

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 883 498**

51 Int. Cl.:

H04N 21/81 (2011.01)
H04N 21/439 (2011.01)
H04N 21/442 (2011.01)
H04N 21/4363 (2011.01)
G10L 19/16 (2013.01)
H04N 21/435 (2011.01)
H04N 21/485 (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **27.03.2015** **E 19163481 (5)**

97 Fecha y número de publicación de la concesión europea: **02.12.2020** **EP 3522554**

54 Título: **Procesador de datos y transporte de datos de control del usuario a decodificadores de audio y renderizadores**

30 Prioridad:

28.05.2014 EP 14170416

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
07.12.2021

73 Titular/es:

**FRAUNHOFER-GESELLSCHAFT ZUR
FÖRDERUNG DER ANGEWANDTEN
FORSCHUNG E.V. (100.0%)
Hansastraße 27c
80686 München, DE**

72 Inventor/es:

**SCHREINER, STEPHAN;
NEUKAM, SIMONE;
FUCHS, HARALD;
PLOGSTIES, JAN y
DÖHLA, STEFAN**

74 Agente/Representante:

PONTI & PARTNERS, S.L.P.

ES 2 883 498 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

Procesador de datos y transporte de datos de control del usuario a decodificadores de audio y renderizadores

5 **[0001]** La presente invención se refiere a un procesador de datos de audio de acuerdo con la reivindicación 1, un método para procesar datos de audio de acuerdo con la reivindicación 14 y un programa de ordenador de acuerdo con la reivindicación 15 para ejecutar el método de procesamiento de datos de audio.

10 **[0002]** En instalaciones de aparatos electrónicos para Consumidores domésticos (CE por sus siglas en inglés), la funcionalidad se reparte entre varios dispositivos por medio de interfaces normalizadas. Además, con frecuencia los equipos (de alta calidad) están contruidos no solo formando un único dispositivo, sino que se dispone de sofisticados dispositivos únicos (considérense las Cajas de Conexión (o decodificadores) de televisión, equipos de televisión, receptores AVR). Estos dispositivos se comunican a través de interfaces normalizadas (tales como HDMI).

15 **[0003]** Mientras un primer dispositivo extrae los flujos deseados y ofrece todas las interfaces al usuario, con frecuencia un segundo dispositivo realiza la decodificación en "modo esclavo" sin interfaz alguna para el usuario. Cuando llega el momento de la interacción y control del decodificador por el usuario, es esencial transferir esta información del usuario desde el dispositivo #1 al dispositivo #2 en esta situación.

20 **[0004]** Por ejemplo, como se ilustra en la Figura 9, a menudo un programa de televisión es recibido por un primer dispositivo tal como un receptor digital externo, que selecciona el canal de transmisión apropiado y extrae los flujos elementales pertinentes que contienen la esencia codificada buscada. Estos flujos extraídos pueden ser alimentados a un segundo dispositivo tal como un receptor de audio-vídeo para su reproducción. La transmisión entre estos dos dispositivos se puede realizar transmitiendo una representación decodificada/descomprimida (audio PCM) o en una representación codificada, especialmente si se aplican restricciones de ancho de banda a la línea de interconexión utilizada.

30 **[0005]** Además, como la selección de los flujos deseados y/o, opcionalmente, la interacción con el usuario se realiza en el dispositivo #1 (por ej. el receptor digital externo), en la mayoría de los casos solo este dispositivo ofrece una interfaz de control al usuario. El segundo dispositivo (por ej. el receptor de A/V) solo presenta una interfaz de configuración a la que habitualmente el usuario solo accede una vez al configurar el sistema y actúa en "modo esclavo" durante los períodos de operación normal.

35 **[0006]** Los esquemas modernos de códec de audio no solo admiten la codificación de señales de audio, sino que también ofrecen un medio para la interactividad con el usuario a fin de adaptar la reproducción y renderización del audio a las preferencias del usuario. El flujo de datos de audio consiste en un número de señales de audio codificadas, por ej. señales de canales u objetos de audio, e información de metadatos adjunta que describe cómo estas señales de audio forman una escena de audio que se renderiza a los altavoces.

40 **[0007]** Los ejemplos de objetos de audio son:

- diálogo en diferentes idiomas,
- diálogo adicional como descripción de audio, o
- fondo de música y efectos.

45 **[0008]** Los ejemplos de información de metadatos son:

- el nivel de volumen por defecto de cada señal de objeto (es decir, con qué volumen se debe mezclar para formar una señal mixta para la presentación al altavoz),
- 50 • la posición espacial por defecto (es decir, dónde debe ser renderizada),
- información, si se admite la interacción con el usuario con respecto a un objeto específico, o
- información sobre cómo se permite interactuar al usuario, por ej. niveles de volumen mínimo/máximo o restricciones sobre las posiciones a las cuales el usuario puede redirigir los objetos.
- clasificación y/o descripción de los objetos de audio

55 **[0009]** En la técnica anterior, dicha interactividad con el usuario se puede proporcionar en la parte del codificador, la cual también se puede denominar autoría. Por ejemplo, el documento US-2014/119-581-A1 describe una herramienta de autoría. Un autor, o creador de contenido, puede editar metadatos asociados con los datos de audio. Dicha edición de los metadatos se lleva a cabo en la parte del codificador de tal manera que los metadatos editados se transmiten del codificador al decodificador.

60 **[0010]** El documento Simone Fug y col.: *Object Interaction Use Cases and Technology* 108. MPEG Meeting; 31-3-2014 - 4-4-2014; Valencia; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), núm. m33224, del 27 de marzo de 2014 (2014-03-27), XP030061676 describe una autoría en la parte del codificador.

Describe una posibilidad de interacción con el usuario adicional en la parte del decodificador implementando un procesador de metadatos de objetos en el decodificador.

[0011] El documento Stephan Schreiner y col.: *Proposed MPEG-H 3D Audio stream format*, 108.MPEG Meeting; 31-3-2014-4-4-2014; Valencia; (MOTION PICTURE EXPERT GROUP ORISO/IEC JTC1/SC29/WG11), núm. m33190, del 26 de marzo de 2014 (2014-03-26), XP030061642 describe una posibilidad de embutir cargas útiles de datos de audio codificados y datos de configuración asociados en un flujo de transporte de autosincronización. Para mejorar la sincronización del flujo de transporte se pueden embutir paquetes de flujo de sincronización de manera más frecuente y también se pueden usar paquetes de desfase de sincronización adicionales.

[0012] Para lograr la interacción con el usuario, es necesario que los decodificadores/renderizadores de audio (por ej. el dispositivo #2) presenten una interfaz adicional (de entrada o interacción) para información de control para la interacción pretendida con el usuario.

[0013] Por otro lado, también podría ser conveniente implementar el control por el usuario para la selección de objetos y audio y la manipulación del dispositivo #1 y alimentar estos datos al dispositivo #2 cuando se implementa la decodificación y renderización en el dispositivo #2 y no en el dispositivo #1.

[0014] Sin embargo, la transmisión de esos datos es restringida debido a que las conexiones normalizadas existentes no admiten la transmisión de datos de control por el usuario ni/o información del renderizador. Por otro lado, la selección de flujos y la interacción del usuario antes descritas con respecto al dispositivo #1, y la decodificación antes descrita con respecto al dispositivo #2 pueden ser procesadas por dos componentes funcionales separados contenidos dentro del mismo dispositivo y con las mismas restricciones a la transmisión de datos entre ambos componentes, es decir que solo se dispone de una interfaz para datos codificados y datos de interacción con el usuario, preferentemente la interfaz de interacción del dispositivo #1, y se puede omitir a la vez una segunda interfaz para datos de interacción con el usuario, es decir una interfaz habitualmente provista por el dispositivo #2. Aunque tanto el dispositivo #1 como el dispositivo #2 están contenidos o implementados dentro del mismo dispositivo (*hardware*), esto lleva a la misma situación descrita respecto al caso de los dispositivos #1 y #2.

[0015] Para obtener el caso de uso descrito y a fin de superar las limitaciones antes descritas, se propone embutir los datos de información de control del usuario, o los datos de interacción en general, en el flujo de datos de audio codificados.

[0016] En consecuencia, un objetivo de la presente invención es mejorar los procesadores de datos de audio existentes.

[0017] En términos generales, el primer dispositivo puede ser configurado en forma de procesador de datos de audio, que comprende: una interfaz de receptor para recibir datos de audio codificados y metadatos relacionados con los datos de audio codificados; un analizador de metadatos para analizar los metadatos para determinar una posibilidad de manipulación de datos de audio; una interfaz de interacción para recibir una entrada de interacción y para generar, a partir de la entrada de interacción, datos de control de la interacción relacionados con la posibilidad de manipulación de datos de audio; y un generador de flujos de datos para obtener los datos de control de la interacción y los datos de audio codificados y los metadatos y para generar un flujo de datos de salida, comprendiendo el flujo de datos de salida los datos de audio codificados, por lo menos una porción de los metadatos, y los datos de control de la interacción de acuerdo con lo definido en la reivindicación 1. Se definen otras formas de realización preferidas en las reivindicaciones dependientes y otras independientes que se adjuntan.

[0018] Los datos de audio codificados pueden comprender objetos de audio codificados separados, donde por lo menos una porción de los metadatos está asociada a un objeto de audio correspondiente, donde el analizador de metadatos está configurado para analizar la porción correspondiente a los objetos de audio codificados a fin de determinar, respecto a por lo menos un objeto de audio, la posibilidad de manipulación de objetos, donde la interfaz de interacción está configurada para generar, respecto a dicho por lo menos un objeto de audio codificado, los datos de control de la interacción a partir de la entrada de interacción relacionada con dicho por lo menos un objeto de audio codificado. De esa manera, los objetos de audio pueden ser fácil y directamente manipulados dentro de sus correspondientes posibilidades de manipulación de objetos almacenadas dentro de los metadatos mediante el uso de los respectivos datos de control de interacción.

[0019] La interfaz de interacción puede estar configurada para presentar, a un usuario, la posibilidad de manipulación de datos de audio derivada de los metadatos por el analizador de metadatos, y para recibir, del usuario una entrada del usuario con respecto a la manipulación de datos específica de la posibilidad de manipulación de datos. Esto puede representar una manera práctica de ofrecer una interfaz del usuario a un usuario para interactuar con el dispositivo de la invención, por ej. para manipular objetos de audio, preferentemente fuera de un decodificador.

[0020] El generador de flujos de datos puede estar configurado para procesar un flujo de datos que comprende los datos de audio codificados y los metadatos recibidos por la interfaz de receptor sin decodificar los datos de audio

codificados, o para copiar los datos de audio codificados y por lo menos una porción de los metadatos sin cambios en el flujo de datos de salida, donde el generador de flujos de datos está configurado para agregar una porción adicional de datos que contiene los datos de control de la interacción a los datos de audio codificados y/o los metadatos contenidos en el flujo de datos de salida. Esto otorga la ventaja de una menor complejidad, ya que no es necesario
 5 que el procesador de datos de audio decodifique las señales de audio. Solo necesita analizar los metadatos y reescribirlos en la parte de metadatos del flujo de datos de audio codificados.

[0021] El generador de flujos de datos puede estar configurado para generar, en el flujo de datos de salida, los datos de control de la interacción en el mismo formato que los metadatos. De esa manera, se puede integrar
 10 ventajosamente cualquier dato de control de la interacción al flujo de datos de salida.

[0022] El generador de flujos de datos puede estar configurado para asociar, con los datos de control de la interacción, un identificador en el flujo de datos de salida, identificador que es diferente de un identificador asociado a los metadatos. La ventaja de utilizar un identificador diferente para los metadatos manipulados es que se podría
 15 habilitar un decodificador remoto para identificar la interacción en virtud del flujo de datos manipulados recibido recibiendo también, a la vez, los datos originales.

[0023] El generador de flujos de datos puede estar configurado para agregar, a los datos de control de la interacción, datos de verificación de firma, indicando los datos de verificación de firma información sobre una
 20 aplicación, un dispositivo o un usuario que realiza una interacción, por ej. una manipulación de datos de audio o el otorgamiento de la entrada del usuario. Mediante el transporte de los datos originales y los manipulados, es posible un reinicio de los metadatos. Una verificación de firma en los metadatos permite rastrear el origen de la manipulación.

[0024] El analizador de metadatos puede estar configurado para identificar una posibilidad de deshabilitación
 25 con respecto a uno o más objetos de audio representados por los datos de audio codificados, donde la interfaz de interacción está configurada para recibir una información de deshabilitación correspondiente a dicho uno o más objetos de audio, y donde el generador de flujos de datos está configurado para marcar dichos uno o más objetos de audio como deshabilitados en los datos de control de la interacción o para eliminar dichos uno o más objetos de audio deshabilitados de los datos de audio codificados de manera que el flujo de datos de salida no incluya datos de audio
 30 codificados correspondientes a dichos uno o más objetos de audio deshabilitados. Por consiguiente, el flujo de datos se puede adaptar a los objetos de audio que estén disponibles en realidad o en el momento, de tal manera que se pueda reducir el contenido de datos total de un flujo de bits en curso.

[0025] El generador de flujos de datos puede estar configurado para generar de manera dinámica el flujo de
 35 datos de salida, donde en respuesta a una nueva entrada de interacción, los datos de control de la interacción se actualicen para coincidir con la nueva entrada de interacción, y donde el generador de flujos de datos está configurado para incluir los datos de control de interacción actualizados en el flujo de datos de salida. De esa manera, se puede enviar un flujo de datos con información en tiempo real. Dicho de otro modo, se puede actualizar y procesar la entrada de interacción referente a cualquier valor de objeto de audio específico de manera expeditiva, preferentemente en
 40 tiempo real.

[0026] La interfaz de receptor puede estar configurada para recibir un flujo principal de datos de audio que comprende los datos de audio codificados y metadatos relacionados con los datos de audio codificados, y para recibir asimismo datos de audio opcionales que comprenden un objeto de audio opcional, donde los metadatos relacionados
 45 con dicho objeto de audio opcional están contenidos en dicho flujo de datos de audio principal. Con esta configuración, el procesador de datos de audio puede fusionar los datos de audio codificados del objeto de audio opcional seleccionado con el flujo principal de datos de audio para dar lugar a un flujo de datos de audio completo de salida generado por el generador de flujos de datos. De esa manera, posteriormente o bajo demanda también se pueden presentar a un usuario objetos de audio opcionales.
 50

[0027] El analizador de metadatos puede estar configurado para determinar la posibilidad de manipulación de audio correspondiente a un objeto de audio faltante que no está incluido en los datos de audio codificados, donde la interfaz de interacción está configurada para recibir una entrada de interacción correspondiente al objeto de audio faltante, y donde la interfaz de receptor está configurada para solicitar datos de audio correspondientes al objeto de audio
 55 faltante a un proveedor de datos de audio o para recibir los datos de audio correspondientes al objeto de audio faltante de un subflujo diferente contenido en un flujo de transmisión o una conexión por protocolo de internet. De esa manera, un dispositivo o un usuario puede manipular de antemano un objeto de audio adicional que opcionalmente está disponible, es decir, cuando falta en realidad. A continuación, se puede solicitar, entonces, el objeto de audio adicional, a través de internet u otro flujo de transmisión.
 60

[0028] El generador de flujos de datos puede estar configurado para asignar, en el flujo de datos de salida, un tipo de paquetes adicional a los datos de control de la interacción, siendo el tipo de paquetes adicional diferente de los tipos de paquetes correspondientes a los datos de audio codificados y los metadatos, o donde el generador de flujos de datos está configurado para agregar, incorporándolos al flujo de datos de salida, datos de relleno de un tipo
 65 de paquetes de datos de relleno, donde una cantidad de datos de relleno se determina sobre la base de la demanda

de tasa de datos determinada por una interfaz de salida del procesador de datos de audio. De esa manera, solo es necesario asignar un tipo adicional de paquetes para lograr el transporte de los metadatos o datos de control de la interacción manipulados, respectivamente. Además, puede ser necesario que el procesador de datos de audio agregue datos de relleno adicionales a un flujo de transmisión de datos subsiguiente para cumplir con el requisito de

5 tasa de datos normalmente más elevada correspondiente a ese vínculo. Estos datos de relleno pueden no contener información alguna y lo más probable es que sean ignorados por el decodificador.

[0029] El procesador de datos de audio puede ser implementado en forma de dispositivo separado, donde la interfaz de receptor puede conformar una entrada al dispositivo separado a través de una conexión por cable o

10 inalámbrica, donde el procesador de datos de audio puede comprender además una interfaz de salida conectada al generador de flujos de datos, estando la interfaz de salida configurada para emitir el flujo de datos de salida, donde la interfaz de salida ejecuta una salida del dispositivo y comprende una interfaz inalámbrica o un conector de cable. De esa manera, se puede proporcionar una conectividad sencilla, por ejemplo, dentro de una red.

15 **[0030]** La presente invención también se puede realizar mediante un método para procesar datos de audio, método que comprende: recibir datos de audio codificados y metadatos relacionados con los datos de audio codificados; analizar los metadatos para determinar una posibilidad de manipulación de datos de audio; recibir una entrada de interacción y generar, a partir de la entrada de interacción, datos de control de la interacción relacionados con la posibilidad de manipulación de datos de audio; y obtener los datos de control de la interacción y los datos de

20 audio codificados y los metadatos y generar un flujo de datos de salida, comprendiendo el flujo de datos de salida los datos de audio codificados, por lo menos una porción de los metadatos y los datos de control de la interacción.

[0031] La presente invención se puede realizar además mediante un programa de ordenador para llevar a cabo, al ejecutarse en un ordenador o un procesador, el método de procesamiento de datos de audio antes

25 mencionado.

[0032] La presente invención se puede realizar además por medio de las siguientes formas de realización: La posibilidad de manipulación de datos de audio puede ser seleccionada de un grupo que comprende por lo menos una de una selección de un objeto, una selección entre varios idiomas, una selección de objetos de audio adicionales

30 opcionales, una manipulación de objetos, un cambio de volumen de uno o más objetos, un cambio de posición de los objetos, como por ejemplo mover un comentario adicional de un altavoz central a un altavoz derecho o a una posición arbitraria entre ambos, una selección de opciones preestablecidas en lugar de la selección y manipulación de cada objeto por separado, donde se selecciona una opción preestablecida de los metadatos, donde una opción preestablecida es una preselección de objetos recomendados por un creador de contenidos para una aplicación

35 específica o una situación de uso específica, donde una opción preestablecida contiene una combinación de objetos con, por ejemplo, diferentes niveles de volumen, posiciones y datos de compresión de volumen/rango dinámico en comparación con una presentación por defecto.

[0033] El generador de flujos de datos puede estar configurado para generar los datos de control de la interacción en forma de información independiente o como información dependiente, donde la información dependiente depende de los metadatos y da lugar, si se la aplica a los datos de audio decodificados, junto con los metadatos, a una manipulación de datos definida por la entrada de interacción.

40

[0034] Los datos de audio codificados pueden comprender objetos de audio opcionales y los metadatos pueden comprender metadatos correspondientes a los objetos de audio opcionales, donde la interfaz de receptor puede estar configurado para recibir además un flujo principal de datos de audio que consta de datos de audio principales, donde el generador de flujos de datos puede estar configurado para generar el flujo de datos de salida de tal manera que el flujo de datos de salida comprenda a su vez los datos de audio principales.

45

50 **[0035]** El generador de flujos de datos puede estar configurado para agregar datos de protección contra errores al flujo de datos de salida y para asignar otro tipo de paquetes a los datos de protección contra errores, donde el generador de flujos de datos está configurado para derivar los datos de protección contra errores de los datos de audio codificados, los metadatos o los datos de control de interacción.

55 **[0036]** El generador de flujos de datos puede estar configurado para generar el flujo de datos de salida en forma de flujo de datos para el flujo continuo o en forma de archivo basado en contenedores en un formato de archivo tal como el formato de archivos ISO MPEG-4.

[0037] También se sugiere que el procesador de datos de audio no tenga una funcionalidad de decodificar los

60 datos de audio codificados.

[0038] El procesador de datos de audio puede ser implementado en un receptor digital externo, un aparato de televisión o un grabador-receptor de audio/vídeo.

65 **[0039]** El procesador de datos de audio puede comprender además una interfaz de salida para transmitir el

flujo de datos de salida a otro dispositivo a través de una conexión HDMI.

[0040] El procesador de datos de audio se puede proporcionar, es decir, integrar o implementar, junto con un decodificador dentro del mismo dispositivo (*hardware*). Por ejemplo, se puede incluir el procesador de datos de audio y un decodificador juntos dentro de un televisor, un receptor digital externo, un receptor de A/V, o similar. El procesador de datos de audio y el decodificador se pueden comunicar por medio de estructuras de buses de datos internos. Esa configuración puede ser deseada especialmente en dispositivos de TV que comprenden soluciones de Sistemas basados en Chips (del inglés System-on-Chip (SoC)).

[0041] En consecuencia, o por otra parte, el procesador de datos de audio puede ser implementado en forma de componente funcional independiente y separado en el mismo dispositivo, de manera similar al caso descrito anteriormente con respecto al caso de un dispositivo separado, donde la única diferencia reside en que la interfaz de salida ejecuta una salida del procesador de datos de audio en una conexión interna del dispositivo, por ejemplo utilizando un bus de datos interno.

[0042] Con respecto a las características antes citadas, el procesador de datos de audio de acuerdo con la invención puede ofrecer una fácil interacción con un dispositivo o un usuario sumado, al mismo tiempo, a una configuración de dispositivo sencilla, preferentemente empleando las instalaciones existentes.

[0043] Más aun, el procesador de datos de audio de acuerdo con la invención otorga una solución al problema antes citado incluyendo una interacción con dispositivos o interacción con el usuario como datos de interacción adicionales dentro del flujo de bits de audio. Implementando las características precedentemente descritas, las implementaciones del decodificador pueden requerir solo una interfaz que tome tanto los datos de representación codificados como los datos de control de interacción. Puede no ser necesario que las interconexiones ya existentes implementen nuevos canales para la información de control, sino que el esfuerzo de implementación se concentra en el códec en sí. En configuraciones complejas, también se garantiza que la información de control de interacción esté estrechamente vinculada a la esencia codificada y, por lo tanto, no se pueda perder al alimentarla a través de varias etapas de procesamiento.

[0044] En los dibujos se exponen ejemplos de las formas de realización de acuerdo con la presente invención y se describirán a continuación, donde:

La Fig. 1 ilustra un procesador de datos de audio de acuerdo con la presente invención,

La Fig. 2 ilustra un método para procesar datos de audio de acuerdo con la invención,

La Fig. 3 ilustra un ejemplo de datos de audio codificados y metadatos relacionados,

La Fig. 4 ilustra un ejemplo de procesamiento de flujos de entrada y salida,

La Fig. 5 ilustra otro ejemplo de procesamiento de flujos de entrada y salida,

La Fig. 6 ilustra un procesador de datos de audio que procesa datos de audio opcionales,

La Fig. 7 ilustra un procesador de datos de audio que ha sido implementado en forma de dispositivo separado,

La Fig. 8 ilustra un ejemplo de caso de uso con un primer dispositivo y un segundo dispositivo, y

La Fig. 9 ilustra un ejemplo de situación con un receptor digital externo y un receptor de audio-vídeo.

[0045] En la totalidad de este documento, y en particular en la siguiente descripción, se utiliza el término "interacción" en el sentido de una interacción realizada por un usuario o una interacción realizada por un dispositivo, así como una interacción en general, es decir, una interacción en el sentido habitual. Dicho de otro modo, "interacción" se puede referir a una "interacción del usuario" o a una "interacción del dispositivo", o a una interacción en general. En ciertas partes de la descripción, los términos "usuario" e "interacción" se utilizan como sinónimos. Por ejemplo, una interfaz del usuario se puede utilizar de manera equivalente en el sentido de una interfaz de interacción y viceversa.

[0046] Más aun, un "usuario" puede ser un usuario humano o un usuario máquina, como por ejemplo un dispositivo (*hardware*) o un dispositivo implementado por software.

[0047] Además, la interfaz del usuario puede estar presente en una configuración preestablecida específica para el dispositivo que, de forma exclusiva o además de la entrada del usuario, puede controlar la manipulación de datos.

[0048] La Figura 1 ilustra un procesador de datos de audio 1 de acuerdo con la presente invención. El

procesador de datos de audio 1 comprende una interfaz de receptor 2 para recibir un flujo de entrada codificado 15 que comprende datos de audio codificados 3 y metadatos 4. Los metadatos 4 están asociados a los datos de audio codificados 3, relación que se indica por medio de la flecha 110. Por ejemplo, los datos de audio codificados 3 pueden contener objetos de audio en tanto que los metadatos 4 pueden contener información adicional sobre las posibilidades de manipulación de dichos objetos de audio.

[0049] El procesador de datos de audio 1 comprende además un analizador de metadatos 5 para analizar los metadatos 4 a fin de determinar una posibilidad de manipulación de datos de audio. Por ejemplo, un nivel de volumen ajustable, una posición espacial ajustable o un idioma seleccionable puede representar una posibilidad de manipulación de datos de audio de un objeto de audio.

[0050] Más aun, el procesador de datos de audio 1 comprende una interfaz de interacción 6 para recibir una entrada de interacción 7. La interfaz de interacción 6 está configurada a su vez para generar datos de control de la interacción 8 sobre la base de la entrada de interacción 7. Dichos datos de control de la interacción 8 están asociados a la posibilidad de manipulación de datos de audio antes citada. Por ejemplo, un usuario puede interactuar con el dispositivo ajustando el nivel de volumen o la posición espacial de un objeto de audio, o seleccionando un idioma por medio de la interfaz de interacción 6. En este caso, la interfaz de interacción 6 es una interfaz del usuario 6 que puede generar datos de control por el usuario 8 correspondientes, es decir, relacionados con la elección del usuario.

[0051] Además, o por otro lado, la interfaz de interacción 6 puede ser una interfaz de interacción (específica) de un dispositivo 6. En este caso, la interfaz de interacción de un dispositivo 6 está configurada para generar datos de control de interacción del dispositivo 8 sobre la base de la entrada de interacción del dispositivo 7. Por ejemplo, podría haber un dispositivo tal como un auricular o similar conectado a la interfaz de interacción 6. La conexión entre el auricular y la interfaz de interacción 6 puede ser detectada por el procesador de audio y, por consiguiente, se la puede considerar como entrada de interacción 7. De esa manera, al conectarse el auricular, la interfaz de interacción 6 emite datos de control de interacción específicos para auriculares 8, como por ejemplo la manipulación de objetos de audio, por ej. una reducción automática del volumen, un idioma preseleccionado o un ajuste de la configuración de *hardware*.

[0052] Dicho de otro modo, en lugar de una interacción manual del usuario, la interfaz de interacción 6 selecciona automáticamente los objetos o los ajustes sobre la base de la detección de ciertos dispositivos. La interfaz de interacción 6 genera datos de control de interacción específicos del dispositivo 8.

[0053] El procesador de datos de audio 1 comprende además un generador de flujos de datos 9. El generador de flujos de datos 9 obtiene los datos de control de la interacción 8, los datos de audio codificados 3 y los metadatos 4. El generador de flujos de datos 9 está configurado para generar un flujo de datos de salida 10 que comprende los datos de control de la interacción 8, los datos de audio codificados 3 y los metadatos 4 antes citados.

[0054] La Figura 2 ilustra un método correspondiente para procesar datos de audio de acuerdo con la presente invención.

[0055] En el paso 201 se reciben datos de audio codificados 3 y metadatos relacionados 4.

[0056] En el paso 202 se analizan los metadatos 4 para determinar una posibilidad de manipulación de datos de audio.

[0057] En el paso 203 se recibe una entrada de interacción, donde en el paso 204 se generan datos de control de la interacción relacionados con la posibilidad de manipulación de datos de audio a partir de dicha entrada de interacción.

[0058] En el paso 205 se obtienen los datos de control de la interacción y los datos de audio codificados y los metadatos, y se genera un flujo de datos de salida, donde dicho flujo de datos de salida comprende los datos de audio codificados, por lo menos una porción de los metadatos y los datos de control de la interacción.

[0059] Con referencia a la Figura 3, los datos de audio codificados 3 comprenden objetos de audio separados codificados 11, 12. Más aun, por lo menos una porción 13, 14 de los metadatos 4 está asociada (según lo indican las flechas 110, 120) a un objeto de audio 11, 12 correspondiente. Por ejemplo, la porción 'MD1' 13 de los metadatos 4 está asociada al objeto de audio codificado correspondiente 'AO1' 11, en tanto que la porción 'MD2' 14 de los metadatos 4 está asociada a un objeto de audio codificado correspondiente 'AO2' 12.

[0060] El analizador de metadatos 5 está configurado para analizar la porción correspondiente 13, 14 a los objetos de audio codificados 11, 12 a fin de determinar la posibilidad de manipulación de objetos de por lo menos uno de dichos objetos de audio 11, 12. Dicho de otro modo, el analizador de metadatos 5 analiza los metadatos 13, 14 correspondientes a los respectivos objetos de audio 11, 12 para determinar la posibilidad de manipulación de objetos de audio para cada objeto de audio 11, 12. Por ejemplo, el analizador de metadatos 5 determina que el objeto de audio 'AO1' 11 puede comprender un nivel de volumen ajustable. El analizador de metadatos 5 puede presentar esta

capacidad de ajuste del nivel de volumen (posibilidad de manipulación de objetos de audio) a un usuario a través de la interfaz del usuario 6.

[0061] La interfaz del usuario 6 está configurada para generar, respecto a por lo menos un objeto de audio 11, 12, los datos de control del usuario 8 a partir de la entrada del usuario 7 relacionada con dicho por lo menos un objeto de audio codificado 11, 12. Por ejemplo, el usuario puede desear ajustar el nivel de volumen del objeto de audio 'AO1' 11 y por ello produce la entrada respectiva 7 a través de la interfaz del usuario 6. La interfaz del usuario 6 genera respectivos datos de control por el usuario 8 que contienen la información de que el usuario desea ajustar el nivel de volumen del objeto de audio 'AO1' 11 y en qué medida desea hacerlo.

[0062] En consecuencia, la interfaz del usuario 6 está configurada para presentar a un usuario la posibilidad de manipulación de objetos de audio de un objeto de audio 11, 12 derivada de los metadatos 4 por el analizador de metadatos 5. La interfaz del usuario 6 está configurada a su vez para recibir una entrada de usuario 7 del usuario con respecto a la manipulación de datos específica (por ej. un nivel de volumen específico o un idioma específico) de la posibilidad de manipulación de datos (por ej. un rango de ajuste del nivel de volumen o una serie de idiomas disponibles).

[0063] Con referencia a la Figura 1 y a la Figura 9, el generador de flujos de datos 9 está configurado para procesar un flujo de datos 15 que comprende los datos de audio codificados 3 y los metadatos 4 recibidos por la interfaz de receptor 2 sin decodificar los datos de audio codificados 3. Por ejemplo, suponiendo que un procesador de datos de audio 1 de acuerdo con la invención está implementado en un receptor digital externo 19, 26, este puede enviar un flujo de datos de salida 10, 32 a un receptor de audio-vídeo externo 28, 33 que comprende un decodificador. En este caso, el flujo de datos de salida 10, 32 aún puede estar codificado, ya que la decodificación no será ejecutada por el receptor digital externo 19, 26 sino por el receptor de audio-vídeo 28, 33.

[0064] Por otro lado, el generador de flujos de datos 9 está configurado para copiar los datos de audio codificados 3 y los metadatos 4 sin cambios en el flujo de datos de salida 10.

[0065] En todo caso, el generador de flujos de datos 9 está configurado para agregar una porción adicional de datos que contiene los datos de control de la interacción 8 a los datos de audio codificados 3 y/o a los metadatos 4 en el flujo de datos de salida 10, como se puede apreciar en la Figura 4.

[0066] Con referencia a la Figura 5, el generador de flujos de datos 9 está configurado a su vez para fusionar dos flujos de entrada 15a, 15b para formar un flujo de salida común 10, donde se agrega una porción adicional de datos que contiene los datos de control de la interacción 8 a los datos de audio codificados 3 y/o a los metadatos 4 en el flujo de datos de salida 10.

[0067] Preferentemente, el generador de flujos de datos 9 está configurado para generar, en el flujo de datos de salida 10, los datos de control de la interacción 8 en el mismo formato que los metadatos 4. De esa manera, se pueden combinar fácilmente los datos de control de la interacción 8 con los metadatos existentes 4.

[0068] Si, como se mencionó anteriormente, el generador de flujos de datos 9 copia los metadatos 4, los metadatos originales 4 pueden permanecer en el flujo de salida 10 además de todo metadato manipulado que contiene datos de control de interacción 8 adicionales correspondientes a respectivos objetos de audio 11, 12. Tanto los metadatos originales como los manipulados pueden ser enviados a un decodificador 28, 33 para dar lugar a que el decodificador 28, 33 identifique las diferencias como resultado de la interacción (usuario) y obtenga toda la información acerca de los valores por defecto según la intención del creador de contenidos, o calcular el resultado de la interacción (usuario) de los metadatos originales 4 y los metadatos manipulados 4' (o los datos de control de la interacción 8).

[0069] También con referencia a la Figura 1, el generador de flujos de datos 9 está configurado para generar de manera dinámica el flujo de datos de salida 10. Cada vez que un usuario o un dispositivo presenta una nueva entrada de interacción 7 a la interfaz de interacción 6, los datos de control de la interacción 8 se actualizan de manera acorde para coincidir con dicha nueva entrada de interacción 7. El generador de flujos de datos 9 incluye estos datos de control de la interacción actualizados 8 en el flujo de datos de salida 10.

[0070] La Figura 6 ilustra un procesador de datos de audio 1 de acuerdo con la invención, donde se procesa información de audio opcional. Como se puede ver, el flujo de datos de entrada 15 es un flujo principal de datos de audio que comprende datos de audio codificados 3 y metadatos relacionados 4. Más aun, la interfaz de receptor 2 recibe además datos de audio opcionales 16 que comprenden un objeto de audio opcional 17.

[0071] Sin embargo, los metadatos relacionados con dicho objeto de audio opcional adicional 'AOx' 17, es decir la información con respecto a las posibilidades de manipulación de dicho objeto de audio opcional 'AOx' 17, están contenidos en el flujo principal de datos de audio 15. De esa manera, el objeto de audio 17 es conocido, aunque no está presente y, por lo tanto, es opcional.

[0072] Por ejemplo, el usuario escucha una orquesta que contiene batería, instrumentos de cuerda y un piano. Opcionalmente se pueden agregar instrumentos de viento. Si el oyente ahora desea agregar un instrumento de viento, puede hacerlo agregando un instrumento de viento opcional, por ejemplo, una trompeta, como objeto de audio opcional 17. Como las posibilidades de manipulación correspondientes a dicha trompeta ya están contenidas dentro de los 5 metadatos 4 del flujo principal de datos de audio 15, el usuario tiene la posibilidad de manipular la trompeta ahora agregada de acuerdo con sus deseos.

[0073] También con referencia a la Figura 6, dicho objeto de audio adicional 'AOx' 17 puede ser un objeto de audio faltante que no está incluido en los datos de audio codificados 3 y/o en los metadatos 4. De esa manera, no se 10 conoce el objeto de audio 17 y por lo tanto falta.

[0074] En este caso, la interfaz de receptor 2 está configurada para solicitar datos de audio 16 pertenecientes a dicho objeto de audio que falta 17 de un proveedor de datos de audio 35. La interfaz de receptor 2 también está configurada para recibir dichos datos de audio 16 de un subflujo diferente contenido en un flujo de transmisión 36. La 15 interfaz de receptor 2 está configurada a su vez para obtener dichos datos de audio 16 de internet 37 por medio de una conexión por protocolo de internet.

[0075] Por ejemplo, un usuario que ve una película puede optar por un idioma específico de la serie de idiomas disponible que contiene, por ejemplo, inglés, alemán y francés. Se conoce un cuarto idioma, pero no está presente y, 20 por lo tanto, falta. Sin embargo, posteriormente se puede incluir un cuarto idioma a través de internet, por ejemplo.

[0076] Con referencia a la Figura 4 y a la Figura 5 nuevamente, el flujo de datos de entrada 15 y el flujo de datos de salida 10 pueden presentarse en general en una estructura de paquetes. Por ejemplo, el transporte de Audio MPEG-H a través de interfaces en serie está definido por la sintaxis de transporte MHAS (véase la sección 13 de 25 N14459 (Texto de Borrador del Comité ISO/IEC 23008-3) [1]). Esta sintaxis está definida en forma de paquetes.

[0077] Por lo tanto, para lograr el transporte de los metadatos 4 o los datos de control de la interacción 8 manipulados, solo es necesario asignar un tipo de paquete adicional para la nueva información de control.

[0078] Además, un primer dispositivo 'dispositivo #1' 19 que comprende el procesador de datos de audio 1 podría querer agregar datos de relleno 18 adicionales a un flujo de transmisión 10 subsiguiente para satisfacer el requisito de tasa de datos dado, habitualmente mucho más elevada para ese vínculo. Estos datos de relleno 18 pueden 30 no contener información alguna y se espera que sean ignorados por un segundo dispositivo que recibe el flujo de salida manipulado 10. Para lograr esto, se puede asignar otro tipo de paquete de datos.

[0079] Además, como la capa de transporte de los flujos entrantes al 'dispositivo #1' 19 puede tener su propia protección contra errores, aunque el vínculo saliente no ofrece dicha capa de seguridad, el dispositivo #1 puede 35 agregar paquetes de datos que contienen datos de verificación de paridad. Estos pueden ser agregados a los flujos de MHAS en forma de tipo de paquetes adicional.

[0080] Además, como la capa de transporte puede acarrear datos adicionales como información lateral, estos paquetes de datos también pueden estar contenidos en el flujo de audio MHAS como tipo de paquetes adicional. Un 40 ejemplo de estos datos son los descriptores contenidos en el flujo de Transporte y programa de MPEG-2.

[0081] Otro ejemplo de almacenamiento de datos de audio codificados es el formato de archivos ISO mp4. Como ocurre con el formato de flujos, también es posible en el caso de un formato de archivos leer, manipular y 45 reescribir los metadatos en el archivo o almacenar los datos de control del usuario además de los metadatos originales sin cambiar los datos de señal de audio codificada.

50 Referencias

[0082]

[1] ISO N14459 (ISO/IEC 23008-3 Committee Draft Text)

55 [2] IEC 60958-3: *Digital audio interface – Part 3: Consumer applications*

[3] IEC 61937-11, *Digital audio – Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 – Part 11: MPEG-4 AAC and its extensions in LATM/LOAS*

[0083] Con referencia, ahora, a la Figura 7, el procesador de datos de audio 1 se implementa en forma de 60 dispositivo separado 'dispositivo #1' 19. En dicho dispositivo separado 19, la interfaz de receptor 2 forma una entrada 20 al dispositivo separado 19 a través de una conexión por cable 21 o una conexión inalámbrica 22.

[0084] El procesador de datos de audio 1 comprende además una interfaz de salida 23 que está conectada al generador de flujos de datos 9 y produce una salida para el dispositivo 19. Además, la interfaz de salida 23 está 65 configurada para emitir el flujo de datos de salida 10 a través de una interfaz inalámbrica 24 o un conector cableado

25.

[0085] Se pueden describir otras maneras de poner en práctica la invención, a título de ejemplo, en una situación en que existen dos dispositivos separados. Un primer dispositivo 'dispositivo #1' comprende el procesador de datos de audio de acuerdo con la invención. Un segundo dispositivo 'dispositivo #2' recibe los datos de audio procesados, aunque aún codificados, del 'dispositivo #1' para decodificar dichos datos de audio.

[0086] Como se puede apreciar en las Figuras 8 y 9, el primer dispositivo 19, 26 recibe un flujo de entrada 15 que comprende datos de audio codificados 3 y metadatos relacionados 4. El primer dispositivo 19, 26 lee la información de metadatos 4 del flujo entrante de datos de audio 15 o del flujo de transporte y deja intactos los datos de señal de audio codificados 3. El primer dispositivo 19, 26 analiza los metadatos 4 y presenta información acerca de los objetos a la aplicación, por ej. en la interfaz de interacción 6, incluyendo las restricciones a la manipulación de objetos que forman parte de los metadatos 4.

[0087] De la interfaz de aplicación o interacción 6 un usuario puede seleccionar y manipular los objetos para adaptar la presentación de audio a sus preferencias personales:

- Selección de objetos: por ej. seleccionar uno de varios idiomas, seleccionar objetos de audio adicionales opcionales, etc.
- Manipulación de objetos: por ej. cambio de volumen de los objetos, cambio de la posición de los objetos, como por ejemplo el desplazamiento de un comentario adicional del altavoz central al altavoz derecho o una posición arbitraria entre estos,
- Selección de opciones preestablecidas: en lugar de seleccionar y manipular cada objeto por separado, el usuario también puede seleccionar una opción preestablecida de los metadatos. Una opción preestablecida es una preselección de objetos recomendada por el creador de contenidos para aplicaciones o situaciones de uso específicas. Una opción preestablecida puede contener una combinación de objetos con, por ej., datos de compresión de niveles de volumen, posiciones y volumen/rango dinámico diferentes en comparación con la presentación por defecto.

[0088] En el paso siguiente, el primer dispositivo 19, 26 almacena la información acerca de la interactividad del usuario (datos de control de la interacción 8) en los flujos de datos de audio codificados 10, 32. El primer dispositivo 19, 26 puede escribir los valores modificados o la cantidad de manipulación, por ej. los valores de desplazamiento y el factor de multiplicación nuevamente en la parte de los metadatos 4 o una parte especializada de los flujos de datos de audio codificados de manera que la salida del primer dispositivo 19, 26 sea, una vez más, un flujo de audio codificado válido 10, 32.

[0089] El primer dispositivo 19, 26 puede usar un identificador, etiqueta o tipo de paquetes diferente para encapsular los metadatos manipulados o los datos de control del usuario 8, respectivamente. Los metadatos originales 4 pueden permanecer en el flujo de salida 10, 32 además de los metadatos manipulados. Se utiliza un identificador, etiqueta o tipo de paquetes diferente para los metadatos manipulados o los datos de control de la interacción 8, respectivamente, para habilitar al segundo dispositivo 28, 33 para identificar si los metadatos 4 han sido manipulados anteriormente. Los metadatos originales 4 permanecen en el flujo 10, 32 para permitir que el segundo dispositivo 28, 33 identifique las diferencias como resultado de la interacción del usuario o del dispositivo y para reunir toda la información acerca de los valores por defecto pretendidos por el creador de contenidos, o para calcular el resultado de la interacción del usuario o del dispositivo a partir de los metadatos originales 4 y los metadatos manipulados (o datos de control por el usuario 8).

[0090] Como parte de los metadatos manipulados, los datos de verificación de firma pueden estar incluidos en los metadatos 4. La firma puede contener información acerca de la aplicación, el dispositivo o el usuario que manipuló los metadatos 4.

[0091] Los objetos de audio que no son seleccionados por el usuario pueden ser marcados como deshabilitados en los metadatos 4 o los datos de control de interacción 8, o de lo contrario, la parte de audio codificada de estos objetos puede ser eliminada del flujo de audio 10.

[0092] El proceso de interactividad del usuario o el dispositivo puede ser dinámico, es decir que cada vez que el usuario o el dispositivo cambia los ajustes para la selección y manipulación, el primer dispositivo 19, 26 escribe esos valores cambiados nuevamente en la parte de metadatos 4 de los flujos de datos de audio codificados 10.

[0093] También es posible que el segundo dispositivo 28, 33 manipule a su vez los metadatos 4, ya sea debido a un proceso automático (por ej. para adaptar la escena de audio a la situación de escucha) o una interfaz de interacción adicional. En este caso, el segundo dispositivo 28, 33 puede reescribir los valores manipulados en los flujos de datos de audio codificados 10, por ej. sobrescribiendo los valores escritos por el primer dispositivo 19, 26.

[0094] Con referencia, una vez más, a la Figura 6, se describe otra manera de llevar a la práctica la invención

a manera de ejemplo en un llamado Caso de uso híbrido.

[0095] Los datos de audio codificados 3 de los objetos de audio opcionales 17 seleccionados pueden no formar parte del flujo principal de datos de audio 15, sino que pueden ser transferidos utilizando otros canales de transporte.

5 Por ejemplo, el flujo principal de datos de audio 15 se transfiere por un canal de transmisión, en tanto que los datos de audio codificados 3 de los objetos de audio opcionales 17 son transferidos bajo demanda a través de una conexión IP 37.

[0096] Los metadatos completos 4 correspondientes a todos los objetos están incluidos en el flujo principal de
10 datos de audio 15 de tal manera que toda la información respecto a la interacción y la selección de objetos está disponible en el primer dispositivo 19. Por lo tanto, el proceso de interacción y almacenamiento de los datos de control de la interacción 8 al flujo 10 es idéntico al caso antes descrito.

[0097] Si un usuario selecciona un objeto 17 y los datos de audio codificados 3 no son parte del flujo principal
15 de datos de audio 15, el primer dispositivo 19 puede recibir los datos de audio codificados 3 de este objeto 17 a través de una conexión de datos diferente de la utilizada para el flujo principal de datos de audio 15, por ej. un subflujo diferente 36 dentro del flujo de transmisión o una conexión IP 37.

[0098] En el paso siguiente, el primer dispositivo 19 fusiona los datos de audio codificados 3 del objeto
20 seleccionado 17 en el flujo de audio principal 15, produciendo así un flujo de datos de audio completo 10 para ser transferido, a su vez, al segundo dispositivo 33.

[0099] Más aun, se propone una posibilidad de transportar los datos de interacción 8 embutidos en un flujo de
25 bits MPEG-H conforme 10.

[0100] Como se puede apreciar en la Figura 9, los sistemas multimedia avanzados con frecuencia no integran
toda la funcionalidad deseada a un dispositivo único, sino que implementan diferentes componentes funcionales en dispositivos especializados tales como receptores digitales externos 26, un aparato de televisión 27 o un receptor AVR
28, por ejemplo. Estos dispositivos se comunican a través de interfaces normalizadas tales como HDMI.

[0101] Sin embargo, también es posible que por lo menos el procesador de datos de audio y un decodificador
estén integrados en un dispositivo único. El procesador de datos de audio puede ser proporcionado, es decir integrado o implementado, junto con un decodificador dentro del mismo dispositivo (*hardware*). Por ejemplo, el procesador de
30 datos de audio y un decodificador se pueden proporcionar en conjunto dentro de un televisor, un receptor digital
35 externo, un receptor A/V, o similar. El procesador de datos de audio y el decodificador se pueden comunicar a través de estructuras de buses de datos internos. Esa configuración puede ser particularmente deseada en dispositivos de
televisión que comprenden soluciones de Sistema en Chip (SoC).

[0102] En consecuencia, o por otro lado, el procesador de datos de audio puede ser implementado en forma
40 de componente funcional independiente y separado en el mismo dispositivo, caso similar al descrito anteriormente con respecto a los dos dispositivos separados, donde la única diferencia reside en que la interfaz de salida ejecuta una salida del procesador de datos de audio por una conexión interna del dispositivo, por ejemplo utilizando un bus de
datos interno.

[0103] Un caso de uso de reproducción de contenido de MPEG-H con múltiples dispositivos participantes es el
45 caso en que un programa de televisión es recibido por un primer dispositivo 19 tal como un receptor digital externo
(STB) 26, que selecciona el canal de transmisión apropiado y extrae los flujos elementales pertinentes que contienen la esencia codificada buscada. En este caso también se implementa la interacción, es decir el control por el usuario
de la selección de elementos de audio e interacción/manipulación.

[0104] El decodificador MPEG-H 31 puede no estar ubicado en el STB 26 sino, por el contrario, en el receptor
50 de audio-vídeo (AVR) 28. Este caso de uso está ilustrado en la Figura 9.

[0105] En este caso, es necesario alimentar los flujos extraídos al AVR 28 para la reproducción; la transmisión
55 entre estos dos dispositivos 26, 28 se puede realizar ya sea transmitiendo una representación
decodificada/descomprimida (PCM con audio), o, especialmente si se aplican restricciones de ancho de banda a la
línea de interconexión utilizada, en una representación codificada.

[0106] El AVR 28 solo presenta entonces una interfaz de configuración a la que habitualmente el usuario
60 accede solo una vez al configurar el sistema y actúa en "modo esclavo" en los períodos de operación normal.

[0107] Como la interacción tiene lugar en el STB (el dispositivo #1) 26 y la decodificación y renderización son
implementadas en el AVR (el dispositivo #2) 28, es inevitable poder enviar al usuario información de interactividad del
STB 26 al AVR 28.

65

[0108] Para lograr el caso de uso descrito y para superar las limitaciones descritas, se propone incluir datos de información de interacción 8 en los flujos de datos de audio codificados 10, como se puede apreciar en la Figura 8.

[0109] El primer dispositivo 19 lee la información de metadatos 4 de la `mpegh3daConfig()` del flujo entrante de datos de audio 15 o a través de una señalización fuera de banda tal como los Descriptores de MPEG-2 TS. A continuación, analiza los metadatos de elementos de audio 4 y presenta información acerca de los objetos en la interfaz de interacción 6, incluyendo las restricciones a la manipulación de objetos que forman parte de los metadatos 4. De la interfaz de interacción 6 el usuario puede seleccionar y manipular el elemento de audio para adaptar la presentación de audio a sus preferencias personales. Los "datos de interacción del usuario" describen la selección y manipulación de los objetos.

[0110] En el paso siguiente, el primer dispositivo 19 escribe estos datos en el flujo de datos de audio MHAS 32 utilizando un nuevo `MHASPacketType`. La salida del primer dispositivo 19 es, una vez más, un flujo de audio codificado válido 10, 32. Los metadatos originales 4 presentes en `mpegh3daConfig()` y en los datos de señal de audio codificados 3 no se modifican.

[0111] La presencia de un paquete de este `MHASPacketType` habilita al segundo dispositivo 28, 33 para identificar que se ha producido una interacción (por el usuario). Los metadatos originales 4 se mantienen en el flujo 10, 32 para permitir que el segundo dispositivo 10, 33 obtenga toda la información acerca de los valores por defecto, de acuerdo con la intención del creador de contenidos.

[0112] Los elementos de audio que no son seleccionados por el usuario pueden ser marcados como deshabilitados en los metadatos 4, o de lo contrario se puede eliminar del flujo de audio 10 la parte de audio codificada 3 de los objetos.

[0113] El proceso de interacción, es decir la interactividad del dispositivo o del usuario, puede ser dinámico, es decir que cada vez que un usuario o un dispositivo cambia los ajustes para la selección y manipulación, el primer dispositivo 19, 26 escribe esos valores modificados de nuevo en la parte específica de los flujos de datos de audio codificados 10.

Extensión MHAS

[0114] El transporte de Audio MPEG-H a través de interfaces en serie está definido en la sintaxis de transporte de MHAS (véase la sección 13 de N14459 (ISO/IEC 23008-3 Committee Draft Text) [1]). Esta sintaxis se define en forma de paquetes. Por lo tanto, para efectuar el transporte de los datos de interacción del usuario, solo se necesita asignar un tipo de paquetes adicional para la nueva información de control:

Tabla 1 — Sintaxis de MHASPacketPayload()

Sintaxis	Núm. de bits	Mnemónico
<pre> MHASPacketPayload(MHASPacketType) { switch (MHASPacketType) { case PACTYP_SYNC: 0xA5; /* syncword*/ break; case PACTYP_MPEGH3DACFG: mpeg3daConfig(); break; case PACTYP_MPEGH3DAFRAME: mpeg3daFrame(); break; case PACTYP_SYNCGAP: syncSpacingLength = escapedValue(16,24,24); break; case PACTYP_MARKER: for (i=0; i< MHASPacketLength; i++) { marker_byte(i); } break; case PACTYP_USERINTERACTION: mpeg3daElementInteraction(); break; } ByteAlign(); } </pre>	8 16,40,64 8	uimsbf uimsbf

Tabla 2 — Valor de MHASPacketType

MHASPacketType	Valor
PACTYP_SYNC	0
PACTYP_MPEGH3DACFG	1
PACTYP_MPEGH3DAFRAME	2
<i>/* reserved for ISO use */</i>	3-6
PACTYP_SYNCGAP	7
PACTYP_MARKER	8
PACTYP_USERINTERACTION	12
<i>/* reserved for ISO use */</i>	13-127
<i>/* reserved for use outside of ISO scope */</i>	128-261
<i>/* reserved for ISO use */</i>	262-389
<i>/* reserved for use outside of ISO scope */</i>	390-517

NOTA: Es obligatorio que los valores de MHASPacketType de aplicación específica se encuentren en el espacio reservado para el uso fuera del alcance de ISO. Estos son omitidos por un decodificador, ya que el decodificador requiere un mínimo de estructura para omitir estas extensiones.

5

PACTYP_USERINTERACTION

[0115] El MHASPacketType PACTYP_USERINTERACTION se puede utilizar para alimentar los datos de interacción de los elementos al decodificador.

10

[0116] Para este tipo de paquetes, MHASPacketLabel tiene el mismo valor que para el paquete de MHASPacketType PACTYP_MPEGH3DACFG, al cual hacen referencia los datos de interacción (usuario) 8.

Sintaxis

15

[0117] El elemento de sintaxis mpeg3daElementInteraction() (véase [2]) de la interfaz para interacción del usuario es reutilizado para transferir los datos de interacción (usuario) 8 contenidos en el flujo de bits 10.

Referencias

20

[0118]

[1] ISO/IEC JTC1/SC29/WG11 N14459, "Text of ISO/IEC 23008-3/CD, 3D audio"

[2] FhG, *Normative Interface for User Interaction*, Input to the Meeting of AHG on 3D Audio, DRC and Audio Maintenance, junio 2–3, 2014, París, Francia

5

[0119] Más aun, se propone una posible sintaxis para una interfaz normativa para la interactividad con el usuario.

10 [0120] Se propone extraer la interfaz para el control de interacción de la interfaz para el control de la renderización. Luego el elemento de sintaxis `mpegh3daAudioRendering()` define los parámetros de renderización y un elemento de sintaxis recientemente definido `mpegh3daElementInteraction()` contiene la información que se necesita para la interacción (usuario).

Definición del elemento de sintaxis `mpegh3daElementInteraction()`

15

[0121] El elemento de sintaxis `mpegh3daElementInteraction()` proporciona una interfaz para cualquier posible interacción (usuario). Se definen dos modos de interacción.

20 [0122] El primero es el modo de interacción avanzada, en el cual se puede señalar la interacción para cada grupo de elementos que está presente en la escena de audio. Este modo habilita al usuario para elegir libremente (dentro de las restricciones de las definiciones de los grupos de conmutación) qué grupos reproducir y para interactuar con todos ellos (dentro de los rangos y restricciones dados).

25 [0123] El segundo modo es un modo de interacción básica, en que el usuario podría elegir una de las `GroupPresets` definidas (del elemento de sintaxis `mae_AudioSceneInfo()`, véase 14.2 of [1]) como opción preestablecida. Seguidamente se define el estado activo–inactivo de los grupos a que se hace referencia en las condiciones de las `GroupPresets` elegidas y no puede ser modificado por el usuario. El usuario solo puede cambiar el estado activo–inactivo de los demás grupos y las posiciones y ganancias de todos los grupos de acuerdo con las concesiones y rangos definidos.

30

[0124] Se distinguen cuatro modificaciones de elementos diferentes:

- **Interactividad Activa/Inactiva:** Se activa o desactiva un grupo de elementos (*Nota editorial: anteriormente llamada "cambio de objeto" [4]*)
- 35 - **Interactividad de Posiciones:** Se cambian las posiciones de un grupo de elementos (azimut, elevación y distancia, *nota editorial: anteriormente denominada "cambio de posición" [4]*)
- **Interactividad de ganancia:** Se cambia el nivel/ganancia de un grupo de elementos (*Nota editorial: anteriormente denominada "cambio de ganancia" [4]*)
- **Interactividad de CABLE:** Una salida de CABLE es una salida arbitraria además de los altavoces conectados.
- 40 Se enruta el contenido de audio de los elementos de un grupo hacia una salida de CABLE, por ejemplo, un contenido para personas con discapacidad auditiva o una pista de idioma adicional.

[0125] Todas las modificaciones se definen en un nivel de grupos de elementos, puesto que los grupos reúnen elementos relacionados que solo deben ser manipulados de forma conjunta.

45

[0126] Se introduce una firma para señalar, por ejemplo, en qué dispositivo ha tenido lugar la interactividad.

Tabla 3a Sintaxis de mpeg3daElementInteraction()

Sintaxis	Núm de bits	Mnemónico
mpeg3daElementInteraction() { ei_InteractionSignatureDataLength ; if (ei_InteractionSignatureDataLength > 0) { ei_InteractionSignatureDataType ; for (c = 0; c < bsInteractionSignatureDataLength + 1; c++) { ei_InteractionSignatureData [c]; } } ElementInteractionData(); }	8 8 8	uimsbf uimsbf uimsbf

ei_InteractionSignatureDataLength

Este campo define la longitud de la siguiente firma de interacción en Byte.

5 **ei_InteractionSignatureDataType**

Este campo define el tipo de firma. Los siguientes valores son posibles:

[0127] En la Tabla 3b se expone otra forma de realización de mpeg3daElementInteraction():**Tabla 3b Sintaxis de mpeg3daElementInteraction()**

Sintaxis	Núm de bits	Mnemónico
mpeg3daElementInteraction() { ei_InteractionSignatureDataLength ; if (ei_InteractionSignatureDataLength > 0) { ei_InteractionSignatureDataType ; for (c = 0; c < bsInteractionSignatureDataLength + 1; c++){ ei_InteractionSignatureData [c]; } } ElementInteractionData(); hasLocalZoomAreaSize ; if (hasLocalZoomAreaSize) { LocalZoomAreaSize(); } }	8 8 8 1	uimsbf uimsbf uimsbf bslbf

10

Tabla 4a Valor de ei_InteractionSignatureDataType

valor	significado
0	Cadena genérica
1-127	Reservado para uso por ISO
128-255	Reservado para uso fuera del alcance de ISO

ei_InteractionSignatureData

Este campo contiene una firma que define quién originó los datos de interacción.

15

[0128] En la Tabla 4b se expone otra forma de realización de ei_InteractionSignatureDataType:**Tabla 4b Valor de ei_InteractionSignatureDataType**

valor	significado
0	Cadena genérica en UTF-8 de acuerdo con ISO/IEC 10646
1-127	Reservado para uso de ISO
128-255	Reservado para usar fuera del alcance de ISO

20

ei InteractionSignatureData

Este campo contiene una firma que define quién ha originado los datos de interacción.

hasLocalZoomAreaSize

5

Bandera que define si hay información acerca del tamaño del área de zoom local. Si esta bandera está habilitada, se aplica el Remapeo de Objetos para Zooming (acercamiento-alejamiento).

Tabla 5 Sintaxis de ElementInteractionData()

Sintaxis	Núm de bits	Mnemónico
ElementInteractionData()		
{		
ei_interactionMode;	1	bslbf
ei_numGroups; /* Channel, Object, HOA, SAOC */	7	uimsbf
If (ei_interactionMode == 0) {		
/* direct element group interaction = advanced */		
ei_GroupInteractivityStatus(ei_numGroups);		
} else {		
/*GroupPreset interaction = basic */		
ei_groupPresetID;	5	uimsbf
ei_GroupInteractivityStatus(ei_numGroups);		
}		

10 ei interactionMode

Bandera que define si se ha elegido el tipo de interacción avanzada o el modo de interacción básica.

ei numGroups

Este campo contiene el número de grupos en la escena de audio.

ei_groupPresetID

Este campo contiene un `groupPresetID` que se define en la escena de audio. Esta ID refleja la elección preestablecida del usuario.

15

Tabla 6a Sintaxis de `ei_GroupInteractivityStatus()`

Sintaxis	Núm de bits	Mnemónico
ei_GroupInteractivityStatus (numGroups)		
{		
for (grp = 0; grp < numGroups; grp++) {		
ei_groupID[grp];	7	uimbsbf
ei_onOff[grp];	1	bslbf
ei_routeToWIRE[grp];	1	bslbf
if (ei_routeToWIRE[grp] == 1) {		
routeToWireID[grp];	4	uimbsbf
}		
if (ei_onOff [grp] == 1) {		
ei_changePosition[grp]; /* position change */	1	bslbf
if (ei_changePosition[grp]) {		
ei_azOffset[grp];	8	uimbsbf
ei_elOffset[grp];	6	uimbsbf
ei_distFact[grp];	4	uimbsbf
}		
ei_changeGain; /* gain change */	1	bslbf
if (ei_changeGain) {		
ei_gain;	7	uimbsbf
}		
}		
}		

ei_groupID

GroupID correspondiente al grupo en curso del cual se describe la interacción.

20 ei routeToWIRE

Este campo define si el contenido de audio del grupo debe ser enrutado hacia una salida de CABLE.

ei routeToWireID

ID de la salida de CABLE hacia donde se debe enrutar el grupo.

25 ei onOff

Define el estado activo-inactivo del grupo actual. En el caso de elegirse el modo de interacción básica (interacción en GroupPresets), este valor tiene que ser idéntico al estado activo-inactivo definido del grupo con el groupID si este grupo es parte de las

condiciones de la GroupPreset elegida con ei_groupPresetID. En el modo de interacción básica no se permite señalar aquí un estado activo–inactivo diferente. El estado activo–inactivo de todos los grupos que no son parte de las condiciones de la GroupPreset elegida, se podría señalar el estado activo–inactivo de manera arbitraria.

- 5 **ei_changePosition** Esta bandera define si la posición de los elementos del grupo se ha modificado.
- ei_azOffset** El cambio de azimut se da como desplazamiento. Este campo puede asumir valores entre AzOffset=-180° y AzOffset=180°:
10 **AzOffset = 1.5 · (ei_azOffset – 128)**
AzOffset = min (max (AzOffset, -180), 180);
- ei_elOffset** El cambio de azimut se da como desplazamiento. Este campo puede asumir valores entre EIOffset=-90° y EIOffset=90°:
15 **EIOffset = 3 · (ei_elOffset – 32)**
EIOffset = min (max (EIOffset, -90), 90);
- ei_distFact** La interacción de distancia se da en términos de factor de multiplicación. Este campo puede asumir valores entre 0 y 15
20 **para dar lugar a DistFactor entre 0,00025 y 8:**
DistFactor = 2^{((ei_distFact-8)-4)}
DistFactor = min(max (DistFactor, 0,00025), 8);
- ei_changeGain** Esta bandera define si la ganancia/nivel de los elementos del grupo se ha modificado.
25
- ei_gain** Este campo define una ganancia adicional de los miembros del grupo en curso. El campo puede asumir valores entre 0 y 127 que representan valores de ganancia de entre
30 **Ganancia = -63 dB y Ganancia = 31 dB en pasos de 1 dB,**
donde
Ganancia [dB] = ei_gain – 64
Ganancia [dB] = min(max (Gain, -63), 31);
Si se ajusta ei_gain a 0, se debe ajustar Gain a menos infinito dB.
- 35 **[0129]** En la Tabla 6b se presenta otra forma de realización de ei_GroupInteractivityStatus():

Tabla 6b Sintaxis de ei_GroupInteractivityStatus()

Sintaxis	Núm. de bits	Mnemónico
ei_GroupInteractivityStatus (numGroups)		
{		
for (grp = 0; grp < numGroups; grp++) {		
ei_groupID[grp];	7	uimbsf
ei_onOff[grp];	1	bslbf
ei_routeToWIRE[grp];	1	bslbf
if (ei_routeToWIRE[grp] == 1) {		
routeToWireID[grp];	16	uimbsf
}		
if (ei_onOff [grp] == 1) {		
ei_changePosition[grp]; /* position change */	1	bslbf
if (ei_changePosition[grp]) {		
ei_azOffset[grp];	8	uimbsf
ei_elOffset[grp];	6	uimbsf
ei_distFact[grp];	4	uimbsf
}		
ei_changeGain; /* gain change */	1	bslbf
if (ei_changeGain) {		
ei_gain;	7	uimbsf
}		
}		
}		
}		
}		

[0130] Los datos de interacción se definen de acuerdo con los campos de Elementos de metadatos de audio (véase 14.2 de [1]). En la definición de elementos de audio de metadatos, se da el rango de interactividad para la interactividad de ganancia en dB, los rangos de interactividad correspondientes a azimut y elevación se dan en términos de valores de desplazamiento mínimo y máximo (con la misma resolución que aquí: 1,5° en el caso del azimut y 3° en el caso de la elevación) y los rangos correspondientes a la distancia se dan en términos de factores de multiplicación. Por lo tanto, la interfaz se define de manera similar.

[0131] En el caso de la interacción del usuario, todos los cambios posibles (activo/inactivo, ganancia, posición, CABLE) correspondientes a todos los grupos de elementos están señalizados dentro de un elemento de sintaxis `ElementInteraction()` en lugar de usar una repetición de un elemento de sintaxis por cada tipo de interacción y cada grupo o elemento.

[0132] La salida de CABLE también se define con respecto a los grupos que se inactivan, puesto que solo entonces es posible enrutar las pistas de idioma alternativos a una salida de CABLE. La posible lógica de grupo de activación violada tiene que ser tratada en el decodificador.

[0133] El tipo de contenidos (canal, objeto, HOA) que se había propuesto anteriormente en [4] es eliminado aquí. Ya es conocido implícitamente por la `groupID`, puesto que cada grupo de elementos tiene un `signalGroupType` (Canales, Objetos, SAOC, HOA) que define el tipo de contenido de las señales y se lo señala en el elemento de sintaxis `Signals3d()`.

Interacción básica por medio de GroupPresets

[0134] Si se establece el modo de interacción básica, el usuario puede elegir una de las `GroupPresets` definidas (del elemento de sintaxis `mae_AudioSceneInfo()` `groupCollection`, propuesto como tecnología posible en w14464 [7]).

[0135] Las condiciones de la opción preestablecida elegida definen el estado activo–inactivo de los grupos de referencia. El usuario no debe poder modificar el estado activo–inactivo de estos grupos.

[0136] El correspondiente campo `ei_onOff` del elemento de sintaxis `ei_GroupInteractivityStatus` [2] tiene que reflejar la condición establecida en la `groupPreset` elegida para todos los grupos a que se hace referencia en las condiciones de la `groupPreset` elegida.

[0137] El estado activo–inactivo de todos los demás grupos puede ser elegido por el usuario.

Conclusión

[0138] Esta contribución propone una definición correspondiente a una interfaz para la interacción con el usuario. Se presenta una definición de interfaz que da lugar a dos modos de interacción diferentes (básica y avanzada). Se propone adoptar la interfaz propuesta al CD.

Referencias

[0139]

- [1] ISO/IEC JTC1/SC29/WG11 N14459, *Text of ISO/IEC 23008-3/CD, 3D audio*
- [2] FhG, *Rendering Interfaces to MPEG-H and Unification of Loudspeaker Signaling*, Input to the Meeting of AHG on 3D Audio, DRC and Audio Maintenance, junio 2-3, 2014, París, Francia
- [3] ISO/IEC JTC1/SC29/WG11 N14463, *Normative Interface for Binaural Data*, abril de 2014, Valencia, España
- [4] ISO/IEC JTC1/SC29/WG11 M33134, *3D Audio Decoder Interfaces*, abril de 2014, Valencia, España (Ingresada por Philips)
- [5] ISO/IEC JTC1/SC29/WG11 M31427, *Thoughts on binaural parameterization of MPEG codecs*, octubre de 2013, Ginebra, Suiza (Ingresada por Orange)
- [6] ISO/IEC JTC1/SC29/WG11 M30249, *BRIR interface format: update and implementation*, julio de 2013, Viena, Austria (Ingresada por Philips)
- [7] ISO/IEC JTC1/SC29/WG11 N14464 *Candidate Technologies for 3D Audio*

[0140] Más aun, la sección 13 del texto de ISO/IEC 23008-3 CD [1] define la sintaxis de MHAS que se debe utilizar para encapsular las cargas útiles de Audio 3D de MPEG-H en los flujos (tiempo real). Con este documento se proponen capacidades adicionales para habilitar un uso fiable de MHAS en otros sistemas de transporte aparte del Flujo de transporte de MPEG-2.

Filldata (datos de relleno)

[0141] Ciertos canales de transmisión solo pueden ser operados a una tasa de bits instantánea constante. Para

conseguir esto, se propone un tipo de paquetes de datos de relleno para poder llenar el flujo de MHAS hasta obtener una tasa de bits determinada.

CRC

5

[0142] La capa de Flujo de transporte MPEG-2 [2] se hace cargo de la protección del flujo elemental MHAS encapsulado. De esa manera se puede identificar la pérdida o corrupción de datos en la transmisión.

10 **[0143]** Por otra parte, las interfaces en serie comunes (por ej. AES/EBU, S/PDIF, [3], [4], [5]) no ofrecen una protección suficiente contra errores. Se propone un tipo de paquetes CRC opcional para habilitar la detección de errores si se utiliza MHAS en esas interfaces.

Encapsulación de descriptores

15 **[0144]** Se utilizan flujos de MHAS para transportar audio codificado a y desde los codificadores/decodificadores de flujos de transporte MPEG-2. Se propone transportar información de descriptores relacionada en tipos de paquetes MHAS adicionales.

Tabla 7 — Sintaxis de MHASPacketPayload()

Sintaxis	Núm. de bits	Mnemónico
MHASPacketPayload(MHASPacketType)		
{		
switch (MHASPacketType) {		
case PACTYP_SYNC:		
0xA5 ; /* syncword*/	8	uimsbf
break;		
case PACTYP_MPEGH3DACFG:		
mpegh3daConfig();		
break;		
case PACTYP_MPEGH3DAFRAME:		
mpegh3daFrame();		
break;		
case PACTYP_FILLDATA:		
for (i=0; i< MHASPacketLength; i++) {		
mhas_fill_data_byte(i) ;	8	bslbf
}		
break;		
case PACTYP_SYNGAP:		
syncSpacingLength = escapedValue(16,24,24);	16,40,64	uimsbf
break;		
case PACTYP_MARKER:		
for (i=0; i< MHASPacketLength; i++) {		
marker_byte(i) ;	8	
}		
break;		
case PACTYP_CRC16:		
mhasParity16Data ;	16	bslbf
break;		
case PACTYP_CRC32:		
mhasParity32Data ;	32	bslbf
break;		
case PACTYP_DESCRIPTOR:		
for (i=0; i< MHASPacketLength; i++) {		
mhas_descriptor_data_byte(i) ;	8	bslbf
}		
break;		
}		
ByteAlign();		
}		

20

Tabla 8a — Valor de MHASPacketType

MHASPacketType	Valor
PACTYP_SYNC	0
PACTYP_MPEGH3DACFG	1
PACTYP_MPEGH3DAFRAME	2
<i>/* reserved for ISO use */</i>	3-5
PACTYPE_FILLDATA	6
PACTYP_SYNGAP	7
PACTYPE_MARKER	8
PACTYPE_CRC16	9
PACTYPE_CRC32	10
PACTYPE_DESCRIPTOR	11
<i>/* reserved for ISO use */</i>	13-127
<i>/* reserved for use outside of ISO scope */</i>	128-261
<i>/* reserved for ISO use */</i>	262-389
<i>/* reserved for use outside of ISO scope */</i>	390-517

NOTA: Es obligatorio que los valores de MHASPacketType de aplicación específica se encuentren en el espacio reservado para uso fuera del alcance de ISO. Estos son omitidos por un decodificador, ya que el decodificador requiere un mínimo de estructura para omitir estas extensiones.

[0145] En la Tabla 8b se detalla otra forma de realización de MHASPacketType:

5

Tabla 8b — Valor de MHASPacketType

MHASPacketType	Valor
PACTYP_SYNC	6
PACTYP_MPEGH3DACFG	1
PACTYP_MPEGH3DAFRAME	2
<i>/* reserved for ISO use */</i>	3-5
PACTYPE_FILLDATA	0
PACTYP_SYNGAP	7
PACTYPE_MARKER	8
PACTYPE_CRC16	9
PACTYPE_CRC32	10
PACTYPE_DESCRIPTOR	11
<i>/* reserved for ISO use */</i>	13-127
<i>/* reserved for use outside of ISO scope */</i>	128-261
<i>/* reserved for ISO use */</i>	262-389
<i>/* reserved for use outside of ISO scope */</i>	390-517

NOTA: Es obligatorio que los valores de MHASPacketType de aplicación específica se encuentren en el espacio reservado para uso fuera del alcance de ISO. Estos son omitidos por un decodificador, ya que el decodificador requiere un mínimo de estructura para omitir estas extensiones

mhasParity16Data un campo de 16 bits que contiene el valor CRC que produce una salida de cero de los 16 registros en el decodificador con el polinomio:

10

$$x^{16} + x^{15} + x^5 + 1$$

y el estado inicial del registro de desplazamiento de 0xFFFF.

mhasParity32Data un campo de 32 bits que contiene el valor CRC que produce una salida de cero de los 32 registros en el decodificador con el polinomio:

15

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

y el estado inicial del registro de desplazamiento de 0xFFFFFFFF.

mhas_fill_data_byte elementos de datos de 8 bits, no se aplican restricciones

20 Marcador de Acceso Aleatorio / Reproducción Inmediata

[0146] Cuando el primer **marker_byte** de la carga útil del paquete es "0x02", el siguiente paquete del tipo PACTYP_MPEGH3DAFRAME con idéntica etiqueta MHASPacketLabel es codificado siguiendo las reglas establecidas en la sección 5.5.5 "Audio Preroll".

25

Marcador de límites de programa

[0147] Cuando el primer **marker_byte** de la carga útil del paquete es "0x03", se produce un límite de programa en este punto temporal y todos los paquetes siguientes pertenecen a un nuevo programa.

PSCTYP_CRC16 y PACTYP_CRC32

[0148] Se puede utilizar el **MHASPacketType** PACTYP_CRC16 y PACTYP_CRC32 para la detección de errores en el paquete anterior MHAS estableciendo **MHASPacketLabel** en el mismo valor. Seguirá directamente al paquete MHAS al que se refiere su valor CRC. Esto puede ser ventajoso cuando se transporta un flujo MHAS por un canal propenso a errores.

[0149] El método de detección de errores utiliza uno del generador de polinomios y estados de registro de desplazamiento asociados según lo definido respecto a **mhasParity16Data** o **mhasParity32Data** respectivamente.

[0150] Los bits incluidos en la verificación de CRC son la **MHASPacketPayload()** completa para el paquete MHAS asociado seguido por las palabras **mhasParity16Data** o **mhasParity32Data** respectivamente.

[0151] En caso de no haber error alguno, cada una de las salidas del registro de desplazamiento ha de ser cero. En el codificador CRC el campo de datos **mhasParity16Data** / **mhasParity32Data** es codificado con un valor tal que así se garantice.

PACTYP_FILLDATA

[0152] El **MHASPacketType** PACTYP_FILLDATA ofrece la posibilidad de agregar datos de relleno para ajustar la tasa de bits instantánea. Esto puede ser deseable en ciertas aplicaciones en tiempo real utilizando un canal de transmisión a tasa constante.

[0153] Como los paquetes de este tipo no están asociados a ciertos datos de carga útil, se fija **MHASPacketLabel** en 0.

[0154] Se espera que el decodificador pase por alto los datos transmitidos en paquetes del tipo PACTYP_FILLDATA. Además, las herramientas intermedias de procesamiento de flujos de MHAS están habilitadas para eliminar dichos paquetes del flujo.

[0155] Está permitido fijar **MHASPacketLength** en 0. Esto da un tamaño mínimo de paquete de 2 bytes.

PACTYP_DESCRIPTOR

[0156] El PACTYP_DESCRIPTOR se puede utilizar para incorporar descriptores MPEG-2 TS/PS a flujos MHAS. Los datos transportados como **mhas_descriptor_data_byte** tienen la misma sintaxis y semántica que las definidas respecto al descriptor() de ISO/IEC 13818-1.

[0157] Para este tipo de paquetes y en el caso de los descriptores transmitidos en el primer bucle de descriptores de TS_program_map_section() (véase ISO/IEC 13818-1), se ajusta **MHASPacketLabel** a 0.

[0158] Para este tipo de paquetes y para los descriptores asignados a un flujo elemental (es decir el segundo bucle de descriptores de la TS_program_map_section()), **MHASPacketLabel** se ajusta al mismo valor que el PACTYPE_CONFIG del flujo elemental asociado.

Referencias

[0159]

- [1] ISO/IEC JTC1/SC29/WG11 N14459, *Text of ISO/IEC 23008-3/CD, 3D audio*
- [2] ISO/IEC 13818-1:2013, Information technology — Generic Coding of moving pictures and associated audio information: Systems
- [3] IEC 60958-3: *Digital audio interface – Part 3: Consumer applications*
- [4] IEC 61937-11, *Digital audio – Interface for non-linear PCM encoded audio bitstreams applying IEC 60958 – Part 11: MPEG-4 AAC and its extensions in LATM/LOAS*
- [5] SMPTE 2041: Format for Non-PCM Audio y Data in AES-3 - MPEG-4 AAC and HE AAC Compressed Digital Audio in ADTS and LATM/LOAS Wrappers

[0160] Aunque se ha descrito la presente invención en el contexto de diagramas de bloques en que los bloques representan componentes reales o lógicos de hardware, la presente invención también puede ser implementada por

un método implementado por ordenador. En este último caso, los bloques representan pasos correspondientes del método, donde estos pasos representan las funcionalidades ejecutadas por bloques lógicos o físicos de hardware correspondientes.

5 **[0161]** Si bien se han descrito algunos aspectos en el contexto de un aparato, es obvio que estos aspectos también representan una descripción del método correspondiente, en el cual un bloque o dispositivo corresponde a un paso del método o a una característica de un paso del método. De manera análoga, los aspectos descritos en el contexto de un paso del método también representan una descripción de un bloque o ítem correspondiente o de una característica de un aparato correspondiente. Algunos o todos los pasos del método pueden ser ejecutados por (o
10 utilizando) un aparato de hardware, como por ejemplo un microprocesador, un ordenador programable o un circuito electrónico. En algunas formas de realización, uno o más de los pasos más importantes del método pueden ser ejecutados por ese tipo de aparato.

[0162] La señal de la invención transmitida o codificada puede ser almacenada en un medio de
15 almacenamiento digital o puede ser transmitida por un medio de transmisión tal como un medio de transmisión inalámbrico o un medio de transmisión por cable tal como internet.

[0163] Dependiendo de ciertos requisitos de implementación, las realizaciones de la invención pueden ser implementadas en hardware o en software. La implementación se puede realizar empleando un medio de
20 almacenamiento digital, por ejemplo, un disco flexible, un DVD, un Blu-Ray, un CD, una ROM, una PROM, una EPROM, una EEPROM o una memoria FLASH, que tiene almacenadas en la misma señales control legibles electrónicamente, que cooperan (o tienen capacidad para cooperar) con un sistema informático programable de tal manera que se ejecute el método respectivo. Por lo tanto, el medio de almacenamiento digital puede ser legible por un ordenador.

25 **[0164]** Algunas formas de realización de acuerdo con la invención comprenden un portador de datos que comprende señales de control legibles electrónicamente, con capacidad para cooperar con un sistema informático programable, de tal manera que se ejecute uno de los métodos descritos en la presente.

30 **[0165]** En general, las formas de realización de la presente invención pueden ser implementadas en forma de producto de programa de ordenador con un código de programa, siendo el código de programa operativo para llevar a cabo uno de los métodos al ejecutarse el producto de programa de ordenador en un ordenador. El código de programa puede ser almacenado, por ejemplo, en un portador legible por una máquina.

35 **[0166]** Otras formas de realización comprenden el programa de ordenador para llevar a cabo uno de los métodos aquí descritos, almacenado en un portador legible por una máquina.

[0167] Dicho de otro modo, una forma de realización del método de la invención consiste, por lo tanto, en un programa de ordenador que consta de un código de programa para llevar a cabo uno de los métodos aquí descritos
40 al ejecutarse el programa de ordenador en un ordenador.

[0168] Otra forma de realización del método de la invención consiste, por lo tanto, en un portador de datos (o un medio de almacenamiento no transitorio como un medio de almacenamiento digital, o un medio legible por ordenador) que comprende, grabado en el mismo, el programa de ordenador para llevar a cabo uno de los métodos
45 aquí descritos. El portador de datos, el medio de almacenamiento digital o el medio grabado son por lo general tangibles y/o no transitorios.

[0169] Otra forma de realización del método de la invención es, por lo tanto, un flujo de datos o una secuencia de señales que representa el programa de ordenador para llevar a cabo uno de los métodos aquí descritos. El flujo de
50 datos o la secuencia de señales pueden estar configurados, por ejemplo, para ser transferidos a través de una conexión de comunicación de datos, por ejemplo, a través de internet.

[0170] Otra forma de realización comprende un medio de procesamiento, por ejemplo, un ordenador, o un dispositivo lógico programable, configurado, o adaptado, para llevar a cabo uno de los métodos aquí descritos.
55

[0171] Otra forma de realización comprende un ordenador en el que se ha instalado el programa de ordenador para llevar a cabo uno de los métodos aquí descritos.

[0172] Otra forma de realización de acuerdo con la invención comprende un aparato o un sistema configurado
60 para transferir (por ejemplo, por vía electrónica u óptica) un programa de ordenador para llevar a cabo uno de los métodos aquí descritos a un receptor. El receptor puede ser, por ejemplo, un ordenador, un dispositivo móvil, un dispositivo de memoria o similar. El aparato o sistema puede comprender, por ejemplo, un servidor de archivos para transferir el programa de ordenador al receptor.

65 **[0173]** En algunas formas de realización, se puede utilizar un dispositivo lógico programable (por ejemplo, una

matriz de puertas programables en el campo) para llevar a cabo algunas o todas las funcionalidades de los métodos aquí descritos. En algunas formas de realización, una matriz de puertas programables en el campo puede cooperar con un microprocesador para llevar a cabo uno de los métodos aquí descritos. Por lo general, los métodos son llevados a cabo preferentemente por cualquier aparato de hardware.

5

[0174] Las formas de realización precedentemente descritas son meramente ilustrativas de los principios de la presente invención. Se entiende que las modificaciones y variaciones de las disposiciones y los detalles aquí descritos han de ser evidentes para otras personas expertas en la materia. Por lo tanto, solo se pretende limitarse al alcance de las siguientes reivindicaciones de patente y no a los detalles específicos presentados a manera de descripción y

10 explicación de las formas de realización de la presente.

REIVINDICACIONES

1. Un procesador de datos de audio (1) para procesar datos de audio en paquetes, comprendiendo el procesador de datos de audio (1):
 - 5 una interfaz de receptor (2) para recibir datos de audio codificados (3) que comprenden objetos de audio (11, 12) y metadatos (4) relacionados con los objetos de audio (11, 12);
 - un analizador de metadatos (5) para analizar los metadatos (4) para determinar una posibilidad de manipulación de datos de audio de los objetos de audio (11, 12);
 - 10 una interfaz de interacción (6) para recibir una entrada de interacción (7) y para generar, a partir de la entrada de interacción (7), datos de control de la interacción (8) relacionados con la posibilidad de manipulación de datos de audio para manipular los objetos de audio (11, 12) externamente desde un decodificador; y
 - un generador de flujos de datos (9) para obtener los datos de control de la interacción (8) y los datos de audio codificados (3) y los metadatos (4) y para generar un flujo de datos de salida (10) agregando una porción de datos
 - 15 adicionales que contiene los datos de control de la interacción (8) al flujo de datos de salida (10), donde los metadatos originales (4) y los datos de audio codificados (3) no son modificados, de tal manera que el flujo de datos de salida (10) es de nuevo un flujo de audio codificado válido que comprende los datos de audio aún codificados (3), los metadatos (4), y los datos de control de la interacción (8) agregados.
- 20 2. El procesador de datos de audio (1) de la reivindicación 1, en el que la porción de datos adicionales está contenida en un tipo de paquetes que es diferente a los tipos de paquetes para los datos de audio codificados (3) y los metadatos (4).
3. El procesador de datos de audio (1) de la reivindicación 1 o 2, en el que la interfaz de interacción
 - 25 proporciona dos modos de interacción,
 - en el que un primer modo de interacción es un modo de interacción avanzada que se señala para cada grupo de elementos que está presente en una escena de audio para habilitar al usuario para elegir libremente qué grupos reproducir y para interactuar con todos ellos, y
 - 30 en el que un segundo modo es un modo de interacción básica que comprende Group Presets, para habilitar al usuario para elegir una de las Group Presets como opción preestablecida.
4. Un procesador de datos de audio (1) de una cualquiera de las reivindicaciones 1 a 3, en el que los datos de audio codificados (3) comprenden objetos de audio codificados separados (11; 12), en el que al menos una porción
 - 35 (13; 14) de los metadatos (4) está relacionada con un objeto de audio correspondiente (11; 12),
 - en el que el analizador de metadatos (5) está configurado para analizar la porción correspondiente (13; 14) para los objetos de audio codificados (11; 12) para determinar, para al menos un objeto de audio (11), la posibilidad de manipulación de objetos,
 - 40 en el que la interfaz de interacción (6) está configurada para generar, para al menos un objeto de audio codificado (11), los datos de control de la interacción (8) a partir de la entrada de interacción (7) relacionada con al menos un objeto de audio codificado (11).
5. Un procesador de datos de audio (1) de una de las reivindicaciones 1 a 4, en el que la interfaz de
 - 45 interacción (6) está configurada para presentar, a un usuario, la posibilidad de manipulación de datos de audio derivada de los metadatos (4) por el analizador de metadatos (5), y para recibir, del usuario, una entrada del usuario (7) con respecto a la manipulación de datos específica de la posibilidad de manipulación de datos.
6. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,
 - 50 en el que el generador de flujos de datos (9) está configurado para generar, en el flujo de datos de salida (10), los datos de control de la interacción (8) en el mismo formato que los metadatos (4).
7. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,
 - 55 en el que el generador de flujos de datos (9) está configurado para asociar, con los datos de control de la interacción (8), un identificador en el flujo de datos de salida (10), siendo el identificador diferente a un identificador asociado con los metadatos (4).
8. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,
 - 60 en el que el generador de flujos de datos (9) está configurado para agregar, a los datos de control de la interacción (8), datos de verificación de firma, indicando los datos de verificación de firma información sobre una aplicación, un dispositivo o un usuario que lleva a cabo una manipulación de los datos de audio o que proporciona la entrada de interacción.
 - 65

9. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,

en el que el analizador de metadatos (5) está configurado para identificar una posibilidad de deshabilitación para uno o más objetos de audio (11, 12) representados por los datos de audio codificados (3), en el que la interfaz de interacción (6) está configurada para recibir una información de deshabilitación para uno o más objetos de audio (11, 12), y en el que el generador de flujos de datos (9) está configurado para marcar uno o más objetos de audio (11, 12) como deshabilitados en los datos de control de la interacción (8) o para eliminar el uno o más objetos de audio deshabilitados (11, 12) de los datos de audio codificados (3) de tal manera que el flujo de datos de salida (10) no incluya datos de audio codificados (3) para uno o más objetos de audio deshabilitados (11, 12).

10. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes, en el que el generador de flujos de datos (9) está configurado para generar de manera dinámica el flujo de datos de salida (10), en el que en respuesta a una nueva entrada de interacción (7), los datos de control de la interacción (8) se actualizan para coincidir con la nueva entrada de interacción (7), y en el que el generador de flujos de datos (9) está configurado para incluir los datos de control de interacción actualizados (8) en el flujo de datos de salida (10).

11. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes, en el que la interfaz de receptor (2) está configurada para recibir un flujo principal de datos de audio (15) que comprende los datos de audio codificados (3) y metadatos (4) relacionados con los datos de audio codificados (3), y para recibir adicionalmente datos de audio opcionales (16) que comprenden un objeto de audio opcional (17), en el que los metadatos (4) relacionados con dicho objeto de audio opcional (17) están contenidos en dicho flujo principal de datos de audio (15).

12. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,

en el que el analizador de metadatos (5) está configurado para determinar la posibilidad de manipulación de audio para un objeto de audio opcional (17) no incluido en los datos de audio codificados (3), en el que la interfaz de interacción (6) está configurada para recibir una entrada de interacción (7) para el objeto de audio opcional (17), y en el que la interfaz de receptor (2) está configurada para solicitar datos de audio (16) para el objeto de audio opcional (17) de un proveedor de datos de audio (35) o para recibir los datos de audio (16) para el objeto de audio opcional (17) de un subflujo diferente contenido en un flujo de transmisión (36) o una conexión por protocolo de internet (37).

13. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes,

en el que el generador de flujos de datos (9) está configurado para agregar, en el flujo de datos de salida (10), datos de relleno (18) en un tipo de paquetes de datos de relleno, en el que se determina una cantidad de datos de relleno (18) en base a un requerimiento de tasa de datos determinado por una interfaz de salida del procesador de datos de audio.

14. Un procesador de datos de audio (1) de una de las reivindicaciones precedentes que se implementa como un primer dispositivo separado (19) que está separado de un segundo dispositivo (28) que está configurado para recibir los datos de audio procesados, pero aún codificados, del primer dispositivo (19) para decodificar dichos datos de audio, en el que la interfaz de receptor (2) forma una entrada (20) al primer dispositivo separado (19) a través de una conexión por cable (21) o inalámbrica (22), en el que el procesador de datos de audio (1) comprende además una interfaz de salida (23) conectada al generador de flujos de datos (9), estando la interfaz de salida (23) configurada para emitir el flujo de datos de salida (10), en el que la interfaz de salida (23) lleva a cabo una salida del primer dispositivo separado (19) y comprende una interfaz inalámbrica (24) o un conector de cable (25).

15. Un método para procesar datos de audio en paquetes, comprendiendo el método:

recibir datos de audio codificados (3) que comprenden objetos de audio (11, 12) y metadatos (4) relacionados con los objetos de audio (11, 12);
analizar los metadatos (4) para determinar una posibilidad de manipulación de datos de audio de los objetos de audio (11, 12);
recibir una entrada de interacción (7) y generar, a partir de la entrada de interacción (7), datos de control de la interacción (8) relacionados con la posibilidad de manipulación de datos de audio para manipular los objetos de audio (11, 12) externamente desde un decodificador; y
obtener los datos de control de la interacción (8) y los datos de audio codificados (3) y los metadatos (4) y generar un flujo de datos de salida (10) agregando una porción de datos adicional que contiene los datos de control de la interacción (8) al flujo de datos de salida (10), donde los metadatos originales (4) y los datos de audio codificados (3) no son modificados, de tal manera que el flujo de datos de salida (10) es de nuevo un flujo de audio codificado válido que comprende los datos de audio aún codificados (3), los metadatos (4) y los datos de control de la

interacción (8) agregados.

16. Programa de ordenador para llevar a cabo, cuando se ejecuta en un ordenador o un procesador, el método de procesar datos de audio de la reivindicación 15.

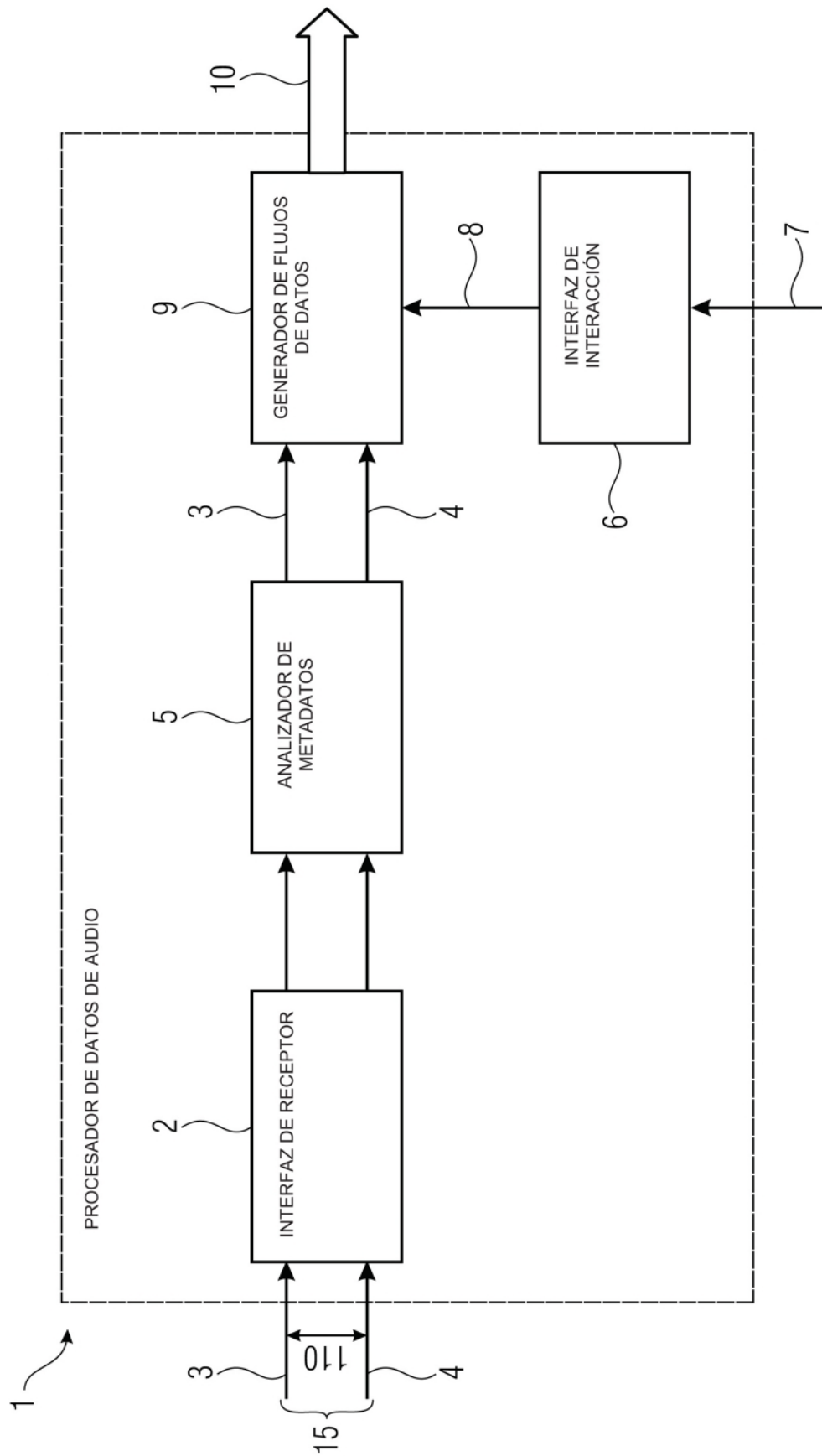


FIG. 1

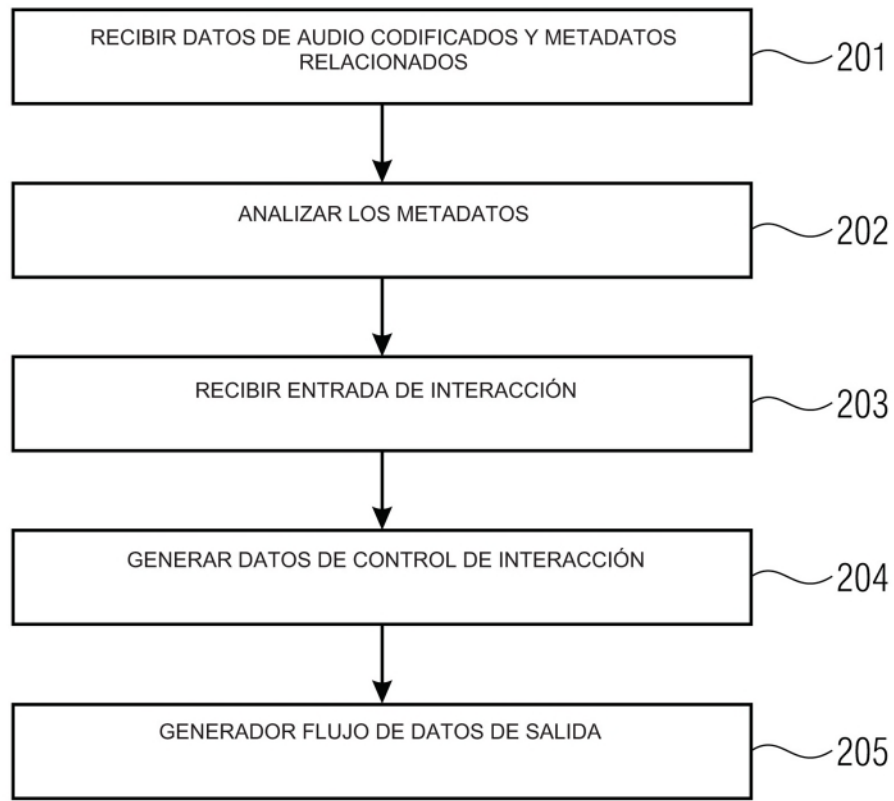


FIG. 2

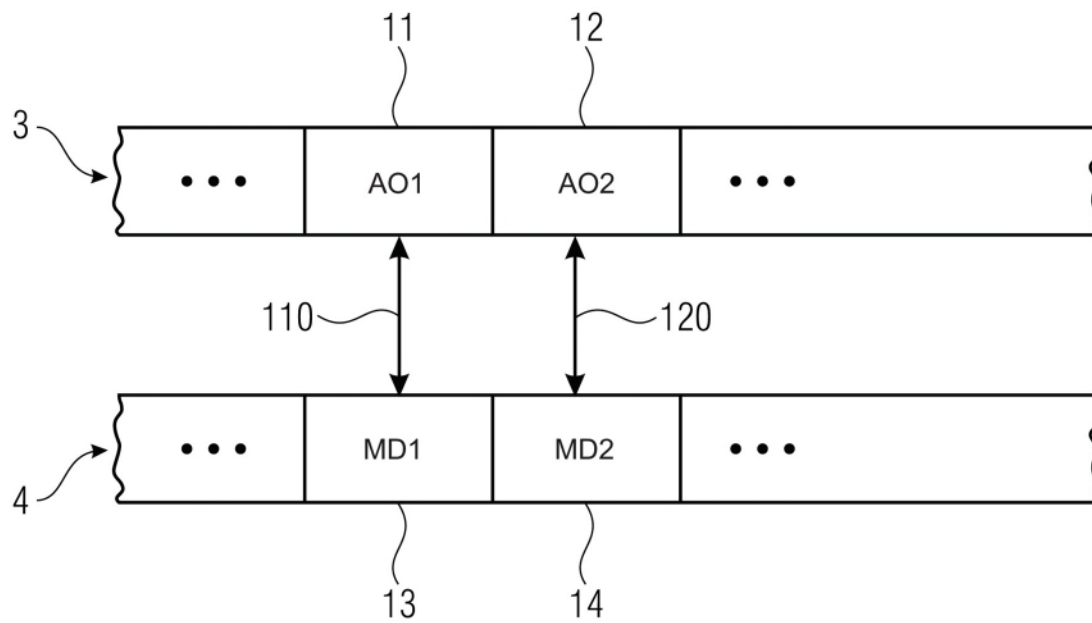


FIG. 3

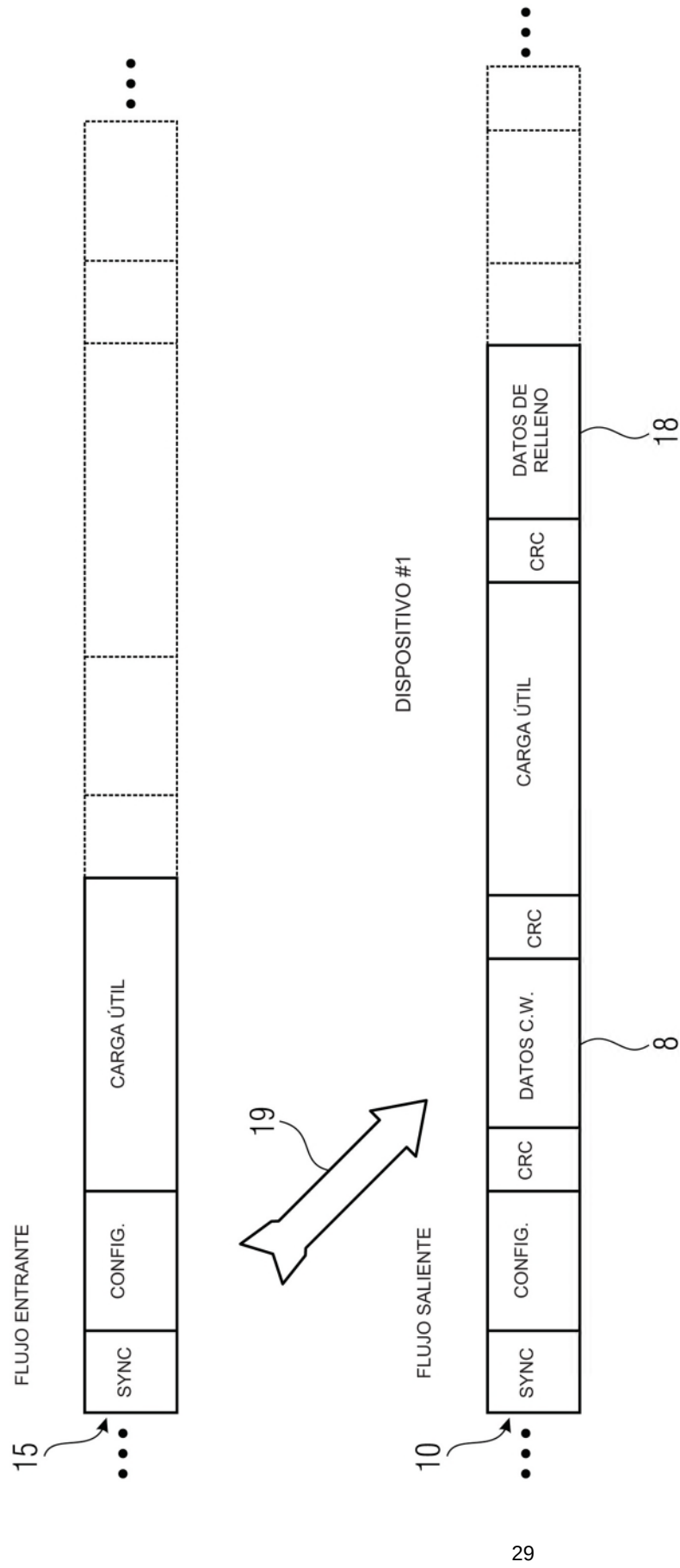


FIG. 4

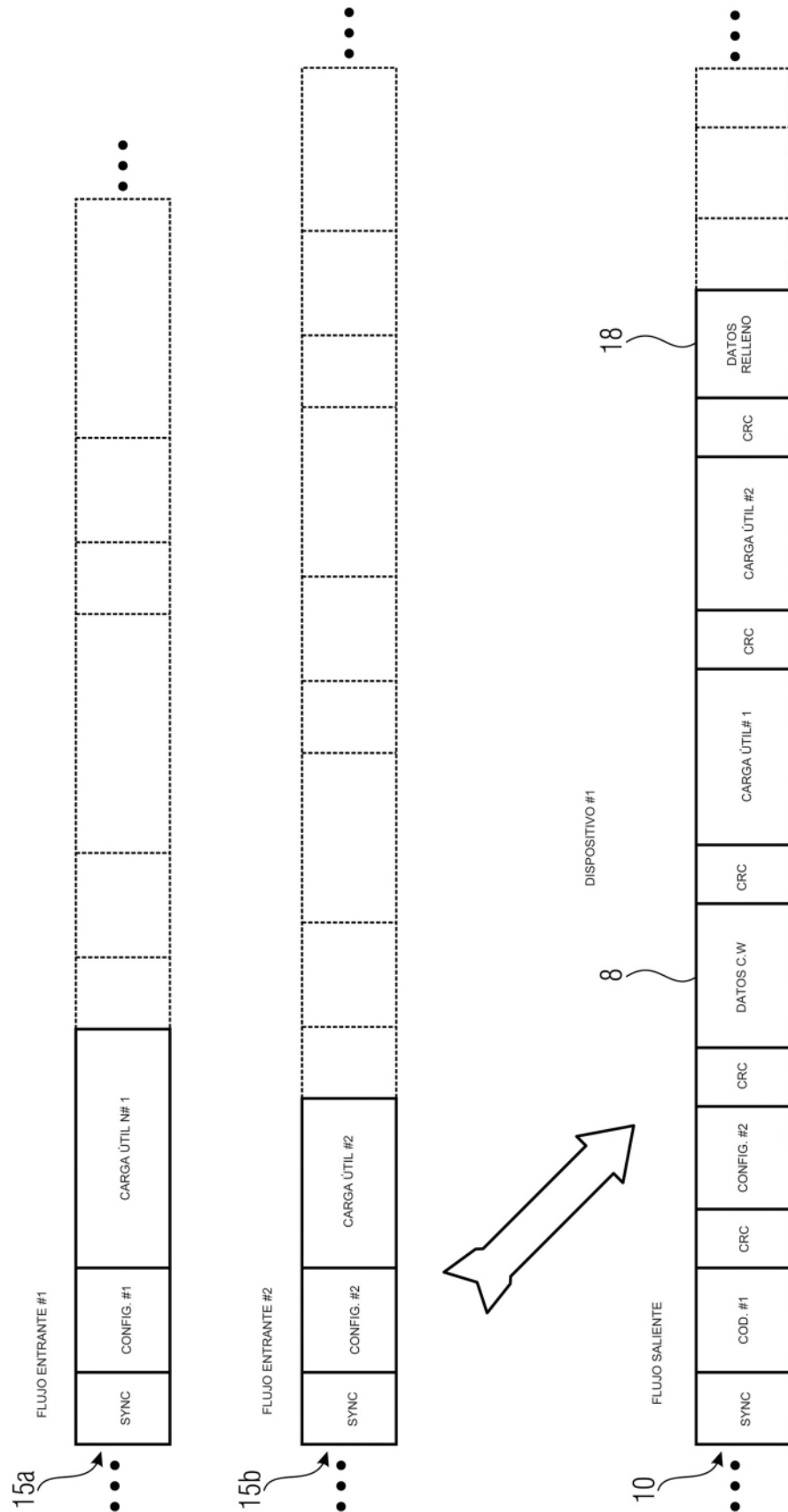


FIG. 5

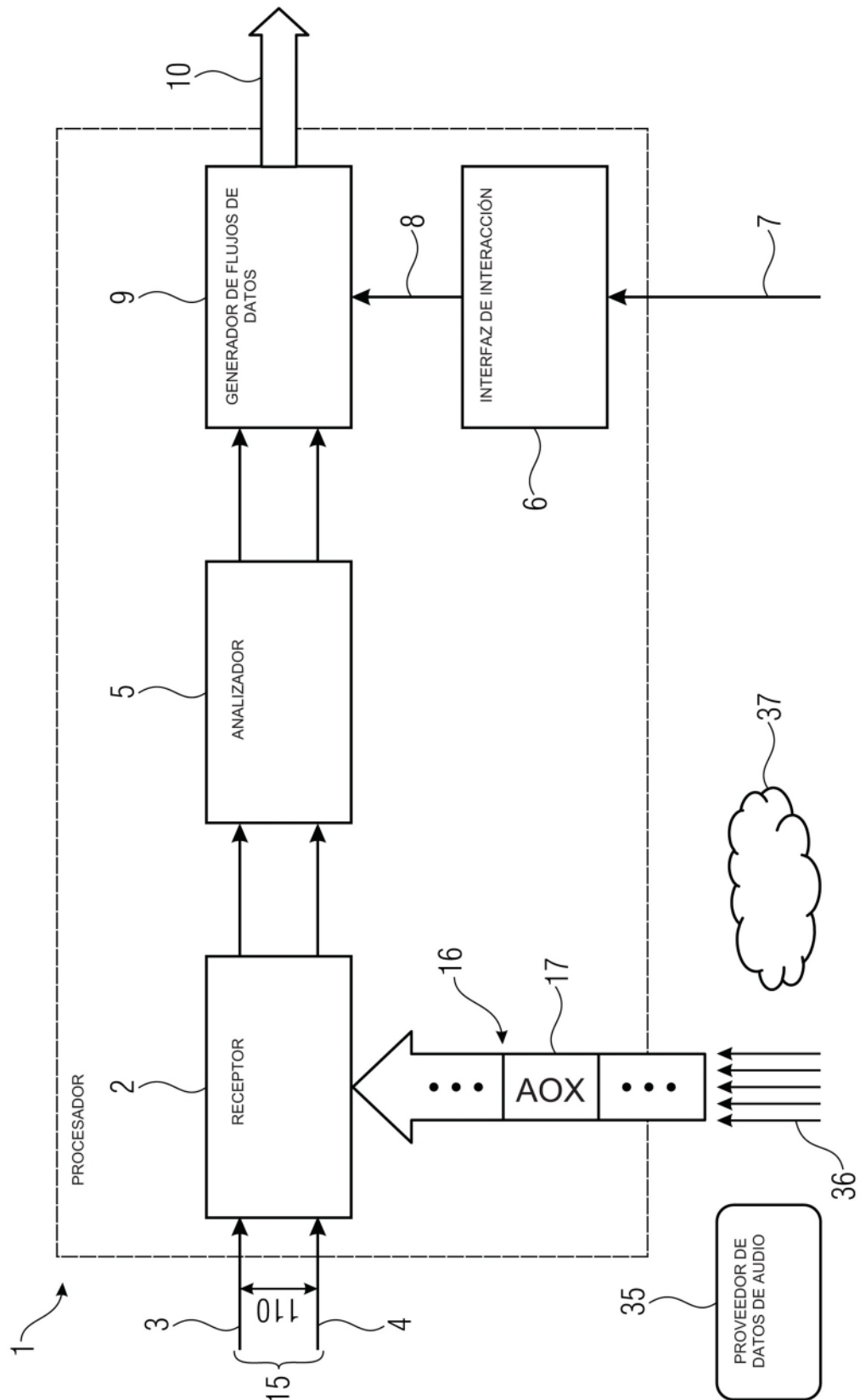


FIG. 6

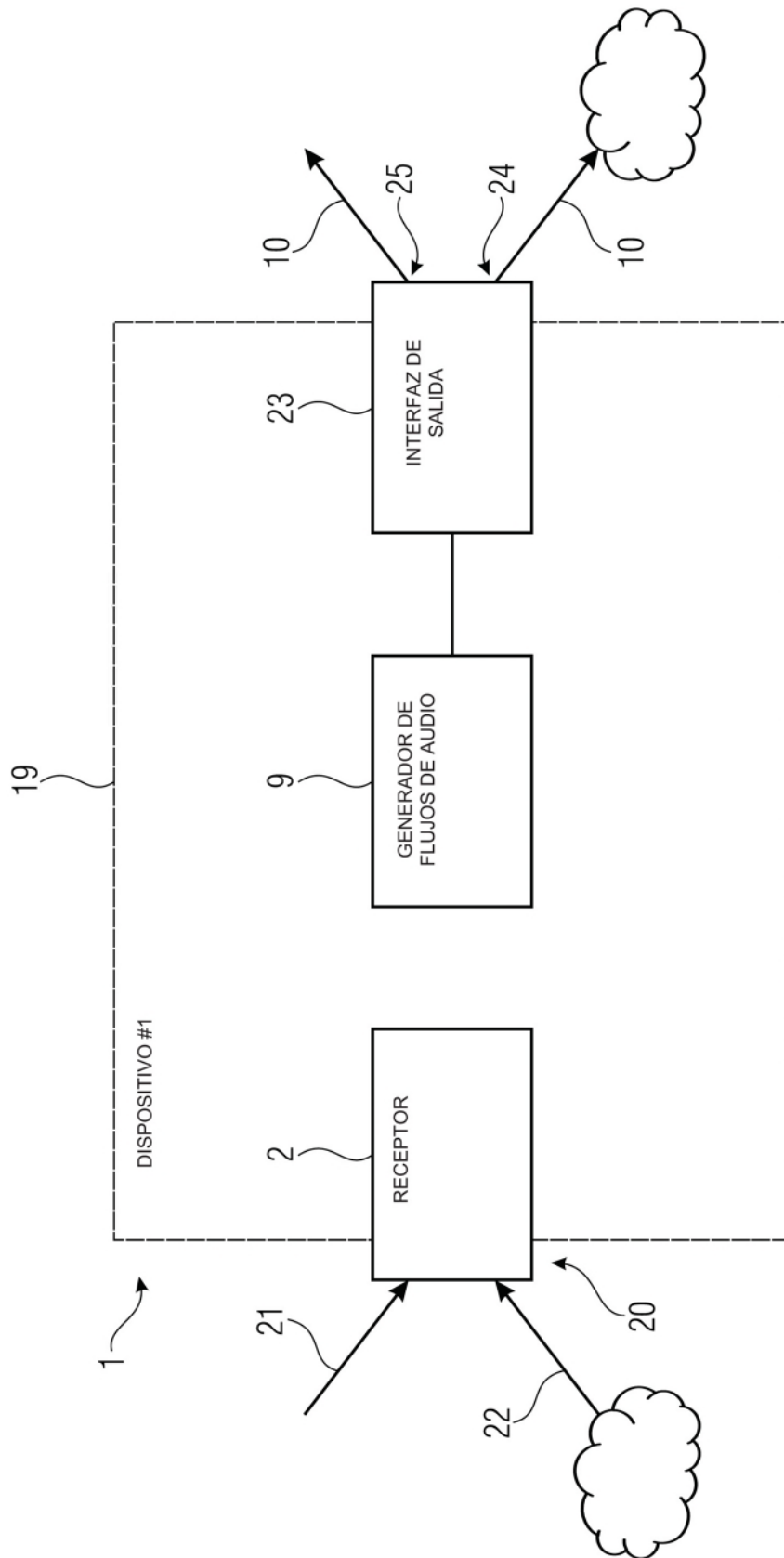


FIG. 7

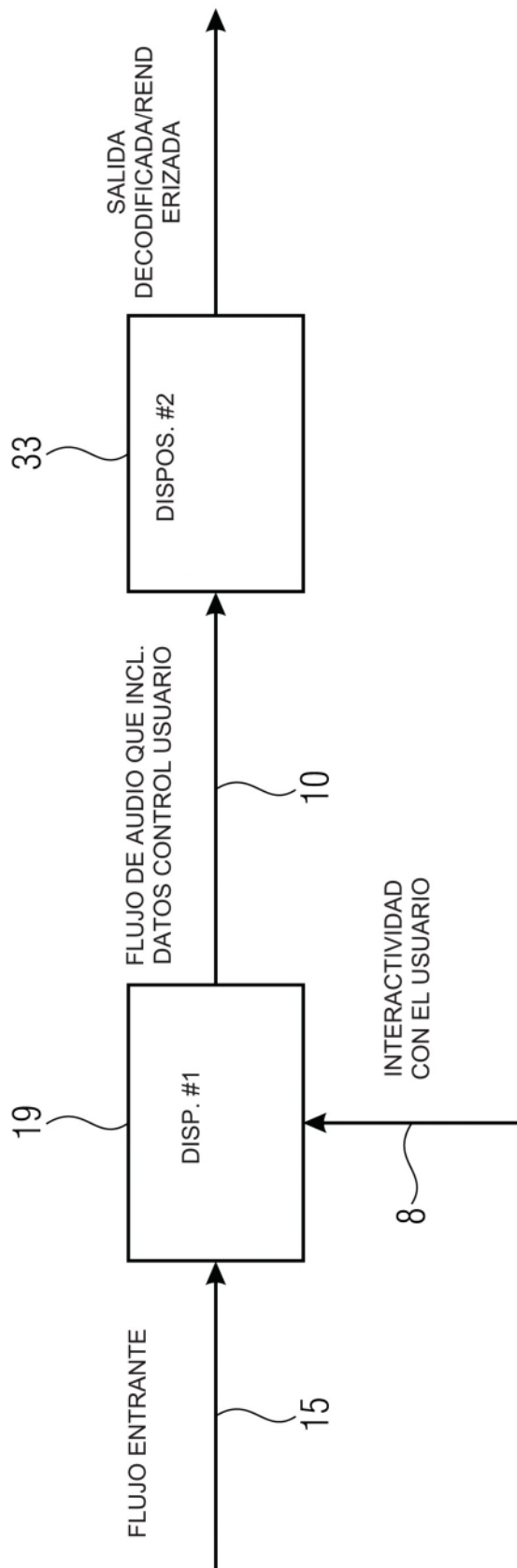


FIG. 8

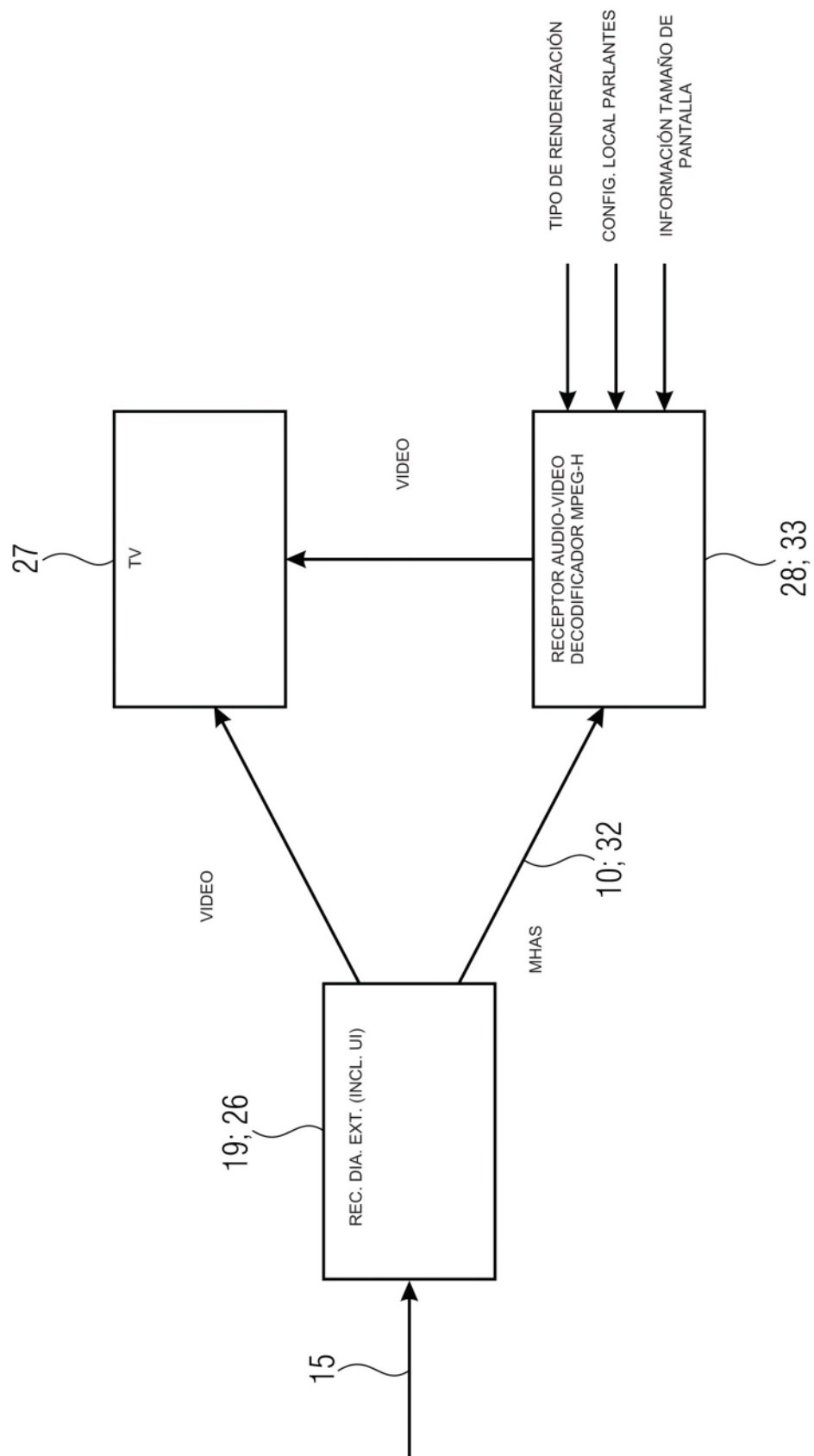


FIG. 9