canales codificados del flujo de bits codificado) por la cascada actualizada más recientemente de las matrices P_0, P_1, \dots, P_n (por ejemplo, una cascada de las versiones interpoladas más recientes de las matrices P_0, P_1, \dots, P_n generada por la etapa 113) y cada conjunto resultante de N muestras transformadas linealmente experimenta la permutación de canales (equivalente a la multiplicación por una matriz de permutación) representada por el bloque titulado "Asign Ca 3" para producir cada conjunto de N muestras del programa de N canales original recuperado sin pérdidas. Con el fin de que el audio de N canales de salida sea exactamente igual que el audio de N canales de entrada (para conseguir la característica "sin pérdidas" del sistema), las operaciones de matrización llevadas a cabo en el codificador 100 deberían ser exactamente (incluidos los efectos de cuantificación) las inversas de las operaciones de matrización llevadas a cabo en el decodificador 102 en el cuarto subflujo del flujo de bits codificado (es decir, cada multiplicación en la etapa 109 del decodificador 102 por una cascada de las matrices P_0, P_1, \dots, P_n). De este modo, en la Figura 6, las operaciones de matrización en la etapa 103 del codificador 100 se identifican como una cascada de las matrices inversas de las matrices P_0, P_1, \dots, P_n , en la secuencia inversa aplicada en la etapa 109 del decodificador 102, concretamente: $P_n^{-1}, \dots, P_1^{-1}, P_0^{-1}$.

45 En algunas implementaciones, el subsistema de análisis sintáctico 105 está configurado para extraer una palabra de comprobación del flujo de bits codificado, y la etapa 109 está configurada para verificar si los N canales (de al menos un segmento de un programa de audio multicanal) recuperados por la etapa 109 se han recuperado correctamente, cotejando una segunda palabra de comprobación derivada (por ejemplo, por la etapa 109) de muestras de audio generadas por la etapa 109 con la palabra de comprobación extraída del flujo de bits codificado.

La etapa "Asign Ca 3" del decodificador 102 aplica a la salida de la etapa 109 la inversa de la permutación de canales aplicada por el codificador 100 (es decir, la matriz de permutación representada por la etapa "Asign Ca 3" del decodificador 102 es la inversa de aquella representada por el elemento "Asign Ca Inv3" del codificador 100).

50 En las variaciones a los subsistemas 100 y 102 del sistema mostrado en la Figura 6, se omiten uno o más de los elementos o se incluyen unidades de procesamiento de datos de audio adicionales.

55 Los coeficientes de matrices de renderización P_0^8, \dots, P_n^8 (o P_0^6, \dots, P_n^6 , o P_0^2 y P_1^2) afirmados a la etapa 108 (o 107 o 106) del decodificador 100 son metadatos (por ejemplo, metadatos de posición espacial) del flujo de bits codificado que son indicativos de (o pueden ser procesados con otros datos que van a ser indicativos de) la ganancia relativa o absoluta de cada canal de altavoz que se va a incluir en una mezcla descendente de los canales del contenido de N canales original codificado por el codificador 100.

Por el contrario, la configuración del sistema de altavoz de reproducción que se va a emplear para renderizar un conjunto completo de canales de un programa de audio basado en objetos (que es recuperado sin pérdidas por el decodificador 102) es típicamente desconocida en el tiempo en que se genera el flujo de bits codificado por el codificador 100. Los N canales recuperados sin pérdidas por el decodificador 102 pueden necesitar ser procesados (por ejemplo, en un sistema de renderización incluido en el decodificador 102 [pero no mostrado en la Figura 6] o acoplado al decodificador 102) con otros datos (por ejemplo, datos indicativos de la configuración de un sistema de altavoz de reproducción particular) para determinar cuánto debería contribuir cada canal del programa a una mezcla del contenido de audio (en cada instante de la mezcla renderizada) indicado por la señal de altavoz para un altavoz de sistema de reproducción particular. Dicho sistema de renderización puede procesar los metadatos de trayectoria espacial en (o asociados con) cada canal de objeto recuperado sin pérdidas, para determinar las señales de altavoz para los altavoces del sistema de altavoz de reproducción particular que se va a emplear para la reproducción del contenido recuperado sin pérdidas.

En algunas realizaciones del codificador inventivo, el codificador está provisto de (o genera) una especificación variable dinámicamente $\mathbf{A}(t)$ que especifica cómo transformar todos los canales de un programa de audio de N canales (por ejemplo, un programa de audio basado en objetos) en un conjunto de N canales codificados, y al menos una especificación variable dinámicamente de mezcla descendente que especifica cada mezcla descendente del contenido de los N canales codificados a una presentación de M1 canales (en la que M1 es menor que N, por ejemplo, $M1 = 2$, o $M1 = 8$, cuando N es mayor que 8). En algunas realizaciones, la función del codificador consiste en empaquetar el audio codificado y los datos indicativos de cada una de dichas especificaciones variables dinámicamente en un flujo de bits codificado que tienen un formato predeterminado (por ejemplo, un flujo de bits de TrueHD). Por ejemplo, esto se puede realizar de tal manera que un decodificador heredado (por ejemplo, un decodificador TrueHD heredado) sea capaz de recuperar al menos una presentación de mezcla descendente (que tenga M1 canales), mientras que un decodificador mejorado puede utilizarse para recuperar (sin pérdidas) el programa de audio de N canales original. Dadas las especificaciones variables dinámicamente, el codificador puede suponer que el decodificador determinará las matrices primitivas interpoladas P_0, P_1, \dots, P_n a partir de los valores de interpolación (por ejemplo, información de matrices primitivas semilla y matrices delta semilla) incluidos en el flujo de bits codificado que se va a entregar al decodificador. El decodificador lleva a cabo después la interpolación para determinar las matrices primitivas interpoladas que invierten las operaciones del codificador que produjo el contenido de audio codificado del flujo de bits codificado (por ejemplo, para recuperar sin pérdidas el contenido que fue codificado, experimentando operaciones de matriz, en el codificador). Opcionalmente el codificador puede elegir las matrices primitivas para los subflujos inferiores (es decir, los subflujos indicativos de mezclas descendentes del contenido de un subflujo de N canales superior) para ser matrices primitivas no interpoladas (e incluir una secuencia de conjuntos de dichas matrices primitivas no interpoladas en el flujo de bits codificado), mientras que también supone que el decodificador determinará las matrices primitivas interpoladas (P_0, P_1, \dots, P_n) para la recuperación sin pérdidas del contenido del subflujo (de N canales) superior a partir de los valores de interpolación (por ejemplo, información de matriz primitiva semilla y matriz delta semilla) incluidos en el flujo de bits codificado que se va a entregar al decodificador.

Por ejemplo, un codificador (por ejemplo, la etapa 44 del codificador 40 o la etapa 103 del codificador 100) puede estar configurado para elegir matrices primitivas semilla y matrices delta semilla (para su uso con una función de interpolación, $f(t)$), mediante el muestreo de la especificación $\mathbf{A}(t)$ en diferentes instantes de tiempo t_1, t_2, t_3, \dots (que pueden estar estrechamente espaciados), derivando las matrices primitivas semilla correspondientes (por ejemplo, como en un codificador TrueHD convencional) y para calcular después la tasa de variación de los elementos individuales en las matrices primitivas semilla para calcular los valores de interpolación (por ejemplo, información "delta" indicativa de una secuencia de matrices delta semilla). El primer conjunto de matrices primitivas semilla serían las matrices primitivas derivadas de la especificación para el primero de dichos instantes de tiempo, $\mathbf{A}(t_1)$. Es posible que un subconjunto de las matrices primitivas pueda no cambiar en absoluto a lo largo del tiempo, en cuyo caso el decodificador respondería a la información de control adecuada en flujo de bits codificado poniendo a cero cualquier información delta correspondiente (es decir, para establecer la tasa de variación de dicho subconjunto de matrices primitivas a cero).

Las variaciones a la realización de la Figura 6 del codificador y el decodificador inventivos pueden omitir la interpolación para algunos (es decir, al menos uno) de los subflujos del flujo de bits codificado. Por ejemplo, se pueden omitir las etapas de interpolación 110, 111 y 112, y se pueden actualizar las matrices correspondientes P_0^2, P_1^2 y $P_0^6, P_1^6, \dots, P_n^6$, and $P_0^8, P_1^8, \dots, P_n^8$ (en el flujo de bits codificado) con suficiente frecuencia de modo que la interpolación entre los instantes en los que se actualizan no sea necesaria. Como otro ejemplo, si las matrices $P_0^6, P_1^6, \dots, P_n^6$ se actualizan con la frecuencia suficiente de modo que la interpolación en los tiempos entre las actualizaciones no sea necesaria, la etapa de interpolación 111 es innecesaria y puede ser omitida. De este modo, un decodificador convencional (no configurado de acuerdo con la invención para llevar a cabo la interpolación) podría renderizar la presentación de mezcla descendente de 6 canales en respuesta al flujo de bits codificado.

Como se señaló anteriormente, las especificaciones de matrices de renderización dinámicas (por ejemplo, $\mathbf{A}(t)$) pueden originarse no solo a partir de la necesidad de renderizar programas de audio basados en objetos, sino también debido a la necesidad de implementar la protección de recorte. Las matrices primitivas interpoladas pueden

permitir un aumento más veloz y una liberación de la protección de recorte de una mezcla descendente, así como una reducción de la tasa de datos requerida para transmitir los coeficientes de matrización.

5 A continuación, describimos un ejemplo de operación de una implementación del sistema de la Figura 6. En este caso, el programa de entrada de N canales es un programa de audio basado en objetos de 3 canales que incluye un canal de base, C, y 2 canales de objeto, U y V. Se desea que el programa se codifique para su transporte mediante un flujo del TrueHD que tiene 2 subflujos de tal manera que una mezcla descendente de 2 canales (una renderización del programa a una configuración de altavoz de 2 canales) pueda ser recuperada utilizando el primer subflujo y el programa de entrada de 3 canales original puede ser recuperado sin pérdidas utilizando ambos subflujos.

10 Haremos que la ecuación de renderización (o la ecuación de mezcla descendente) del programa de entrada a la mezcla de 2 canales sea dada por:

$$A_2(t) = \begin{bmatrix} 0,707 & \text{sen}(vt) & \text{cos}(vt) \\ 0,707 & \text{cos}(vt) & \text{sen}(vt) \end{bmatrix}$$

15 donde la primera columna corresponde a las ganancias del canal de base (un canal central, C) que se alimenta igualmente de los canales L y R. La segunda columna y la tercera columna, respectivamente, corresponden al canal de objeto U y al canal de objeto V. La primera hilera corresponde al canal L de la mezcla descendente de 2 canales y la segunda hilera corresponde al canal R. Los 2 objetos se mueven el uno hacia el otro a una velocidad determinada por.

Examinaremos las matrices de renderización en 3 instantes de tiempo diferentes t1, t2 y t3. En este ejemplo,

$$A_2(t1) = \begin{bmatrix} 0,707 & 0 & 1 \\ 0,707 & 1 & 0 \end{bmatrix}$$

20 supondremos que t1 = 0, es decir, En otras palabras, en el t1, el objeto U se alimenta completamente de R y el objeto V se mezcla de forma descendente completamente en L. A medida que los objetos se acercan entre sí, su contribución al altavoz más lejano aumenta. Para desarrollar el ejemplo todavía más,

digamos que $v = \frac{\pi}{4} \times \frac{1}{40T}$, donde T es la longitud de una unidad de acceso (típicamente 0,8333 ms o 40 muestras en una tasa de muestreo de 48kHz). De este modo en t = 40T los 2 objetos están en el centro de la escena. Ahora consideramos t2 = 15T y t3 = 30T, de modo que:

$$A_2(t2) = \begin{bmatrix} 0,707 & 0,2903 & 0,9569 \\ 0,707 & 0,9569 & 0,2902 \end{bmatrix}$$

25

$$A_2(t3) = \begin{bmatrix} 0,707 & 0,5556 & 0,8315 \\ 0,707 & 0,8315 & 0,5556 \end{bmatrix}$$

Consideremos la descomposición de la especificación proporcionada A2(t) en matrices primitivas de entrada y salida.

30 En aras de la simplicidad, supongamos que las matrices P_0^2, P_1^2 son matrices de identidad y Assign Ca 0 (en el decodificador 102) es la asignación de canales de identidad, es decir, igual a la permutación trivial (matriz de identidad).

Podemos ver que:

$$\begin{bmatrix} 0,707 & 0 & 1 \\ 0,707 & 1 & 0 \\ 1 & -1,414 & 4,243 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & -0,707 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 0,7070 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1,414 & 4,243 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t1) \quad P_1^{-1}(t1) \quad P_2^{-1}(t1) \quad \text{Assign Ca Inv 1}(t1)$

Las primeras 2 hileras del producto anterior son exactamente la especificación $A_2(t1)$. En otras palabras, las matrices primitivas $P_0^{-1}(t1)$, $P_1^{-1}(t1)$, $P_2^{-1}(t1)$, y la asignación de canales indicada por Asign Ca Inv 1 (t1) en conjunto dan lugar a la transformación del canal de entrada C, el Objeto U y el Objeto V en 3 canales internos, de los cuales los 2 primeros son exactamente las mezclas descendentes L y R requeridas. De este modo, la descomposición anterior

- 5 de $A(t1)$ en las matrices primitivas $P_0^{-1}(t1)$, $P_1^{-1}(t1)$, $P_2^{-1}(t1)$, y la asignación de canales Asign Ca Inv 1 (t1) es una elección válida de matrices primitivas de entrada si las matrices primitivas de salida y la asignación de canales para la presentación de 2 canales se han elegido para ser matrices de identidad. Nótese que las matrices primitivas de entrada son invertibles sin pérdidas para recuperar C, el Objeto U y el Objeto V por un decodificador que opera en los 3 canales internos. Un decodificador de 2 canales, sin embargo, solo necesitaría los canales internos 1 y 2 y
- 10 aplicar las matrices primitivas de salida P_0^2, P_1^2 y Asign Ca 0, que en este caso son todas de identidad.

De forma similar, podemos identificar:

$$\begin{bmatrix} 0,707 & 0,2903 & 0,9569 \\ 0,707 & 0,9569 & 0,2903 \\ 1 & -1,004 & 4,890 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1,666 & 1 & -0,4713 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2,5 & 0,707 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1,003 & 4,889 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t2) \qquad P_1^{-1}(t2) \qquad P_2^{-1}(t2) \qquad \text{Asign Ca Inv 1 (t2)}$

donde las 2 primeras hileras son idénticas a $A(t2)$, y

$$\begin{bmatrix} 0,707 & 0,5556 & 0,8315 \\ 0,707 & 0,8315 & 0,5556 \\ 1 & -0,628 & 7,717 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1,2759 & 1 & -0,1950 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -4,624 & 0,707 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0,628 & 7,717 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$P_0^{-1}(t3) \qquad P_1^{-1}(t3) \qquad P_2^{-1}(t3) \qquad \text{Asign Ca Inv 1 (t3)}$

- 15 donde las primeras 2 hileras son idénticas a $A(t3)$.

Un codificador TrueHD heredado (que no implementa la presente invención) puede optar por transmitir las (inversas de las) matrices primitivas diseñadas más arriba en el t1, t2 y t3, es decir, $\{P_0(t1), P_1(t1), P_2(t1)\}$, $\{P_0(t2), P_1(t2), P_2(t2)\}$, $\{P_0(t3), P_1(t3), P_2(t3)\}$. En este caso, la especificación en cualquier momento t entre t1 y t2 se aproxima por la especificación en $A(t1)$, y entre t2 y t3 se aproxima por $A(t2)$.

- 20 En la realización ejemplar del sistema de la Figura 6, la matriz primitiva $P_0^{-1}(t)$ en $t = t1$, o $t = t2$, o $t = t3$ opera en el mismo canal (canal 2), es decir, la hilera no trivial en los 3 casos es la segunda hilera. Similar es el caso con $P_1^{-1}(t)$ y $P_2^{-1}(t)$. Además, Asign Ca Inv 1 en cada uno de los instantes de tiempo es igual.

- 25 De este modo, para implementar la codificación mediante la realización ejemplar del codificador 100 de la Figura 6), podemos calcular las siguientes matrices delta:

$$\Delta_0(t) = \frac{P_0(t2) - P_0(t1)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0,0222 & 0 & -0,0157 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_1(t) = \frac{P_1(t2) - P_1(t1)}{15} = \begin{bmatrix} 0 & 0,0333 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_2(t1) = \frac{P_2(t2) - P_2(t1)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0,0274 & -0,0431 & 0 \end{bmatrix}$$

y

$$\Delta_0(t2) = \frac{P_0(t3) - P_0(t2)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0,0261 & 0 & -0,0184 \\ 0 & 0 & 0 \end{bmatrix}$$

5

$$\Delta_1(t2) = \frac{P_1(t3) - P_1(t2)}{15} = \begin{bmatrix} 0 & 0,1416 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Delta_2(t2) = \frac{P_2(t3) - P_2(t2)}{15} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0,0250 & -0,01885 & 0 \end{bmatrix}$$

En contraste con el codificador TrueHD heredado, un codificador TrueHD habilitado para la matrización interpolada (la realización ejemplar del codificador 100 de la Figura 6) puede optar por enviar las matrices (primitivas y delta) semilla

$$10 \quad \{P_0(t1), P_1(t1), P_2(t1)\}, \{\Delta_0(t1), \Delta_1(t1), \Delta_2(t1)\}, \{\Delta_0(t2), \Delta_1(t2), \Delta_2(t2)\}.$$

Las matrices primitivas y las matrices delta en cualquier instante de tiempo intermedio se derivan mediante interpolación. Las ecuaciones de mezcla descendente conseguidas en un tiempo dado t entre t1 y t2 se pueden derivar como las primeras 2 hileras del producto:

$$\left(P_0^{-1}(t1) - \Delta_0(t1) * \frac{t}{T} \right) \left(P_1^{-1}(t1) - \Delta_1(t1) * \frac{t}{T} \right) \left(P_2^{-1}(t1) - \Delta_2(t1) * \frac{t}{T} \right) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} y$$

15 entre t2 y t3, como

$$\left(P_0^{-1}(t2) - \Delta_0(t2) * \frac{t}{T} \right) \left(P_1^{-1}(t2) - \Delta_1(t2) * \frac{t}{T} \right) \left(P_2^{-1}(t2) - \Delta_2(t2) * \frac{t}{T} \right) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

En las matrices anteriores, $\{P_0(t2), P_1(t2), P_2(t2)\}$ en realidad no se transmiten, sino que se derivan como las matrices primitivas del último punto de la interpolación con las matrices delta $\{\Delta_0(t1), \Delta_1(t1), \Delta_2(t1)\}$.

20 Conocemos de este modo las ecuaciones de mezcla descendente conseguidas en cada instante "t" para ambos de los escenarios anteriores. Podemos calcular de este modo la falta de correspondencia entre la aproximación en un tiempo dado "t" y la especificación real para dicho instante de tiempo. La Figura 7 es un gráfico de la suma de los errores cuadrados entre la especificación conseguida y la especificación real en diferentes instantes del tiempo t, mediante la utilización de la interpolación de matrices primitivas (la curva denominada "Matrización interpolada") y con las matrices primitivas (no interpoladas) de constante a intervalos (la curva denominada "Matrización no interpolada"). Resulta evidente a partir de la Figura 7 que la matrización interpolada da lugar a la obtención de la especificación $A_2(t)$ significativamente de forma más cercana en comparación con la matrización no interpolada en la región 0-600 s (t1 - t2). Para conseguir el mismo nivel de distorsión con la matrización no interpolada, podría haber sido necesario el envío de actualizaciones de matriz en múltiples puntos entre el t1 y el t2.

30 La matrización no interpolada puede dar lugar a una mezcla descendente conseguida que es más cercana a la especificación real en algunos instantes de tiempo intermedios (por ejemplo, entre 600 s – 900 s en el ejemplo de la Figura 7) pero el error en la matrización no interpolada se acumula continuamente con la reducción del tiempo para la siguiente actualización de matriz mientras que el error con la matrización interpolada disminuye cerca de los

puntos de actualización (en este caso a $t_3 = 30 \cdot T = 1200s$). El error en la matrización interpolada podría reducirse todavía más enviando otra actualización delta entre el t_2 y el t_3 .

Varias realizaciones de la invención implementan una o más de las siguientes características:

- 5 1. transformación de un conjunto de canales de audio a un número igual de otros canales de audio mediante la aplicación de una secuencia de matrices primitivas (preferiblemente, matrices primitivas unitarias) en la que cada una de al menos algunas de las matrices primitivas es una matriz primitiva interpolada calculada como una combinación lineal (determinada de acuerdo con una función de interpolación) de una matriz primitiva semilla y una matriz delta semilla que operan en el mismo canal de audio. El coeficiente de combinación lineal es determinado por la función de interpolación (es decir, cada coeficiente de una matriz primitiva interpolada es una combinación lineal $A + f(t)B$, en la que A es un coeficiente de la matriz primitiva semilla, B es un coeficiente correspondiente de la matriz delta semilla y $f(t)$ es el valor de la función de interpolación en el tiempo, t, asociado con la matriz primitiva interpolada). En algunos casos, la transformación se lleva a cabo en el contenido de audio codificado de un flujo de bits codificado para implementar la recuperación sin pérdidas del contenido de audio que se ha codificado para generar el flujo de bits codificado;
- 10 2. una transformación según la característica 1 anterior, en la que la aplicación de una matriz primitiva interpolada se consigue mediante la aplicación de la matriz primitiva semilla y la matriz delta semilla separadamente en los canales de audio que se van a transformar, y la combinación de forma lineal de las muestras de audio resultantes (por ejemplo, las multiplicaciones de matriz por la matriz primitiva semilla se llevan a cabo en paralelo con las multiplicaciones de matriz por la matriz delta semilla, como en el circuito de la Figura 4);
- 15 3. una transformación según la característica 1 anterior, en la que el factor de interpolación se mantiene sustancialmente constante durante algunos intervalos (por ejemplo, intervalos cortos) de muestras de un flujo de bits codificado, y la matriz primitiva semilla más reciente se actualiza (mediante interpolación) solo durante los intervalos en los que el factor de interpolación cambia (por ejemplo, con el fin de reducir la complejidad del procesamiento en un decodificador);
- 20 4. una transformación según la característica 1 anterior, en la que las matrices primitivas interpoladas son matrices primitivas unitarias. En este caso, la multiplicación por una cascada de matrices primitivas unitarias (en un codificador) seguida por la multiplicación (en un decodificador) por una cascada de sus inversas se puede implementar sin pérdidas con el procesamiento de precisión finita;
- 25 5. una transformación según la característica 1 anterior, en la que la transformación se lleva a cabo en un decodificador de audio que extrae los canales de audio codificados y las matrices semilla codificadas de un flujo de bits codificado, en el que el decodificador está configurado preferiblemente para verificar si el audio (posmatrizado) decodificado se ha determinado correctamente, cotejando una palabra de comprobación derivada del audio posmatrizado con una palabra de comprobación extraída del flujo de bits codificado;
- 30 6. una transformación según la característica 1 anterior, en la que la transformación se lleva a cabo en un decodificador de un sistema de codificación de audio sin pérdidas que extrae los canales de audio codificados y las matrices semilla codificadas de un flujo de bits codificado, y los canales de audio codificados han sido generados por un codificador correspondiente que aplica las matrices primitivas inversas sin pérdidas al audio de entrada, codificando de esta manera el audio de entrada sin pérdidas en el flujo de bits;
- 35 7. una transformación según la característica 1 anterior, en la que la transformación se lleva a cabo en un decodificador que multiplica los canales codificados recibidos por una cascada de matrices primitivas, y solamente un subconjunto de las matrices primitivas se determina mediante la interpolación (es decir, las versiones actualizadas de las otras matrices primitivas se pueden entregar al decodificador de vez en cuando, pero el decodificador no lleva a cabo la interpolación para actualizarlas);
- 40 8. una transformación según la característica 1 anterior, en la que las matrices primitivas semilla, las matrices delta semilla y la función de interpolación se eligen de tal manera que un subconjunto de los canales codificados creado por un codificador se puede transformar mediante operaciones de matrización llevadas a cabo (utilizando las matrices y la función de interpolación) por un decodificador para conseguir mezclas descendentes específicas del audio original codificado por el codificador;
- 45 9. una transformación según la característica 8 anterior, en la que el audio original es un programa de audio basado en objetos y las mezclas descendentes específicas corresponden a la renderización de canales del programa a distribuciones de altavoz estáticas (por ejemplo, estéreo, o un canal 5.1, o un canal 7.1);
- 50 10. una transformación según la característica 9 anterior, en la que los objetos de audio indicados por el programa son dinámicos de modo que las especificaciones de mezcla descendente a una distribución de altavoz estática particular cambien instantáneamente, con el cambio instantáneo acomodado mediante la realización de la matrización interpolada en los canales de audio codificados para crear una presentación de mezcla descendente;
- 55

11. una transformación según la característica 1 anterior, en la que un decodificador habilitado para la interpolación (configurado para llevar a cabo la interpolación de acuerdo con una realización de la invención) es también capaz de decodificar los subflujos de un flujo de bits codificado de conformidad con una sintaxis heredada sin llevar a cabo la interpolación para determinar cualquier matriz interpolada;

5 12. una transformación según la característica 1 anterior, en la que las matrices primitivas están diseñadas para aprovechar la correlación entre canales para conseguir una compresión mejor; y

13. una transformación según la característica 1 anterior, en la que la matrización interpolada se utiliza para conseguir especificaciones de mezcla descendente dinámicas diseñadas para la protección de recorte.

10 Dado que las matrices de mezcla descendente generadas utilizando la interpolación de acuerdo con una realización de la invención (para la recuperación de presentaciones de mezcla descendente de un flujo de bits codificado) cambian típicamente de manera continua cuando el audio fuente es un programa de audio basado en objetos, las matrices primitivas semilla empleadas (es decir, incluidas en el flujo de bits codificado) en realizaciones típicas de la invención necesitan típicamente ser actualizadas con frecuencia para recuperar dichas presentaciones de mezcla descendente.

15 Si las matrices primitivas semilla se actualizan con frecuencia, con el fin de aproximarse de manera cercana a una especificación de matriz continuamente variable, el flujo de bits codificado incluye típicamente datos indicativos de una secuencia de cascadas de conjuntos de matrices primitivas semilla, $\{P_0(t1), P_1(t1), \dots, P_n(t1)\}$, $\{P_0(t2), P_1(t2), \dots, P_n(t2)\}$, $\{P_0(t3), P_1(t3), \dots, P_n(t3)\}$, etc. Esto permite que un decodificador recupere la cascada especificada de matrices en cada uno de los instantes de tiempo de actualización $t1, t2, t3, \dots$ Dado que las matrices de renderización especificadas en los sistemas para la renderización de programas de audio basados en objetos típicamente varían de forma continua en el tiempo, cada matriz primitiva semilla (en una secuencia de cascadas de matrices primitivas semilla incluida en el flujo de bits codificado) puede tener la misma configuración de matriz primitiva (al menos durante un intervalo del programa). Los propios coeficientes en las matrices primitivas pueden cambiar a lo largo del tiempo pero la configuración de matriz no cambia (o no cambia con tanta frecuencia como los

20 coeficientes). La configuración de matriz para cada cascada puede estar determinada por tales parámetros como

1. el número de matrices primitivas en la cascada,
2. el orden de los canales que manipulan,
3. el orden de la magnitud de los coeficientes en ellas,
4. la resolución (en bits) requerida para representar los coeficientes, y
5. las posiciones de los coeficientes que son iguales a cero.

Los parámetros que indican dicha configuración de matriz primitiva pueden permanecer sin cambios durante un intervalo de muchas actualizaciones de matriz semilla. Uno o más de dichos parámetros pueden necesitar ser transmitidos mediante el flujo de bits codificado al decodificador para que el decodificador opere como se desea. Dado que dichos parámetros de configuración pueden no cambiar con tanta frecuencia como las propias actualizaciones de matriz primitiva, en algunas realizaciones la sintaxis de flujo de bits codificado especifica de manera independiente si los parámetros de configuración de matriz se transmiten junto con una actualización a los coeficientes de matriz de un conjunto de matrices semilla. En contraste, en las actualizaciones de matriz de codificación de TrueHD convencional (indicadas por un flujo de bits codificado) son acompañadas necesariamente por actualizaciones de configuración. En las realizaciones contempladas de la invención, el decodificador retiene y utiliza la última información de configuración de matriz recibida si una actualización se recibe solamente para los coeficientes de matriz (es decir, sin una actualización de configuración de matriz).

Aunque está previsto que la matrización interpolada permitirá típicamente una tasa de actualización de matriz semilla baja, se espera que las realizaciones contempladas (en las que una actualización de configuración de matriz puede o puede no acompañar cada actualización de matriz semilla) transmitan de forma eficiente la información de configuración y reduzcan aún más la velocidad de bits requerida para la actualización de las matrices de renderización. En las realizaciones contempladas, los parámetros de configuración pueden incluir parámetros pertinentes para cada matriz primitiva semilla y/o parámetros pertinentes para las matrices delta transmitidas.

Con el fin de minimizar la velocidad de bits transmitida general, el codificador puede implementar una compensación entre la actualización de la configuración de matriz y el gasto de unos pocos bits más en las actualizaciones de coeficientes de matriz mientras que se mantiene la configuración de matriz sin cambios.

La matrización interpolada se puede conseguir transmitiendo la información de pendiente para hacer pasar una matriz primitiva para un canal codificado a otra que opera en el mismo canal. La pendiente se puede transmitir como la tasa de variación de los coeficientes de matriz por unidad de acceso ("AU", por sus siglas en inglés). Si $m1$ y $m2$

son coeficientes de matriz primitiva para los tiempos que se encuentran a K unidades de acceso de distancia, entonces la pendiente para interpolar desde m1 hasta m2 se puede definir como $\text{delta} = (m2 - m1)/K$.

Si los coeficientes m1 y m2 comprenden bits que tienen el siguiente formato: m1 = a,bcdefg y m2 = a,bcuvwxyz, donde ambos coeficientes se especifican con un número específico (que se puede designar como "bits_frac") de los bits de la precisión, entonces la pendiente "delta" se indicaría por un valor de la forma 0,0000mnop (con una precisión más alta y ceros iniciales extra requeridos debido a la especificación de deltas en una base por AU). La precisión adicional requerida para representar la pendiente "delta" se puede definir como "precisión_delta". Si una realización de la invención incluye una etapa de inclusión de cada valor delta directamente en un flujo de bits codificado, el flujo de bits codificado necesitaría incluir valores que tengan un número de bits, "B", que satisface la expresión: $B = \text{bits_frac} + \text{precisión_delta}$. Claramente, no es eficiente transmitir los ceros iniciales después de la cifra decimal. De este modo, en algunas realizaciones, lo que se codifica en el flujo de bits codificado (que se entrega al decodificador) es una delta normalizada (un número entero) que tiene la forma: mnpq, que se representa con bits_delta más un bit de señal. Los valores de bits_delta y precisión_delta se pueden transmitir en el flujo de bits codificado como parte de la información de configuración para las matrices delta. En dichas realizaciones, el decodificador está configurado para derivar la delta requerida en este caso como

$$\text{delta} = (\text{delta normalizada en flujo de bits}) * 2^{-(\text{bits_frac} + \text{precisión_delta})}$$

De este modo, en algunas realizaciones, los valores de interpolación incluidos en el flujo de bits codificado incluyen valores delta normalizados que tienen Y bits de precisión (donde $Y = \text{bits_frac}$), y valores de precisión. Los valores delta normalizados son indicativos de versiones normalizadas de valores delta, en las que los valores delta son indicativos de tasas de variación de los coeficientes de las matrices primitivas, cada uno de los coeficientes de las matrices primitivas tiene Y bits de precisión, y los valores de precisión son indicativos del aumento de precisión (es decir, "precisión_delta") requerida para representar los valores delta relativos a la precisión requerida para representar los coeficientes de las matrices primitivas. Los valores delta se pueden derivar mediante el ajuste a escala los valores delta normalizados por un factor de escala que es dependiente de la resolución de los coeficientes de las matrices primitivas y los valores de precisión.

Las realizaciones de la invención se pueden implementar en hardware, firmware o software, o una combinación de los mismos (por ejemplo, como un orden lógico programable). Por ejemplo, el codificador 40 o el 100, o el decodificador 42 o el 102, o los subsistemas 47, 48, 60 y 61 del decodificador 42, o los subsistemas 110-113 y 106-109 del decodificador 102, se pueden implementar en un hardware o firmware programados adecuadamente (o configurados de otro modo), por ejemplo, como un procesador de fines generales, un procesador de señales digitales o un microprocesador programados. A no ser que se especifique lo contrario, los algoritmos o los procesos incluidos como parte de la invención no están relacionados de forma inherente con ningún ordenador u otro aparato particulares. En particular, se pueden utilizar diversas máquinas de fines generales con programas escritos de acuerdo con las enseñanzas de la presente memoria, o puede ser más conveniente construir aparatos más especializados (por ejemplo, circuitos integrados) para llevar a cabo las etapas de método requeridas. De este modo, la invención se puede implementar en uno o más programas informáticos que se ejecutan en uno o más sistemas informáticos programables (por ejemplo, un sistema informático que implementa el codificador 40 o el 100, o el decodificador 42 o el 102, o el subsistema 47, 48, 60 y/o 61 del decodificador 42, o los subsistemas 110-113 y 106-109 del decodificador 102), donde cada uno comprende al menos un procesador, al menos un sistema de almacenamiento de datos (que incluye una memoria y/o elementos de almacenamiento volátiles y no volátiles), al menos un dispositivo o puerto de entrada y al menos un dispositivo o puerto de salida. El código de programa se aplica a los datos de entrada para llevar a cabo las funciones descritas en la presente memoria y generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

Cada uno de dichos programas se puede implementar en cualquier lenguaje informático deseado (incluidos los lenguajes de programación de máquina, de ensamblaje o procedimentales de alto nivel, lógicos u orientados a objetos) para comunicarse con un sistema informático. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado.

Por ejemplo, cuando se implementan mediante secuencias de instrucción de software informático, varias funciones y etapas de las realizaciones de la invención se pueden implementar mediante secuencias de instrucción de software multiproceso que se ejecuta en un hardware de procesamiento de señales digitales adecuado, en cuyo caso los diversos dispositivos, etapas y funciones de las realizaciones pueden corresponder a porciones de las instrucciones de software.

Cada uno de dichos programas informáticos se almacena preferiblemente en o se descarga a medios o a un dispositivo de almacenamiento (por ejemplo, una memoria o medios de estado sólido, o medios magnéticos u ópticos) legible por un ordenador programable de fines generales o especiales, para la configuración y funcionamiento del ordenador cuando los medios o el dispositivo de almacenamiento se leen por el sistema informático para llevar a cabo los procedimientos descritos en la presente memoria. El sistema inventivo también se puede implementar como un medio de almacenamiento legible por ordenador, configurado con (es decir, que

almacena) un programa informático, en el que el medio de almacenamiento así configurado provoca que un sistema informático opere de una manera específica y predefinida para llevar a cabo las funciones descritas en la presente memoria.

5 Aunque las implementaciones se han descrito a modo de ejemplo y en términos de las realizaciones específicas ejemplares, se ha de entender que las implementaciones de la invención no están limitadas a las realizaciones descritas. Por el contrario, se pretenden cubrir diversas modificaciones y disposiciones similares como serían evidentes para aquellos expertos en la técnica. Por consiguiente, al alcance de las reivindicaciones adjuntas se le debería conceder la interpretación más amplia de manera que englobe todas las modificaciones y disposiciones similares mencionadas.

10

REIVINDICACIONES

1. Un método para la codificación de un programa de audio de N canales, en el que el programa se especifica durante un intervalo de tiempo, el intervalo de tiempo incluye un subintervalo desde un tiempo t_1 hasta un tiempo t_2 , y una mezcla variable en el tiempo, $A(t)$, de N canales de señal codificados a M canales de salida se ha especificado durante el intervalo de tiempo, en el que M es menor que o igual a N, donde dicho método incluye las etapas de:
- 5 determinación de una primera cascada de matrices primitivas de $N \times N$ que, cuando se aplica a las muestras de los N canales de señal codificados, implementa una primera mezcla de contenido de audio de los N canales de señal codificados a los M canales de salida, en la que la primera mezcla es acorde con la mezcla variable en el tiempo, $A(t)$, en el sentido de que la primera mezcla es al menos sustancialmente igual a $A(t_1)$, y en la que una matriz primitiva de $N \times N$ se define como una matriz en la que N-1 hileras contienen elementos fuera de la diagonal iguales a cero y los elementos dentro de la diagonal con un valor absoluto de 1;
- 10 determinación de valores de interpolación que, con la primera cascada de matrices primitivas y una función de interpolación definida durante el subintervalo, son indicativos de una secuencia de cascadas de matrices primitivas actualizadas de $N \times N$, de tal manera que cada una de las cascadas de las matrices primitivas actualizadas, cuando se aplican a las muestras de los N canales de señal codificados, implementan una mezcla actualizada, asociada con un tiempo diferente en el subintervalo, de los N canales de señal codificados a los M canales de salida, en la que cada una de dichas mezclas actualizadas es acorde con la mezcla variable en el tiempo, $A(t)$; y
- 15 generación de un flujo de bits codificado que es indicativo del contenido de audio codificado, los valores de interpolación y la primera cascada de matrices primitivas.
- 20 2. El método de la reivindicación 1, en el que cada una de las matrices primitivas es una matriz primitiva unitaria, donde el método incluye además:
- una etapa de generación del contenido de audio codificado mediante la realización de operaciones de matriz en muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en la que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y la
- 25 secuencia de cascadas de matrices incluye una primera cascada de matrices inversas que es una cascada de las inversas de las matrices primitivas de la primera cascada; o
- una etapa de generación del contenido de audio codificado mediante la realización de operaciones de matriz en muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en la que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y cada
- 30 cascada de matrices en la secuencia es la inversa de una cascada correspondiente de las cascadas de matrices primitivas actualizadas de $N \times N$, y $N = M$, de modo que los M canales de salida sean iguales que los N canales del programa recuperado sin pérdidas.
- 35 3. El método de la reivindicación 1, en el que la primera cascada de matrices primitivas implementa una matriz primitiva semilla, y los valores de interpolación son indicativos de una matriz delta semilla para la matriz primitiva semilla.
4. El método de la reivindicación 2, en el que una mezcla descendente variable en el tiempo, $A_2(t)$, de contenido de audio o contenido codificado del programa a M_1 canales de altavoz también se ha especificado durante el intervalo de tiempo, en el que M_1 es un número entero menor que M, y el método también incluye una etapa de:
- 40 determinación de una segunda cascada de matrices primitivas de $M_1 \times M_1$ que, cuando se aplica a muestras de M_1 canales del contenido de audio codificado en cada instante de tiempo t en el intervalo implementa una mezcla descendente del programa de audio de N canales a los M_1 canales de altavoz, en la que la mezcla descendente es acorde con la mezcla variable en el tiempo, $A_2(t)$.
5. Un método para la recuperación de M canales de un programa de audio de N canales, en el que el programa se especifica durante un intervalo de tiempo, el intervalo de tiempo incluye un subintervalo desde un tiempo t_1 hasta un
- 45 tiempo t_2 , y una mezcla variable en el tiempo, $A(t)$, de N canales de señal codificados a M canales de salida se ha especificado durante el intervalo de tiempo, donde dicho método incluye las etapas de:
- obtención de un flujo de bits codificado que es indicativo de contenido de audio codificado, valores de interpolación y una primera cascada de matrices primitivas de $N \times N$, en la que una matriz primitiva de $N \times N$ se define como una
- 50 matriz en la que N-1 hileras contienen elementos fuera de la diagonal iguales a cero y los elementos dentro de la diagonal con un valor absoluto de 1; y
- realización de la interpolación para determinar una secuencia de cascadas de matrices primitivas actualizadas de $N \times N$, a partir de los valores de interpolación, la primera cascada de matrices primitivas y una función de interpolación durante el subintervalo, en la que

la primera cascada de matrices primitivas de $N \times N$, cuando se aplica a muestras de N canales de señal codificados del contenido de audio codificado, implementa una primera mezcla de contenido de audio de los N canales de señal codificados a los M canales de salida, en la que la primera mezcla es acorde con la mezcla variable en el tiempo, $A(t)$, en el sentido de que la primera mezcla es al menos sustancialmente igual a $A(t_1)$, y

5 los valores de interpolación, con la primera cascada de matrices primitivas, y la función de interpolación, son indicativas de una secuencia de cascadas de matrices primitivas actualizadas de $N \times N$, de tal manera que cada una de las cascadas de las matrices primitivas actualizadas, cuando se aplica a muestras de los N canales de señal codificados del contenido de audio codificado, implementa una mezcla actualizada, asociada con un tiempo diferente en el subintervalo, de los N canales de señal codificados a los M canales de salida, en la que cada una de dichas mezclas actualizadas es acorde con la mezcla variable en el tiempo, $A(t)$.

6. El método de la reivindicación 5, en el que cada una de las matrices primitivas es una matriz primitiva unitaria.

7. El método de la reivindicación 6, en el que:

15 el contenido de audio codificado ha sido generado mediante la realización de operaciones de matriz en muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en la que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y la secuencia de cascadas de matrices incluye una primera cascada de matrices inversas que es una cascada de las inversas de las matrices primitivas de la primera cascada; o

20 el contenido de audio codificado ha sido generado mediante la realización de operaciones de matriz en muestras de los N canales del programa, incluida mediante la aplicación de una secuencia de cascadas de matrices a las muestras, en la que cada cascada de matrices en la secuencia es una cascada de matrices primitivas, y cada cascada de matrices en la secuencia es la inversa de una cascada correspondiente de las cascadas de matrices primitivas actualizadas de $N \times N$, y $N = M$, de modo que los M canales de salida sean iguales que los N canales del programa recuperado sin pérdidas.

25 8. El método de la reivindicación 7, en el que una mezcla descendente variable en el tiempo, $A_2(t)$, de contenido de audio o contenido codificado del programa a M_1 canales de altavoz también se ha especificado durante el intervalo de tiempo, en el que M_1 es un número entero menor que N , y el método también incluye las etapas de:

recepción de una segunda cascada de matrices primitivas de $M_1 \times M_1$; y

30 aplicación de la segunda cascada de matrices primitivas de $M_1 \times M_1$ a muestras de M_1 canales del contenido de audio codificado en cada instante de tiempo t en el intervalo para implementar una mezcla descendente del programa de audio de N canales a los M_1 canales de altavoz, en la que la mezcla descendente es acorde con la mezcla variable en el tiempo, $A_2(t)$.

9. El método de la reivindicación 5, en el que:

el flujo de bits codificado es también indicativo de la función de interpolación; o

35 la primera cascada de matrices primitivas implementa una matriz primitiva semilla, y los valores de interpolación son indicativos de una matriz delta semilla para la matriz primitiva semilla.

10. El método de la reivindicación 5, donde dicho método incluye también una etapa de:

40 aplicación de al menos una de las cascadas de matrices primitivas de $N \times N$ actualizadas a muestras del contenido de audio codificado, incluida mediante la aplicación de una matriz primitiva semilla y una matriz delta semilla separadamente a las muestras del contenido de audio codificado para generar muestras transformadas, y combinación de manera lineal de las muestras transformadas de acuerdo con la función de interpolación, generando de esta manera muestras recuperadas indicativas de muestras de los M canales del programa de audio de N canales.

11. El método de la reivindicación 5, en el que:

45 la función de interpolación es sustancialmente constante durante algunos intervalos del flujo de bits codificado, y cada una de las cascadas actualizadas más recientemente de matrices primitivas actualizadas de $N \times N$ se actualiza mediante interpolación solamente durante un intervalo del flujo de bits codificado en el que la función de interpolación no es sustancialmente constante; o

50 en el que los valores de interpolación incluyen valores delta normalizados representables con Y bits, una indicación de este número de bits de precisión y valores de precisión, en los que los valores delta normalizados son indicativos de versiones normalizadas de valores delta, los valores delta son indicativos de tasas de variación de coeficientes de

las matrices primitivas, y los valores de precisión indicativos de un aumento de precisión requerida para representar los valores delta relativos a la precisión requerida para representar los coeficientes de las matrices primitivas.

5 12. El método de la reivindicación 11, en el que los valores delta se derivan mediante el ajuste a escala de los valores delta normalizados por un factor de escala que es dependiente de la resolución de los coeficientes de las matrices primitivas y los valores de precisión.

13. El método de la reivindicación 5, que también incluye las etapas de:

10 extracción de una palabra de comprobación del flujo de bits codificado, y verificación de si los canales de un segmento del programa de audio han sido recuperados correctamente, cotejando una segunda palabra de comprobación derivada de muestras de audio generadas por dicho subsistema de multiplicación de matriz con la palabra de comprobación extraída del flujo de bits codificado.

15 14. Un codificador de audio configurado para codificar un programa de audio de N canales, en el que el programa se especifica durante un intervalo de tiempo, el intervalo de tiempo incluye un subintervalo desde un tiempo t1 hasta un tiempo t2, y una mezcla variable en el tiempo, A(t), de N canales de señal codificados a M canales de salida se ha especificado durante el intervalo de tiempo, en el que M es menor que o igual a N, y en el que dicho codificador está configurado para llevar a cabo el método de una cualquiera de las reivindicaciones 1 a 4.

20 15. Un decodificador configurado para implementar la recuperación de un programa de audio de N canales, en el que el programa se especifica durante un intervalo de tiempo, el intervalo de tiempo incluye un subintervalo desde un tiempo t1 hasta un tiempo t2, y una mezcla variable en el tiempo, A(t), de N canales de señal codificados a M canales de salida se ha especificado durante el intervalo de tiempo, y en el que dicho decodificador está configurado para llevar a cabo el método de una cualquiera de las reivindicaciones 5 a 13.

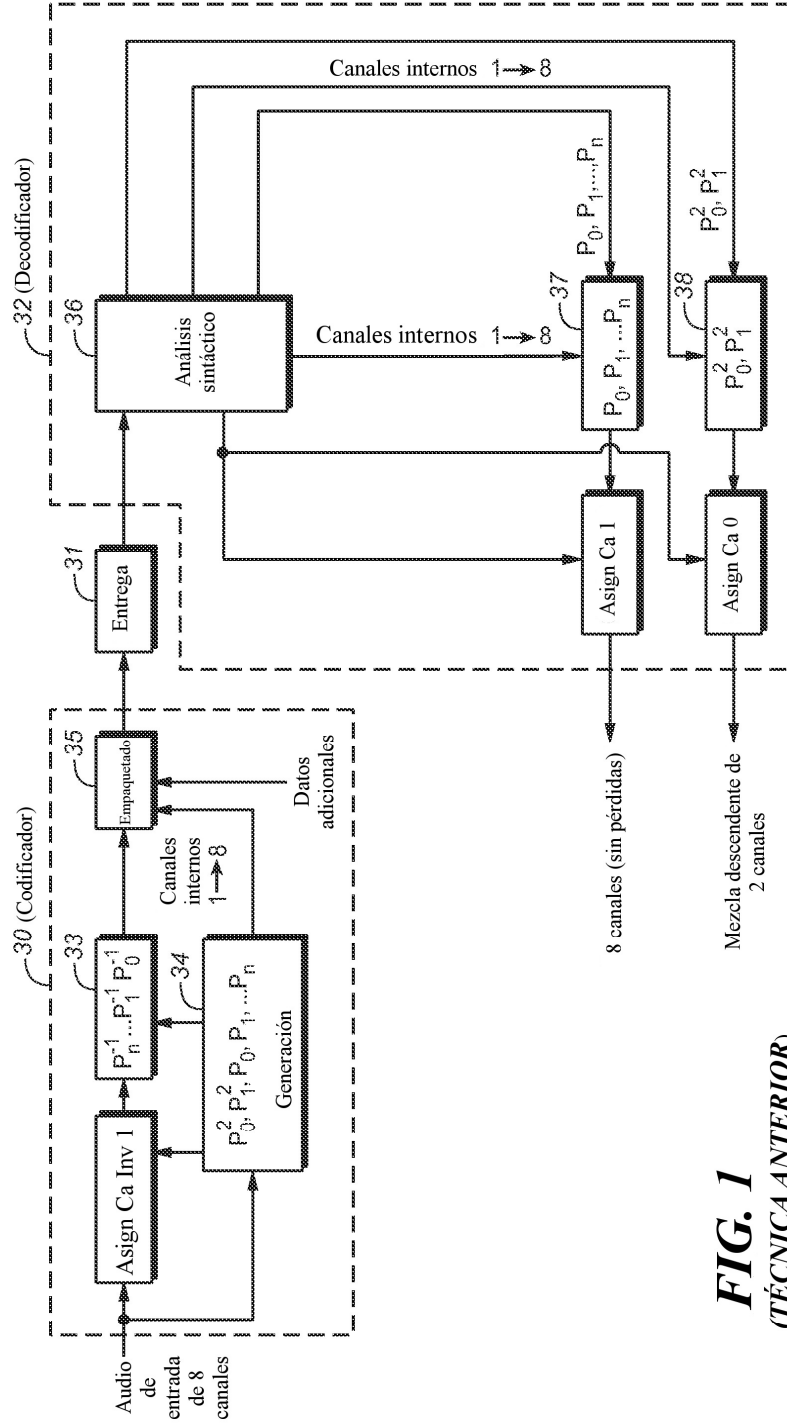


FIG. 1
(TÉCNICA ANTERIOR)

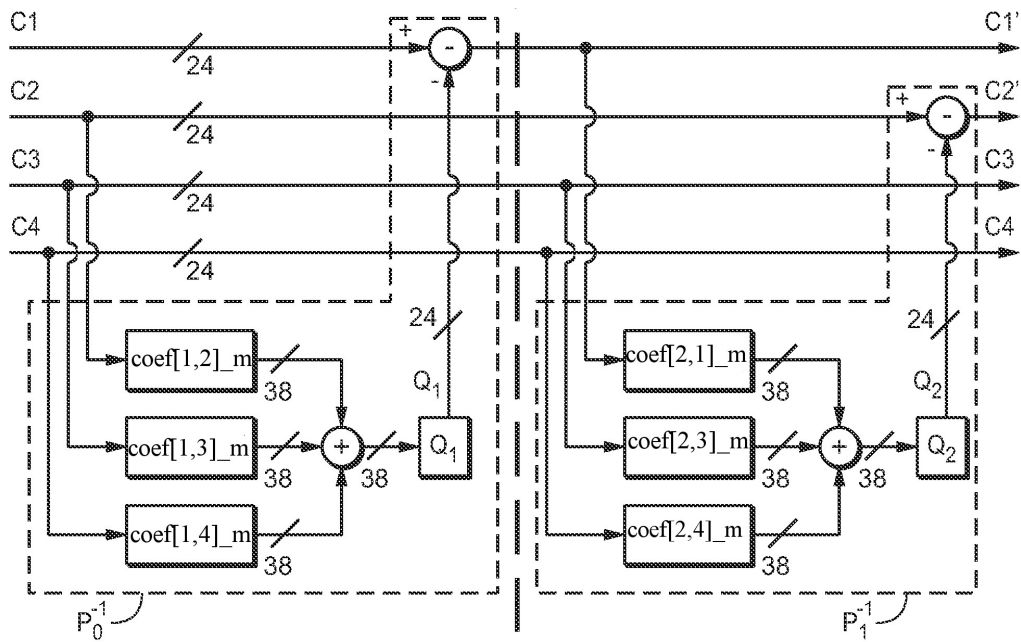


FIG. 2A
(TÉCNICA ANTERIOR)

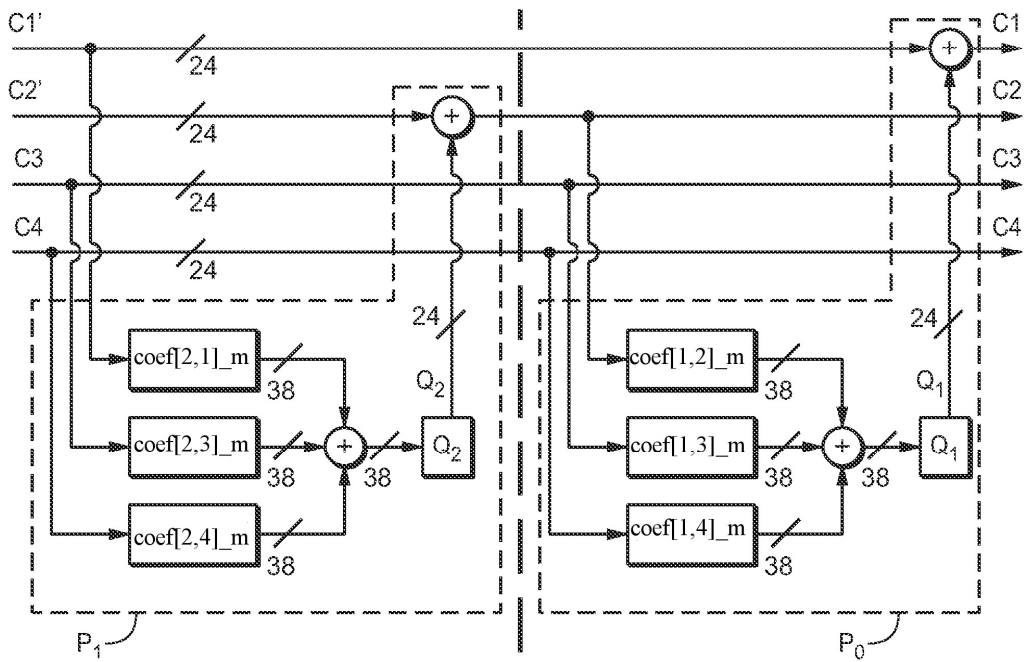


FIG. 2B
(TÉCNICA ANTERIOR)

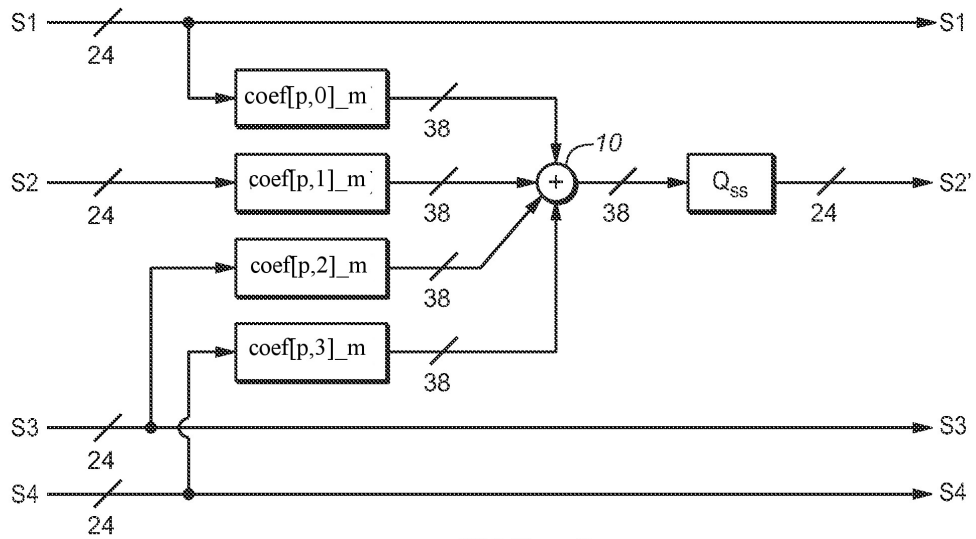


FIG. 3

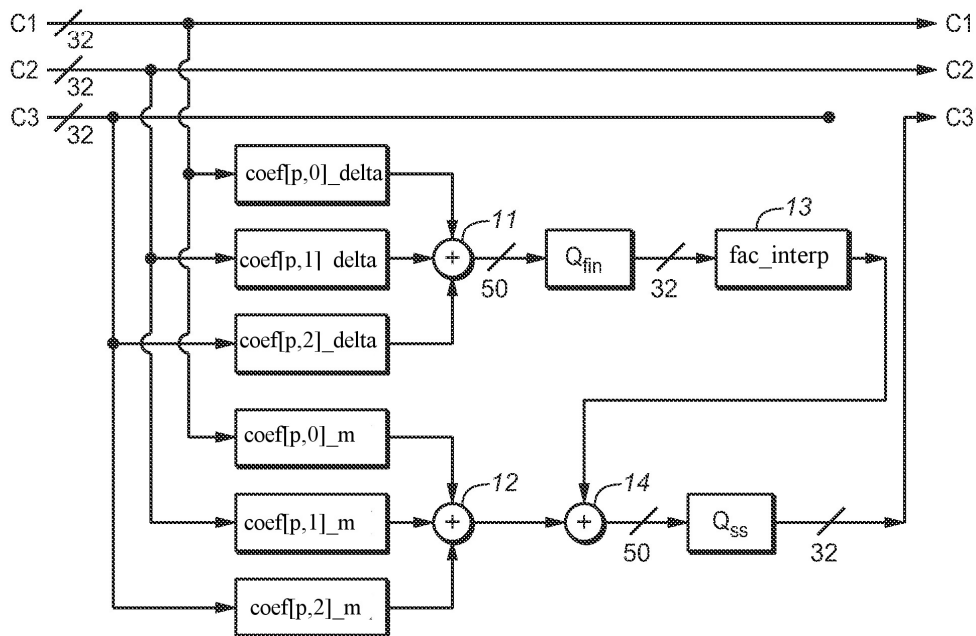


FIG. 4

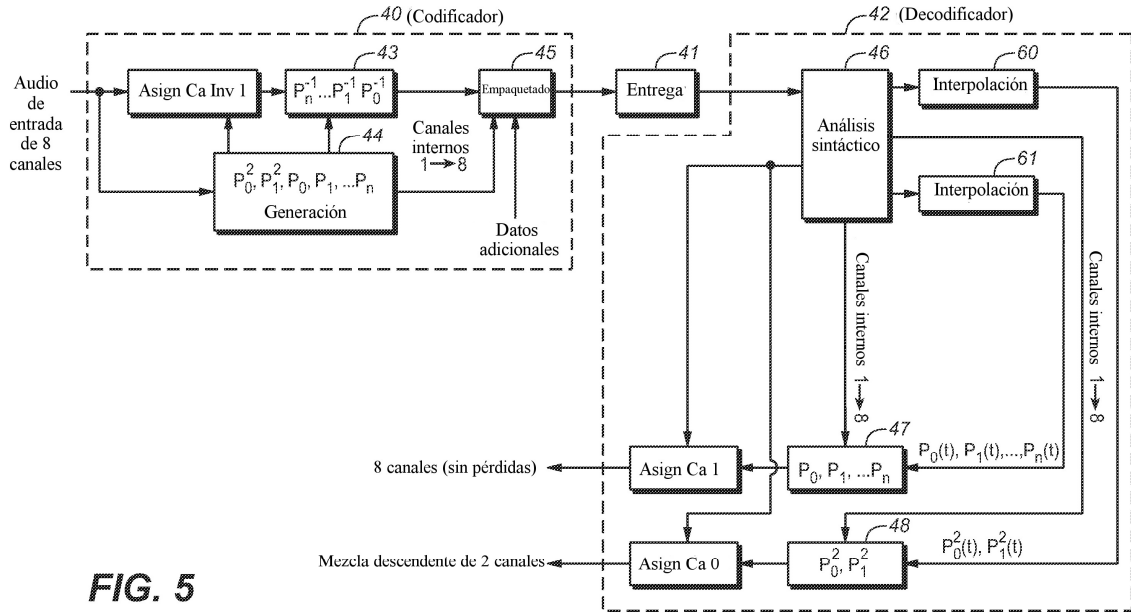


FIG. 5

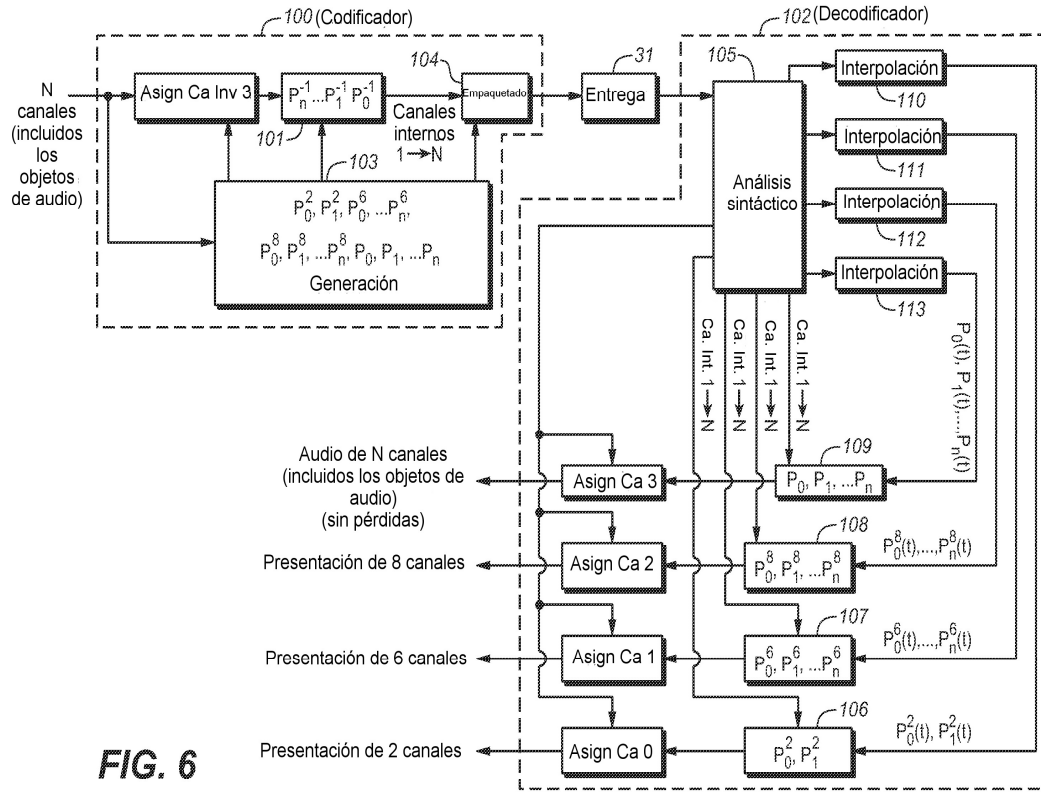


FIG. 6

Distorsión en la aproximación de una especificación de matriz continua

Distorsión en la aproximación de una especificación de matriz continua mediante la interpolación de matrices primitivas o con matrices primitivas de constante a intervalos

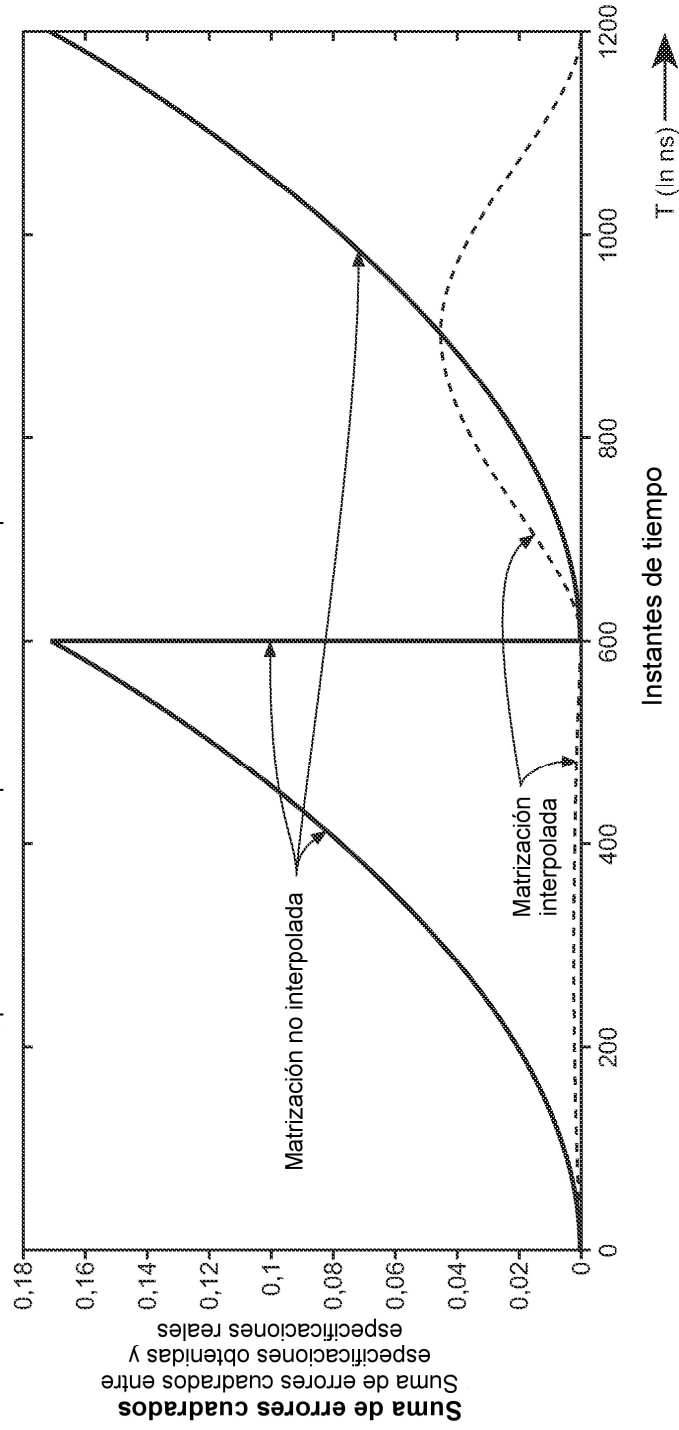


FIG. 7