(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 786 198**

(51) Int. CI.:

G10L 19/038 (2013.01)
G10L 19/07 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA LIMITADA T7

(86) Fecha de presentación y número de la solicitud internacional: **17.12.2013 PCT/IB2013/061034**

(87) Fecha y número de publicación internacional: **25.06.2015 WO15092483**

(96) Fecha de presentación y número de la solicitud europea: **17.12.2013 E 13899497 (5)**

(97) Fecha y número de publicación de la concesión europea modificada tras limitación: **02.04.2025 EP 3084761**

(54) Título: **Codificador de señal de audio**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente limitada:
13.06.2025

(73) Titular/es:

NOKIA TECHNOLOGIES OY (100.00%)
Karakaari 7
02610 Espoo, FI

(72) Inventor/es:

VASILACHE, ADRIANA;
RAMO, ANSSI, SAKARI y
LAAKSONEN, LASSE, JUHANI

(74) Agente/Representante:

DEL VALLE VALIENTE, Sonia

DESCRIPCIÓN

Codificador de señal de audio

Campo

La presente solicitud se refiere a un codificador de señal de audio y, en particular, pero no exclusivamente a un codificador de señal de audio para su uso en un aparato portable.

Antecedentes

Señales de audio, como voz o música, se codifican, por ejemplo, para habilitar una transmisión o almacenamiento eficiente de las señales de audio. Ejemplos de esquemas de codificación de señal de audio se divulgan en los documentos WO 2013/005065A1 y US 2011/0137645 A1.

Codificadores y decodificadores de audio (también conocidos como códecs) se usan para representar señales basadas en audio, tales como música y sonidos ambiente (que en términos de codificación de voz pueden llamarse ruido de fondo). Estos tipos de codificadores habitualmente no utilizan un modelo de voz para el proceso de codificación, sino que usan procesos para representar todos los tipos de señales de audio, incluyendo voz. Codificadores y decodificadores de voz (códecs) pueden considerarse como códecs de audio que se optimizan para señales de voz y pueden operar o bien a una tasa fija o bien una tasa variable.

Codificadores y decodificadores de audio a menudo se diseñan como codificadores de fuente de baja complejidad. En otras palabras, capaces de realizar codificación y decodificación de señales de audio sin requerir procesamiento altamente complejo.

Un ejemplo del cual es la codificación por transformación. Para codificación de audio de señal de música la codificación por transformación generalmente funciona mejor que la tecnología de Predicción Lineal con Excitación por Código Algebraico (ACELP) que se ajusta y dirige mejor para señales de voz. La codificación por transformación se realiza codificando vectores de coeficientes de transformación de tipo subbanda. En otras palabras una señal de audio se divide en subbandas para las que se determina un parámetro y los parámetros representan subvectores que se cuantifican por vector o en rejilla.

Resumen

Según un primer aspecto se proporciona un método implementado por procesador para codificar al menos una señal de audio, como se expone en la reivindicación independiente 1.

Según un segundo aspecto se proporciona un aparato que comprende hardware de procesamiento para implementar la codificación de al menos una señal de audio, expuesto en la reivindicación independiente 5.

La invención se expone en las reivindicaciones independientes. Todas las ocurrencias de la palabra “realización o realizaciones”, si hace referencia a combinaciones de características diferentes de las definidas por las reivindicaciones independientes, se refieren a ejemplos que se presentaron originalmente, pero que no representan realizaciones de la invención reivindicada en la actualidad; estos ejemplos se muestran aún para propósitos de ilustración únicamente.

Breve descripción de los dibujos

Para un mejor entendimiento de la presente invención, se hará ahora referencia a modo de ejemplo a los dibujos adjuntos en los que:

la Figura 1 muestra esquemáticamente un dispositivo electrónico que emplea algunas realizaciones;

la Figura 2 muestra esquemáticamente un sistema de códec de audio según algunas realizaciones;

la Figura 3 muestra esquemáticamente un codificador como se muestra en la Figura 2 según algunas realizaciones;

la Figura 4 muestra un diagrama de flujo que ilustra la operación del codificador mostrado en la Figura 3 según algunas realizaciones;

la Figura 5 muestra esquemáticamente un cuantificador de vector en rejilla como se muestra en la Figura 3 según algunas realizaciones; y

la Figura 6 muestra un diagrama de flujo que ilustra la operación del cuantificador de vector en rejilla mostrado en la Figura 5 según algunas realizaciones.

Descripción de algunas realizaciones de la aplicación

Lo siguiente describe en más detalle posibles códecs de voz y audio multicanal y estéreos, incluyendo códecs de voz y audio de tasa variable por capas o escalable.

Puede existir un problema con enfoques de codificación por transformación actuales en que el uso de retículas eficientes de compresión puede mejorar significativamente la cuantificación. Sin embargo, logran producir tales mejoras con el coste de complejidad de códec significativa.

El concepto como se analiza en detalle por las realizaciones en la presente memoria propone un enfoque que permite una reducción de complejidad de codificación significativa evaluando la distorsión de cuantificación en un espacio de vector transpuesto.

En este sentido se hace referencia primero a la Figura 1 que muestra un diagrama de bloques esquemático de un dispositivo electrónico ilustrativo o aparato 10, que puede incorporar un códec según una realización de la aplicación.

El aparato 10 puede ser, por ejemplo, un terminal móvil o equipo de usuario de un sistema de comunicación inalámbrica. En otras realizaciones el aparato 10 puede ser un dispositivo de audio y video tal como cámara de vídeo, un receptor de televisión (TV), grabador de audio o reproductor de audio tal como un grabador/reproductor de mp3, un grabador de medios (también conocido como un grabador/reproductor de mp4), o cualquier ordenador adecuado para el procesamiento de señales de audio.

El dispositivo electrónico o aparato 10, en algunas realizaciones, comprende un micrófono 11, que se enlaza a través de un convertidor de analógico a digital (ADC) 14 a un procesador 21. El procesador 21 se enlaza adicionalmente a través de un convertidor de digital a analógico (DAC) 32 a altavoces 33. El procesador 21 se enlaza adicionalmente a un transceptor (RX/TX) 13, a una interfaz de usuario (UI) 15 y a una memoria 22.

El procesador 21, en algunas realizaciones, puede configurarse para ejecutar diversos códigos de programa. Los códigos de programa implementados, en algunas realizaciones, comprenden un código de codificación o decodificación de audio como se describe en la presente memoria. Los códigos de programa 23 implementados, en algunas realizaciones, pueden almacenarse, por ejemplo, en la memoria 22 para recuperación por el procesador 21 siempre que se necesite. La memoria 22 podría proporcionar adicionalmente una sección 24 para almacenar datos, por ejemplo, datos que se han codificado según la aplicación.

El código de codificación y decodificación en las realizaciones puede implementarse al menos parcialmente en hardware y/o firmware.

La interfaz de usuario (UI) 15 habilita que un usuario introduzca comandos al dispositivo electrónico 10, por ejemplo, a través de un teclado numérico, y/o para obtener información del dispositivo electrónico 10, por ejemplo, a través de un visualizador. En algunas realizaciones, una pantalla táctil puede proporcionar tanto funciones de entrada como de salida para la interfaz de usuario. El aparato 10, en algunas realizaciones, comprende un transceptor (RX/TX) 13 adecuado para habilitar comunicación con otro aparato, por ejemplo, a través de una red de comunicación inalámbrica.

El transceptor 13 puede comunicarse con dispositivos adicionales por cualquier protocolo de comunicaciones conocido adecuado, por ejemplo, en algunas realizaciones, el transceptor 13 o medio de transceptor puede usar un protocolo de Sistema Universal de Telecomunicaciones Móviles (UMTS) adecuado, un protocolo de red de área local inalámbrica (WLAN) tal como, por ejemplo, IEEE 802.X, un protocolo de comunicación de frecuencia de radio de corto alcance adecuado, tal como Bluetooth, o comunicación de datos trayectoria de infrarrojos (IRDA).

Debe apreciarse de nuevo que la estructura del aparato 10 podría suplementarse y variarse de muchas formas.

Un usuario del aparato 10, por ejemplo, puede usar el micrófono 11 para introducir voz u otras señales de audio que tienen que transmitirse a algún otro aparato o que tienen que almacenarse en la sección de datos 24 de la memoria 22. Una aplicación correspondiente, en algunas realizaciones, puede activarse para este fin por el usuario a través de la interfaz de usuario 15. Esta aplicación en estas realizaciones puede realizarse por el procesador 21, provoca que el procesador 21 ejecute el código de codificación almacenado en la memoria 22. Aunque en los siguientes ejemplos el micrófono 11 se configura para generar las señales de audio a introducir, se entendería que las señales de audio de entrada pueden recibirse desde cualquier entrada adecuada tal como desde la memoria 22 y específicamente dentro de la sección datos almacenados 24 de la memoria 22. En algunas realizaciones, la señal de audio de entrada o al menos una señal de audio puede recibirse a través del transceptor 13. Por ejemplo, el transceptor 13 puede configurarse para recibir señales de audio generadas por micrófonos externos al aparato 10, por ejemplo, un dispositivo Bluetooth acoplado al aparato a través del transceptor 13.

El convertidor de analógico a digital (ADC) 14, en algunas realizaciones, convierte la señal de audio analógica de entrada a una señal de audio digital y proporciona la señal de audio digital al procesador 21. En algunas realizaciones,

el micrófono 11 puede comprender un micrófono integrado y función de ADC y proporcionar señales de audio digitales directamente al procesador para procesamiento.

El procesador 21, en tales realizaciones, procesa a continuación la señal de audio digital de la misma manera que se describe con referencia al sistema mostrado en la Figura 2, y específicamente el codificador mostrado en las Figuras 3, y detalles del codificador mostrado en la Figura 5.

El flujo de bits resultante puede proporcionarse, en algunas realizaciones, al transceptor 13 para transmisión a otro aparato. Como alternativa, los datos de audio codificados, en algunas realizaciones, pueden almacenarse en la sección de datos 24 de la memoria 22, por ejemplo, para una transmisión posterior o para una presentación posterior por el mismo aparato 10.

El aparato 10, en algunas realizaciones, también puede recibir un flujo de bits con datos codificados en consecuencia desde otro aparato a través del transceptor 13. En este ejemplo, el procesador 21 puede ejecutar el código de programa de decodificación almacenado en la memoria 22. El procesador 21, en tales realizaciones, decodifica los datos recibidos, y proporciona los datos decodificados a un convertidor de digital a analógico 32. El convertidor de digital a analógico 32 convierte los datos digitales decodificados en datos de audio analógicos y, en algunas realizaciones, puede emitir el audio analógico a través de los altavoces 33. Ejecución del código de programa de decodificación, en algunas realizaciones, puede desencadenarse también mediante una aplicación llamada por el usuario a través de la interfaz de usuario 15.

Los datos recibidos codificados, en alguna realización, también pueden almacenarse en lugar de una presentación inmediata a través de los altavoces 33 en la sección de datos 24 de la memoria 22, por ejemplo, para una decodificación y presentación posteriores o decodificación y reenvío a aún otro aparato.

Se apreciaría que las estructuras esquemáticas descritas en las Figuras 3 y 5 y las etapas de método mostradas en las Figuras 4 y 6 representan únicamente una parte de la operación de un códec de audio y específicamente parte de un aparato de codificador de audio o método como se muestra ilustrativamente implementado en el aparato mostrado en la Figura 1.

La operación general de códecs de audio según se emplean por realizaciones se muestra en la Figura 2. Sistemas de codificación/decodificación de audio generales comprenden tanto un codificador como un decodificador, como se ilustra esquemáticamente en la Figura 2. Sin embargo, se entendería que algunas realizaciones pueden implementar uno del codificador o decodificador, o ambos del codificador y decodificador. La Figura 2 ilustra un sistema 102 con un codificador 104, un canal de medios o almacenamiento 106 y un decodificador 108, se entendería que como se ha descrito anteriormente algunas realizaciones pueden comprender o implementar uno del codificador 104 o ambos del codificador 104 y decodificador 108.

El codificador 104 comprime una señal de audio de entrada 110 que produce un flujo de bits 112, que, en algunas realizaciones, puede almacenarse o transmitirse a través de un canal de medios 106. El codificador 104, en algunas realizaciones, puede comprender un codificador multicanal que codifica dos o más señales de audio.

El flujo de bits 112 puede recibirse dentro del decodificador 108. El decodificador 108 descomprime el flujo de bits 112 y produce una señal de audio de salida 114. El decodificador 108 puede comprender un decodificador de transformación como parte de la operación de decodificación general. El decodificador 108 también puede comprender un decodificador multicanal que decodifica dos o más señales de audio. La tasa de bits del flujo de bits 112 y la calidad de la señal de audio de salida 114 en relación con la señal de entrada 110 son las principales características que definen el rendimiento del sistema de codificación 102.

La Figura 3 muestra esquemáticamente el codificador 104 según algunas realizaciones.

La Figura 4 muestra esquemáticamente en un diagrama de flujo la operación del codificador 104 según algunas realizaciones.

El concepto para las realizaciones como se describe en la presente memoria es para determinar y aplicar codificación a señales de audio para producir codificación real de tasa de bits baja y alta calidad eficiente. A este respecto, con respecto a la Figura 3, se muestra un codificador 104 de ejemplo según algunas realizaciones. Además, con respecto a la Figura 4, la operación del codificador 104 se muestra en detalle adicional. En los siguientes ejemplos el codificador se configura para generar parámetros de dominio de frecuencia que representan la señal de audio y codificar los parámetros de dominio de frecuencia generados usando una cuantificación en rejilla de vector adecuada, sin embargo se entendería que, en algunas realizaciones, los parámetros usados en la cuantificación en rejilla como se describe en la presente memoria pueden ser cualquier parámetro adecuado que define o representa las señales de audio u otro tipo de señales (por ejemplo, imagen o video).

El codificador 104, en algunas realizaciones, comprende un seccionador de trama 201 o medio adecuado para seccionar la señal de audio. El seccionador de trama 201 se configura para recibir las señales de audio (por ejemplo

una representación de audio mono, estéreo de izquierda y derecha o cualquier representación de audio multicanal), señal de audio de entrada y sección o segmento de la señal de datos de audio en secciones o tramas adecuadas para una transformación de frecuencia o de otro dominio. El seccionador de trama 201, en algunas realizaciones, puede configurarse adicionalmente para formar en ventana estas tramas o secciones de datos de señal de audio según cualquier función de formación de ventana adecuada. Por ejemplo, el seccionador de trama 201 puede configurarse, en algunas realizaciones, para generar tramas de 20 ms que solapan tramas anteriores o siguientes por 10 ms cada una.

La operación de generación de tramas de audio se muestra en la Figura 4 mediante la etapa 501.

En algunas realizaciones, las tramas de audio pueden pasarse a un determinador 203 de parámetros.

En algunas realizaciones, el codificador comprende un determinador 203 de parámetros de medios adecuados para determinar al menos un parámetro que representa las tramas de señal o señales de audio de entrada o de señal de audio de entrada. En los siguientes ejemplos el parámetro es un parámetro de Frecuencia Espectral de Línea (LSF) sin embargo se entendería que, en algunas realizaciones, puede determinarse cualquier parámetro adecuado.

Por ejemplo, en algunas realizaciones, el determinador de parámetro comprende un transformador 203 o medio adecuado para transformar. El transformador 203, en algunas realizaciones, se configura para generar representaciones de parámetro de dominio de frecuencia (u otro dominio adecuado) de estas señales de audio. Estas representaciones de parámetros de dominio de frecuencia pueden, en algunas realizaciones, pasarse al codificador de parámetro 205.

En algunas realizaciones, el transformador 203 puede configurarse para realizar cualquier transformación adecuada de dominio de tiempo a frecuencia en la señal de datos de audio. Por ejemplo, la transformación de dominio de tiempo a frecuencia puede ser una transformada de Fourier discreta (DFT), transformada rápida de Fourier (FFT), transformada de coseno discreta modificada (MDCT). En los siguientes ejemplos se usa una Transformada Rápida de Fourier (FFT).

Además, el transformador puede configurarse adicionalmente para generar representaciones de parámetro de dominio de banda de frecuencia separadas (representaciones de parámetro de subbanda) de cada canal de entrada datos de señal de audio. Estas bandas pueden disponerse de cualquier modo adecuado. Por ejemplo, estas bandas pueden espaciarse linealmente o asignarse porcentual y psicoacústicamente. Los parámetros generados pueden ser cualquier parámetro adecuado.

La operación de determinación o generación de representaciones de parámetro se muestra en la Figura 4 mediante la etapa 503.

En algunas realizaciones, las representaciones, tal como parámetros LSF, se pasan a un codificador de parámetro 205.

En algunas realizaciones, el codificador 104 puede comprender un codificador de parámetro 205. El codificador de parámetro 205 puede configurarse para recibir las representaciones de parámetro de la entrada de señal de audio, por ejemplo, los parámetros LSF determinados. El codificador de parámetro 205, en algunas realizaciones, puede configurarse además para usar cada uno de los valores de parámetros LSF como un subvector y combinar cada subvector para crear un vector a introducir en un cuantificador de vector. En otras palabras, el aparato puede comprender un generador de vectores configurado para generar un primer vector de parámetros (o tuplas de un primer vector que representa los parámetros) que definen al menos una señal de audio.

La salida del cuantificador de vector es, en algunas realizaciones, el codificador y, por lo tanto, las salidas de señales de audio de vector cuantificado son las representaciones de la señal de audio 'codificadas' o codificadas de parámetro.

La operación de codificación o cuantificación vectorial de los parámetros se muestra en la Figura 4 mediante la etapa 505.

En algunas realizaciones, el codificador de parámetro 205 comprende un Generador 451 de vectores. El Generador 451 de vectores se configura para recibir los parámetros de LSF y generar un vector de N dimensiones a partir de estos valores.

La operación de generación de vectores a partir de los parámetros de entrada se muestra en la Figura 4 mediante la subetapa 551.

Los vectores generados, en algunas realizaciones, pueden pasarse al Cuantificador 453 de vector en rejilla.

En algunas realizaciones, el codificador de parámetro 205 comprende un Cuantificador 453 de vector en rejilla. El Cuantificador 453 de vector en rejilla recibe el vector de entrada generado a partir de los parámetros de LSF y genera

una salida de vecino más cercano o NN que se produce dentro de una rejilla definida y, por lo tanto, puede codificarse usando una rejilla similar en el decodificador.

La operación de cuantificación en rejilla del vector se muestra en la Figura 4 mediante la subetapa 553.

La señal codificada puede emitirse.

La operación de emisión de la señal codificada se muestra en la Figura 4 mediante la etapa 507. Esto para el ejemplo puede ser una operación de emisión de vector cuantificado en rejilla como se muestra en la Figura 4 mediante la subetapa 557.

Con referencia a la Figura 5, se muestra un Cuantificador 453 de vector en rejilla de ejemplo según algunas realizaciones. El cuantificador en rejilla 453, en algunas realizaciones, puede definirse mediante el respectivo código de programa 23 de un programa informático que se almacena en una memoria de medio de almacenamiento tangible 22.

Antes de introducir los conceptos y realizaciones con respecto a la invención, analizaremos inicialmente la cuantificación vectorial en rejilla convencional. En algunos cuantificadores en rejilla, se realiza una generación o determinación inicial de un conjunto de vectores de código base potenciales, en la que cada vector de código base potencial determinado de este conjunto de vectores de código base potenciales se asocia con un vector de código base potencial de un conjunto diferente de vectores de código base.

Cada conjunto de vectores de código base potenciales comprende al menos un vector de código base. Ya que cada conjunto de vectores de código base se asocia con al menos un representante de escala de una pluralidad de representantes de escala, puede determinarse un vector de código basándose en un vector de código base de un conjunto de vectores de código base potenciales y un representante de escala del al menos un representante de escala asociado con el conjunto de vectores de código base potenciales. En otras palabras el vector de código puede representarse basándose en un vector de código base escalado por el respectivo representante de escala. Por ejemplo, el representante de escala puede representar un valor de escala, en donde un vector de código puede determinarse basándose en una multiplicación de un vector de código base y el respectivo valor de escala. Además, en algunas realizaciones, el libro de códigos se obtiene aplicando una permutación (con signo) del vector base.

Por ejemplo, al menos un conjunto de vectores de código base se asocia con al menos dos representantes de escala.

Por consiguiente, como un ejemplo, un libro de códigos puede comprender un conjunto de vectores de código que comprende vectores de código basándose en la pluralidad de conjuntos de vectores de código base y basándose en el respectivo al menos un valor de escala asociado con un respectivo conjunto de vectores de código base de la pluralidad de vectores de código base. Este conjunto de vectores de código puede comprender, para cada vector de código base de cada conjunto de vectores de código base y para cada uno del al menos un representante de escala asociado con un respectivo conjunto de vectores de código base, un vector de código basándose en el respectivo vector de código base escalado por el respectivo representante de escala.

Por ejemplo, dichos conjuntos de vectores de código base pueden representar clases guía, en donde cada clase guía comprende un vector guía diferente y permutaciones de dicho vector guía. Por lo tanto, dicho vector guía y las permutaciones de dicho vector guía pueden representar los vectores de código base del respectivo conjunto de vectores de código base.

La pluralidad de conjuntos de vectores de código base puede representar un subconjunto de una segunda pluralidad de conjuntos de vectores de código base. Por ejemplo, bajo la suposición de que cada conjunto de vector de código base representa una clase guía, la pluralidad de clases guía pueden representar un subconjunto de una segunda pluralidad de clases guía. Por lo tanto, la pluralidad de clases guía puede considerarse como una pluralidad truncada de clases guía con respecto a la segunda pluralidad de clases guía.

Por ejemplo, el respectivo vector de código base potencial puede determinarse determinando el vector de código base del al menos un vector de código base del respectivo conjunto de vector de código base que está más cercano al vector de entrada a codificar. Puede usarse cualquier tipo de criterio adecuado para encontrar el vector de código base más cercano con respecto al vector de entrada a codificar.

Como un ejemplo, un vector de código base potencial puede determinarse basándose en un vector de código base más cercano con respecto al vector de entrada de valor absoluto y basándose en información de signos de los valores del vector de entrada, en donde esta información puede comprender el signo de una respectiva posición de respectivos valores en el vector de entrada y se usa para asignar signos a valores del vector de código base potencial determinado. Adicionalmente, como un ejemplo, puede determinarse el vector de código base que está más cercano al vector de entrada de valor absoluto, en donde el vector de entrada de valor absoluto comprende valores absolutos que corresponden a los valores del vector de entrada, en donde el vector de código base potencial representa el vector de código base más cercano determinado, en donde los signos de los valores del vector de código base potencial

corresponden a los signos de los valores del vector de entrada en la misma posición en el vector, en donde esto puede ser válido si la paridad de los vectores de código base del conjunto de vectores de código base es 0. Como otro ejemplo, si la paridad de los vectores de código base del conjunto de vectores de código base es -1, los signos de los valores del vector de código base potencial pueden asignarse correspondiendo a los signos de los valores del vector de entrada en la misma posición en el vector, respectivamente, y si no hay un valor impar de componentes negativos, el valor en el vector de código base potencial que tiene el valor absoluto no nulo más bajo puede cambiar su signo. O, como otro ejemplo, si la paridad de los vectores de código base del conjunto de vectores de código base es +1, los signos de los valores del vector de código base potencial pueden asignarse correspondiendo a los signos de los valores del vector de entrada en la misma posición en el vector, respectivamente, y si no hay un valor par de componentes negativos, el valor en el vector de código base potencial que tiene el valor absoluto no nulo más bajo puede cambiar su signo.

El vector de código para codificar el vector de entrada se determina a continuación convencionalmente basándose en el conjunto de vectores de código potenciales determinados, en donde dicho conjunto de vectores de código potenciales determinados define un subconjunto de vectores de código, comprendiendo dicho subconjunto de vectores de código, para cada vector de código base potencial determinado y cada representante de escala asociado con el conjunto de vectores de código base del respectivo vector de código base potencial, un vector de código basándose en el respectivo vector de código base potencial escalado por el respectivo representante de escala.

Por consiguiente, la búsqueda del vector de código para codificar el vector de entrada se ha realizado en el subconjunto de vectores de código definidos por los vectores de código potenciales determinados y definidos por el respectivo al menos un representante de escala asociado con el conjunto de vectores de código base del respectivo vector de código potencial determinado. Ya que este subconjunto de vectores de código puede representar un subconjunto de vectores de código asociados con el libro de códigos, el número de vectores de código de este subconjunto de vectores de código puede ser menor que el número de vectores de código del conjunto de vectores de código.

Como un ejemplo, cada representante de escala de la pluralidad de representantes de escala puede asociarse con al menos un conjunto de vectores de código, en donde cada conjunto de vectores de código de dicho al menos un conjunto de vectores de código asociado con un respectivo representante de escala se asocia con un conjunto de vectores de código base de la pluralidad de conjuntos de vectores de código base de tal forma que cada conjunto de vectores de código de dicho al menos un conjunto de vectores de código asociado con un respectivo representante de escala comprende vectores de código obtenidos escalando los vectores base del respectivo conjunto asociado de vectores base con el respectivo representante de escala.

Por consiguiente, los vectores de código del al menos un conjunto de vectores de código base asociados con un respectivo representante de escala de la pluralidad de representantes de escala puede determinarse basándose en el escalado de los vectores de código base de cada conjunto de vectores de código base asociados con el representante de escala con este representante de escala.

Por ejemplo, en caso de que dichos conjuntos de vectores de código base representen clases guía, el al menos un conjunto de vectores de código base asociados con un respectivo representante de escala puede considerarse como una unión de clases guía. Se entendería que normalmente la unión de clases guía es independiente de la escala. Por lo tanto, el libro de códigos puede comprender al menos una unión de clases guía, en donde cada unión de clase guía se asocia con uno de al menos unos representantes de escala y con al menos un conjunto de vectores de código base de la pluralidad de vectores de código base. Como un ejemplo, el al menos un representante de escala puede representar la pluralidad de representantes de escala que puede comprender al menos dos representantes de escala.

Por lo tanto, por ejemplo, b_x, con $x\epsilon\{0, 1, ... X-1\}$, representa un conjunto de vectores de código base de la pluralidad de conjuntos de vectores de código base, en donde X representa el número de conjuntos de la pluralidad de conjuntos de vectores de código base. Cada conjunto de vectores de código base se asocia o comprende al menos un vector de código base $b_{x,y}$, en donde B_x representa el número de vectores de código base de un respectivo conjunto de vectores de código base b_x, es decir $y\epsilon\{0, 1, ... B_x-1\}$ es válido. Por ejemplo, el número B_x de vectores de código base de un conjunto de vectores de código base puede ser diferente para diferentes conjuntos de vectores de código base y/o puede ser el mismo para al menos dos conjuntos de vectores de código base.

En otras palabras un vector guía es solo un vector. Junto con todas las permutaciones con signo del vector guía, a continuación, este conjunto forma la clase guía del vector guía (o como se describe en la presente memoria los vectores de código base). Cuando se juntan varias clases guía, se forma una unión de clases guía. A continuación, pueden fijarse una o más escalas a esta unión/uniones.

Por lo tanto, por ejemplo, puede ser posible determinar un vector de código $c_{x,z,y}$ basándose en vector de código base $b_{x,y}$ y basándose en un representante de escala s_z, en donde el índice z representa el índice del respectivo representante de escala de la pluralidad de representantes de escala $s_0 ... s_{s-1}$, es decir $ze\{0, 1, ... S-1\}$ es válido.

Por ejemplo, en caso de que los valores $b_{x,y,t}$ de los vectores de código base $b_{x,y}=[b_{x,y,0}, b_{x,y,1}, b_{x,y,n-1}]$ representan valores absolutos, en donde $te\{0, 1, ... n-1\}$ es válido y n representa la longitud del respectivo vector de código base

$b_{x,y}$, y si el vector de entrada de valor absoluto se usa para determinar el vector de código potencial de un respectivo conjunto de vectores de código base, el signo de cada valor $b_{x,y,t}$ en la $(t+1)^{ésima}$ posición del vector de código base más cercano determinado $b_{x,y}$ puede asignarse basándose en el signo del respectivo valor i_t en la $(t+1)^{ésima}$ posición del vector de entrada i, antes de determinar un vector de código $c_{x,z,y}$ basándose en vector de código base $b_{x,y}$ y basándose en un representante de escala s_z se realiza.

Como un ejemplo, si $i=[i_0, i_1, ..., i_{n-1}]$ representa el vector de entrada, el vector de entrada de valor absoluto puede representarse mediante $[|i_0|, |i_1|, ..., |i_{n-1}|]$. Por ejemplo, el signo de cada valor $b_{x,y,t}$ en la $(t+1)^{ésima}$ posición del vector de código base más cercano determinado $b_{x,y}$ puede asignarse al signo del respectivo valor i_t en la $(t+1)^{ésima}$ posición del vector de entrada, respectivamente, en donde esto puede ser válido si la paridad de los vectores de código base $b_{x,y}$ del conjunto de vectores de código base b_x es 0. Como otro ejemplo, si la paridad de los vectores de código base $b_{x,y}$ del conjunto de vectores de código base b_x es -1, los signos de los valores $b_{x,y,t}$ del vector de código base potencial pueden asignarse correspondiendo a los signos de los valores del vector de entrada en la misma posición en el vector, respectivamente, y si no hay un valor impar de componentes negativos, el valor $b_{x,y,t}$ en el vector de código base potencial que tiene el valor absoluto no nulo más bajo puede cambiar su signo. O, como otro ejemplo, si la paridad de los vectores de código base $b_{x,y}$ del conjunto de vectores de código base b_x es +1, los signos de los valores $b_{x,y,t}$ del vector de código base potencial puede asignarse correspondiendo a los signos de los valores del vector de entrada en la misma posición en el vector, respectivamente, y si no hay un valor par de componentes negativos, el valor $b_{x,y,t}$ en el vector de código base potencial que tiene el valor absoluto no nulo más bajo puede cambiar su signo.

Como un ejemplo no limitante, un vector de código $c_{x,z,y}$ puede determinarse mediante $c_{x,z,y} = [b_{x,y,0} \cdot s_z, b_{x,y,1} \cdot s_z, ..., b_{x,y,n-1} \cdot s_z]$.

Cada uno de los representantes de escala s_z, en donde $ze\{0, 1, ... S-1\}$ es válido, se asocia con al menos un conjunto de vectores de código base. Por ejemplo, como un ejemplo no limitante este respectivo al menos un conjunto de vectores de código base puede representarse mediante el conjunto de vectores de código base b_x, con $xe\{0, 1, ... n_z - 1\}$, en donde n_z puede representar el número de conjuntos de vectores de código base asociados con el respectivo representante de escala s_z, en donde $0<n_z<X$ es válido. Basándose en esta vinculación entre un respectivo representante de escala s_z y el al menos un conjunto de vectores de código base asociado b_x, con $xe\{0, 1, ... n_z-1\}$, puede determinarse el al menos un conjunto de vectores de código asociado $c_{x,z,y}$, con $xe\{0, 1, ... n_z-1\}$ e $ye\{0, 1, ... B_x-1\}$ y $ze\{0, 1, ... S-1\}$.

Por lo tanto, como un ejemplo, puede definirse una estructura de libro de códigos del libro de códigos anteriormente mencionado mediante la pluralidad de representantes de escala s_z, la pluralidad de conjuntos de vectores de código base b_x, y la vinculación entre cada representante de escala con el al menos un conjunto de vectores de código base asociado.

Ya que al menos un conjunto de vectores de código base, por ejemplo, al menos el conjunto de vectores de código base b_0, se asocia con al menos dos representantes de escala, el mismo conjunto de vectores de código base puede usarse para construir vectores de código del al menos un conjunto de vectores de código asociados con un primer representante de escala y para construir vectores de código del al menos un conjunto de vectores de código asociados con al menos un representante de escala adicional.

Es posible determinar, para cada conjunto de vectores de código base de una pluralidad de conjuntos de vectores de código base, un vector de código base potencial para codificar un vector de entrada de otras formas.

Por ejemplo, determinar un vector de código para codificar el vector de entrada a partir de un subconjunto de vectores de código se basa en una métrica de distorsión o distancia determinada, o valor de error.

En tales ejemplos, se selecciona una representación de escala de la pluralidad de representaciones de escala.

Además, se selecciona el vector de código base potencial determinado de un conjunto de vectores de código base asociados con la representación de escala seleccionada.

A continuación puede determinarse un vector de código basándose en el vector de código base potencial seleccionado y en la representación de escala seleccionada, en donde esta determinación de un vector de código puede realizarse como se describe con respecto al método descrito en la presente memoria.

En algunos ejemplos, basándose en el vector de código y el vector de entrada determinados, se determina una métrica de distorsión. Por ejemplo, dicha métrica de distorsión puede basarse en cualquier clase de distancia adecuada entre el vector de código y el vector de entrada determinados. Como un ejemplo, puede usarse una distancia de Hamming o una distancia Euclidiana o cualquier otra distancia. Como un ejemplo, puede omitirse la determinación del vector de código y la métrica de distorsión puede calcularse considerando inherentemente el respectivo vector de código asociado con la representación de escala seleccionada y el conjunto de vectores de código base asociados con esta representación de escala seleccionada.

Por ejemplo, si $c_{x,z,y}$=[$c_{x,z,y,0}$, $c_{x,z,1}$, ..., $c_{x,z,n-1}$] representa el vector de código determinado en la etapa 430 e i=[i_0, i_1, ..., i_{n-1}] representa el vector de entrada, una distancia d puede calcularse basándose en

$$d = \sum_{k=0}^{n-1} \left(i_k - c_{x,z,y,k} \right)^2 .$$

Esta distancia d según la ecuación anterior puede sustituirse con distancia d' calculada basándose en

$$d' = \sum_{k=0}^{n-1} c_{x,z,y,k}^2 - 2 \sum_{k=0}^{n-1} i_k \cdot c_{x,z,y,k}$$

O, como otro ejemplo, en caso de que la métrica de distorsión se determina basándose en una función de ponderación, la distancia d según la ecuación anterior puede modificarse como se indica a continuación:

$$d_w = \sum_{k=0}^{n-1} w_k \cdot \left(i_k - c_{x,z,y,k} \right)^2 ,$$

en la que w_k representan factores de ponderación de la función de ponderación.

Por consiguiente, la distancia d' según la ecuación anterior puede ponderarse por medio de la función de ponderación de la siguiente forma:

$$d'_w = \sum_{k=0}^{n-1} w_k \cdot c_{x,z,y,k}^2 - 2 \sum_{k=0}^{n-1} w_k \cdot i_k \cdot c_{x,z,y,k}$$

Por ejemplo, la métrica de distorsión d, o d', o d_w, o d_w' puede almacenarse, si es la primera métrica de distorsión determinada, o puede compararse con una métrica de distorsión almacenada, en donde la métrica de distorsión almacenada se sustituye si la métrica de distorsión recientemente determinada es mejor que la métrica de distorsión almacenada. Adicionalmente, puede almacenarse el vector de código asociado con la métrica de distorsión almacenada o puede almacenarse un identificador de este vector de código.

A continuación, por ejemplo, la operación puede comprobar si existe algún conjunto adicional de vectores de código base asociados con la representación de escala seleccionada. Si es que sí, a continuación se selecciona el vector de código base potencial determinado de este conjunto adicional de vectores de código base asociados con la representación de escala seleccionada. Si es que no, se hace una comprobación contra una representación de escala adicional de la pluralidad de representaciones de escala.

Si existe una representación de escala adicional de la pluralidad de representaciones de escala, a continuación, se selecciona la representación de escala adicional, de lo contrario puede seleccionarse el vector de código asociado con la mejor métrica de distancia para codificar el vector de entrada.

Por ejemplo, donde los conjuntos de vectores de código base pueden representar clases guía, en donde cada clase guía comprende un vector guía diferente y permutaciones de dicho vector guía. Por lo tanto, el vector guía y las permutaciones de dicho vector guía pueden representar los vectores de código base del respectivo conjunto de vectores de código base. Como un ejemplo, un vector guía es un vector de n dimensiones (indicando n un número entero), cuyos componentes (positivos) se ordenan (por ejemplo, en orden decreciente). La clase guía que corresponde al vector guía entonces consiste en el vector guía y todos los vectores obtenidos a través de todas las permutaciones con signo del vector guía (con algunas restricciones posibles).

Una unión de clases guía puede definirse mediante los conjuntos de vectores de código base asociados con la misma representación de escala de la pluralidad de representaciones de escala y la respectiva representación de escala. Por ejemplo, una unión de clases guía puede asociarse con un conjunto de vectores de código obtenidos por medio del escalado de los vectores de código base de la etapa asociada de vectores de código base con el representante de escala.

Una unión de este tipo de clases guía puede considerarse como un truncamiento. Por lo tanto, si la pluralidad de representaciones de escala son n representaciones de escala, pueden definirse n uniones de clases guía, en donde cada unión de clase guía se define por medio de la respectiva representación de escala y los conjuntos de vectores de código base asociados con la respectiva representación de escala.

Por consiguiente, la pluralidad de representaciones de escala y la pluralidad de conjuntos de vectores de código base pueden definir una pluralidad de unión de clases guía, definiendo de este modo un libro de códigos, en donde, como un ejemplo, cada unión de clases guía puede considerarse como una unión de clases guía escaladas.

Libros de códigos usados dentro de estos códecs de voz y audio pueden, por ejemplo, basarse en estructuras en rejilla, como se describe en la referencia "Multiple-scale leader-lattice VQ with application to LSF quantization" por A. Vasilache, B. Dumitrescu y I. Tabus, Signal Processing, 2002, vol. 82, páginas 563-586, Elsevier. Por ejemplo, puede considerarse para cuantificación una rejilla de D10+, pero también puede considerarse cualquier otra cuantificación en rejilla adecuada.

Por ejemplo, los conjuntos de vectores de código base son clases guía, en donde cada clase guía comprende un vector guía diferente y permutaciones de dicho vector guía, y en el que cada vector guía representa un vector de n dimensiones que comprende n valores absolutos dispuestos en un orden descendente o ascendente.

El vector guía l del respectivo conjunto de vectores de código base b_x puede representarse mediante $l=[l_0, l_1, ..., l_{n-1}]$, en donde $l_0, l_1, ..., l_{n-1}$ son valores absolutos. En caso de un orden descendente, l_0 representa el primer valor más alto, l_1 representa el segundo valor más alto e l_{n-1} representa el n valor más alto. En caso de un orden ascendente, l_0 representa el primer valor más bajo, l_1 representa el segundo valor más bajo e l_{n-1} representa el n valor más bajo.

El valor l_{k-1} del respectivo vector guía, que representa el valor en la $k^{ésima}$ posición en el respectivo vector guía, puede asignarse a una posición en el vector de código base potencial que corresponde a la posición del k valor absoluto más alto (en caso de un vector guía en orden descendente) o a la posición del k valor absoluto más bajo (en caso de un vector guía en orden ascendente) en el vector de entrada. Por ejemplo, esta posición puede indicarse como la posición m. Como un ejemplo, el vector de código base potencial puede representarse mediante $p=[p_0, p_1, ..., p_{n-1}]$.

Por ejemplo, como un ejemplo no limitante, un vector de entrada ilustrativo puede ser

i=[-2,4, 5,0, -1,3, 0,2], en donde el correspondiente vector de entrada con valor absoluto puede ser

ia=[2,4, 5,0, 1,3, 0,2].

En caso del orden descendente del vector guía, el valor en posición k del vector guía, es decir valor l_{k-1}, se asigna a una posición en el vector de código base potencial que corresponde a la posición del k valor absoluto más alto en el vector de entrada. Por ejemplo, comenzando con la primera posición representada por el contador k=1, la posición del primer valor absoluto más alto en el vector de entrada es la posición m=2, ya que el valor 5,0 es el primer valor más alto en el vector de entrada de valor absoluto y se ubica en la posición m=2, es decir ia_1. Por consiguiente, el valor l_0 se asigna a la posición m=2 en el vector de código base potencial, es decir $p_1=l_0$ puede ser válido.

Adicionalmente, el signo (+ o -) del valor asignado en el vector de código base potencial p_{m-1} se establece según el signo del valor del vector de entrada asociado con el k valor absoluto más alto. Por consiguiente,

$$P_{m-1} = l_{k-1} \cdot \text{signo}(l_{m-1})$$

puede ser válido.

Por lo tanto, en el ejemplo no limitante de un vector de entrada ilustrativo i=[-2,4, 5,0, -1,3, 0,2], $p_1=l_0$ puede ser válido ya que el valor i_1=5,0 tiene un signo positivo.

El contador de posición k puede incrementarse, y puede comprobarse si existe otro valor en el vector guía, es decir si $k \leq n$ es válido.

Si es que sí, el método continúa y en el ejemplo no limitante, con respecto a la posición k=2, el valor 2,4 en la posición m=1 representa el segundo (k) valor absoluto más alto en el vector de entrada. Por lo tanto,

$$P_0 = l_1 \cdot \text{signo}(l_o) = -l_1$$

puede ser válido para asignar l_1 con el respectivo signo, ya que el valor i_o=-2,4 en el vector de entrada tiene un signo negativo.

De este modo, para el ejemplo no limitante, el bucle puede iterar a través de las posiciones del vector guía de la siguiente forma:

$$k=3 \rightarrow m=3 \rightarrow P_2 = l_2 \cdot \text{signo}(l_2) = -l_2;$$

y

$$k=4 \rightarrow m=4 \rightarrow P_3 = l_3 \cdot \text{signo}(l_3) = +l_3$$

Por consiguiente, el respectivo vector de código potencial obtenido por el método de ejemplo puede resultar en $p=[-l_1, l_0, -l_2, l_3]$ en caso del respectivo vector guia l ordenado descendentemente.

Si el vector guía l se ordena de una forma ascendente, a continuación el método descrito anteriormente puede realizarse con m representando la posición del k valor más bajo en el vector de entrada de valor absoluto, en donde $p_{m-1} = l_{k-1}\text{signo}(i_{m-1})$ puede ser válido.

El vector de código potencial obtenido p se asocia con el respectivo conjunto de vectores de código base b_x, en donde l representa el vector guía de este respectivo conjunto de vectores de código base. Por ejemplo, con respecto al proceso de ejemplo de determinación de un vector de código basándose en un vector de código base $b_{x,y,t}$ y representante de escala s_z y descritos anteriormente, el vector de código potencial p representa el vector de código base más cercano $b_{x,y}$ del conjunto de vectores de código base b_x con respecto al vector de entrada, en donde el vector de entrada de valor absoluto se usa para determinar el vector de código potencial de un respectivo conjunto de vectores de código base y en el que el signo de cada valor $b_{x,y,k-1}$ en la $k^{\text{ésima}}$ posición del vector de código base más cercano determinado $b_{x,y}$ se asigna con el signo del respectivo valor i_k en la $k^{\text{ésima}}$ posición del vector de entrada i, en donde $0<k\leq n$ es válido.

Por lo tanto, este vector de código base más cercano $b_{x,y}$ que representa el vector de código potencial p puede usarse para determinar un vector de código $c_{x,z,y}$ basándose en el vector de código base más cercano $b_{x,y}$ y basarse en un respectivo representante de escala s_z, como se ha descrito anteriormente.

A cada truncamiento se asigna un representante de escala diferente (por ejemplo, a través de entrenamiento), por ejemplo: escala flotante [] = {0,8, 1,2, 2,7};

Por consiguiente, por ejemplo, un primer conjunto de vectores de código de una pluralidad de vectores de código del libro de códigos se define mediante el primer truncamiento escalado por la primera representación de escala 0,8, un segundo conjunto de vectores de código de la pluralidad de vectores de código del libro de códigos se define mediante el segundo truncamiento escalado por la segunda representación de escala 1,2, y un tercer conjunto de vectores de código de la pluralidad de vectores de código del libro de códigos se define mediante el tercer truncamiento escalado por la tercera representación de escala 2,7, teniendo el libro de códigos una estructura en rejilla de múltiples escalas.

Como un ejemplo, la búsqueda en la estructura en rejilla de múltiples escalas puede verse como que tiene dos fases: la primera puede calcular un vector de código potencial para cada clase guía, es decir para cada conjunto de vectores de código base, y la segunda puede calcular la distorsión únicamente para los vectores de código potenciales.

Por ejemplo, puede aplicarse una función de valor absoluto al vector de entrada i de tal forma que vector de entrada absoluto comprenda los valores absolutos del vector i, y a continuación el vector de entrada absoluto puede clasificarse en un orden descendente (o, como alternativa, ascendente).

Como un ejemplo, una representación de índices puede contener representantes que indican los índices de cada vector de entrada i en el vector de valor absoluto ordenado descendentemente (o ascendentemente). Por ejemplo, dicha representación de índices puede ser una matriz de números enteros 'indx'.

Por ejemplo, si el vector de entrada es [-2,4 5,0 -1,3 0,2], el vector de valor absoluto es [2,4 5,0 1,3 0,2] y la matriz 'indx' es [10 23]. Ya que los vectores guía pueden ordenarse de forma descendente, durante el algoritmo de búsqueda de vecino más cercano, el primer valor del vector guía puede asignarse en la posición que corresponde al componente de valor absoluto más alto del vector de entrada y así sucesivamente.

En el siguiente ejemplo no limitante, 'idx_lead_max' es el número máximo de clases guía de entre todos los truncamientos, que puede corresponder a X, en este ejemplo puede ser 9. Por consiguiente, se definen 9 conjuntos de vectores de código base por medio de las 9 clases guía, en donde la $n^{\text{ésima}}$ clase guía se define mediante &pl[n-1].

Por ejemplo, la matriz 'sign' puede almacenar los signos de los componentes de vector de entrada.

```
/* First part of the search: compute all potential codevectors */
 pl_crt = &pl[0]; /* pl contains the leader vectors */
 for (u=0;u<idx_lead_max;u++)
 {
        for(j=0;j<LATTICE_DIM;j++, pl_crt++)
            j_crt = indx[j];
            if ((*pl_crt) > 0.)
            {
                    cv_pot[u][j_crt] = (*pl_crt)*(float)sign[j_crt]; } else {
                    cv_pot[u][j_crt] = 0.0f; } } }
```

El bucle externo definido por el contador u puede considerarse para asociar cada u con un respectivo vector guía. Por lo tanto, según el contador u, se selecciona un correspondiente conjunto de vectores de código base por medio del bucle externo, ya que cada vector guía corresponde a un conjunto diferente de vectores de código base de la pluralidad de vectores de código base. El bucle interno definido por el valor entero j puede considerarse para determinar un vector de código base potencial asociado con el conjunto seleccionado de vectores de código base, indicando j_crt la posición del (j+1) valor absoluto más alto en el vector de entrada. Por lo tanto, los diferentes vectores de código base potenciales cv_pot se determinan por medio de esta primera parte de la búsqueda ilustrativa.

La segunda parte de la búsqueda puede usarse para determinar un vector de código para codificar el vector de entrada a partir de un subconjunto de vectores de código.

```
/* Second part of the search */
for(I=0;I<no_scales;I++)
{
        s = scale[I];
        s2 = s*s;
        for(k=0;k<LATTICE_DIM;k++) {
        {ws1[k] = w[k]*s*2.0f*in[k];
            ws2[k] = w[k]*s2;
        }
        for(j=0;j<no_leaders[I];j++)
        {
            tmp_dist = 0.0f;
            for(k=0;k<LATTICE_DIM;k++)
            {s = cv_pot[j][k];
                    tmp_dist += (ws2[k]*s-ws1[k])*s;
            }
            if (tmp_dist < min_dist)

               { min_dist = tmp_dist;
               best_scale = I;
               best_idx = k; } } }
```

El bucle externo puede definirse por el contador I, en donde I se emite para seleccionar una representación de escala scale[I] de la pluralidad de representaciones de escala.

LATTICE-DIM define la longitud de los vectores de código que puede corresponder a la longitud del vector de entrada a codificar.

Posteriormente, se calculan los valores ws1[k] y ws2[k] para cada k en (0, ...,LATTICE_DIM), que pueden considerarse que son esa parte de la métrica de distorsión (X3) que es independiente del vector de código base potencial. El valor w[k] representa el valor de la función de ponderación para cada k.

El código de ejemplo mostrado anteriormente adicionalmente tiene un bucle interior j "for(j=0;j<no_leaders[l];j++)", en donde no_leaders[l] define el conjunto de vectores guía asociados con el representante de escala seleccionado scale[l], es decir no_leaders[l] puede corresponder a n_z que representa el número de conjuntos de vectores de código base asociados con el respectivo representante de escala scale[l] y, por lo tanto, este bucle itera a través de cada conjunto de vectores guía asociados con el representante de escala seleccionado scale[l], en donde para el vector guía de este conjunto de vectores guía se ha determinado un código base potencial vectorcv_pot. Por lo tanto, por ejemplo, este bucle selecciona iterativamente cada vector de código base potencial cv_pot el conjunto de vectores de código base asociados con la representación de escala seleccionada, en donde cv_pot[j] puede representar el respectivo $j^{ésimo}$ vector de código base de este conjunto de vectores de código base.

Para cada uno de estos vectores de código base y el representante de escala seleccionado, pueden determinarse la respectiva métrica de distorsión para el vector de código que se asocia con el respectivo vector de código base y el representante de escala seleccionado, por ejemplo, basándose en métrica de distorsión de la siguiente forma:

$$d = \sum_{k=0}^{n-1} \left(ws2[k] \cdot cv\_pot[j][k] - ws1[k] \right) \cdot cv\_pot[j][k]$$

La métrica de distorsión que tiene el valor más bajo se determina para representar la mejor métrica de distorsión, en donde el vector de código asociado con este vector de código de métrica de distorsión puede usarse para codificar el vector de entrada. Por ejemplo, este vector de código puede definirse mediante el mejor representante de escala y el mejor vector de código base potencial del conjunto de vectores de código base potenciales.

Las realizaciones descritas en la presente memoria reducen la complejidad de la cuantificación vectorial no calculando la matriz de vector de códigos potencial cv_pot, sino empleando la versión clasificada de valor absoluto del vector de entrada y determinando o generando el cálculo de distorsión en un espacio transpuesto adecuado.

En algunas realizaciones, el cuantificador de vector en rejilla comprende un clasificador 402 de vector de entrada. El clasificador 402 de vector de entrada o medio adecuado para clasificar el vector de entrada puede configurarse para recibir el vector de entrada.

La operación de recepción del vector de entrada se muestra en la Figura 6 mediante la etapa 501.

El cuantificador de vector en rejilla y clasificador 402 de vector de entrada se configura para clasificar el vector de entrada en un orden descendente de valor absoluto (se entendería que, en algunas realizaciones, la clasificación puede realizarse en un orden ascendente de valor absoluto con los cambios adecuados a las siguientes operaciones).

Por lo tanto, por ejemplo, si el vector de entrada es

$$I = [-2,4\ 5,0\ -1,3\ 0,2],$$

el vector de valor absoluto es

absi = [2,4 5,0 1,3 0,2],

el vector de valor absoluto clasificado que se define en este punto como

cv_pot1 = [5,0 2,4 1,3 0,2]

y la permutación de clasificación 'indx' = [1023].

La clasificación del vector de entrada se muestra en la Figura 6 mediante la etapa 503.

El clasificador de vector de entrada puede pasar a continuación el vector clasificado y permutación de clasificación al determinador 403 de vector de código.

En algunas realizaciones, el Cuantificador 453 de vector en rejilla comprende un determinador de vector de código potencial 403. El determinador de vector de código potencial o medio adecuado para determinar un vector de código potencial se configura para almacenar o generar las clases guía usadas para generar los vectores de código.

Por ejemplo, las clases guía pueden definirse como (en el valor Q1, en otras palabras multiplicado por 2)

```
const Word8 gal_8e[] =
{2, 2, 0, 0, 0, 0, 0, 0,
 1, 1, 1, 1, 1, 1, 1, 1,
 2, 0, 2, 2, 0, 0, 0, 0,
 4, 0, 0, 0, 0, 0, 0, 0,
 3, 1, 1, 1, 1, 1, 1, 1,
 3, 2, 2, 2, 2, 0, 0, 0,
 4, 2, 2, 0, 0, 0, 0, 0,
 3, 3, 1, 1, 1, 1, 1, 1,
 3, 0, 2, 2, 2, 2, 2, 2,
 4, 3, 2, 2, 2, 0, 0, 0,
 4, 4, 0, 0, 0, 0, 0, 0,

 3, 3, 3, 1, 1, 1, 1, 1,
 5, 1, 1, 1, 1, 1, 1, 1,
 4, 2, 2, 2, 2, 2, 0, 0,
 4, 3, 2, 2, 0, 0, 0, 0,
 6, 2, 0, 0, 0, 0, 0, 0,
 3, 3, 3, 2, 1, 1, 1, 1,
 3, 3, 1, 1, 1, 1, 1, 1,
 4, 3, 2, 2, 2, 0, 0, 0,
 4, 3, 4, 2, 0, 0, 0, 0,
 5, 2, 2, 2, 0, 0, 0, 0,
 3, 3, 3, 3, 3, 1, 1, 1,
 5, 3, 3, 3, 1, 1, 1, 1,
 4, 4, 3, 2, 3, 2, 2, 2,
 4, 3, 4, 2, 3, 0, 0, 0,
 5, 2, 2, 2, 3, 2, 0, 0,
 6, 4, 3, 3, 0, 0, 0, 0,
 3, 3, 3, 3, 3, 3, 1, 1,
 5, 3, 3, 3, 1, 1, 1, 1,
 5, 5, 3, 2, 2, 1, 1, 1,
 7, 1, 1, 1, 1, 1, 1, 1,
 4, 4, 4, 2, 2, 2, 2, 0,
 4, 4, 4, 4, 0, 0, 0, 0,
 6, 2, 2, 2, 2, 2, 2, 2,
 6, 4, 2, 2, 2, 0, 0, 0};
```

Estas clases guía, en algunas realizaciones, puede pasarse al determinador 403 de vector de código.

En algunas realizaciones, el Cuantificador 453 de vector en rejilla comprende un determinador 403 de vector de código. El determinador 403 de vector de código o medio adecuado para determinar un vector de código, en algunas realizaciones, puede recibir las clases guía y también el vector de entrada clasificado y vector de permutación. El determinador de vector de código puede determinar a continuación a partir de estos valores el vector de código asociado de salida con el vector de entrada.

Donde la distancia a determinarse es una distancia Euclidiana ponderada, a continuación, en algunas realizaciones, las ponderaciones se transponen según el vector de permutación y se genera un producto de vector de entrada intermedio. Se entendería que, en algunas realizaciones, las ponderaciones son uniformes o la operación de ponderación es opcional donde se emplea la distancia Euclidiana sin ponderar.

Un ejemplo de esto puede mostrarse mediante el siguiente código

```
/* calculate intermediary product between transposed weights and sorted input
vector */
for (j=0;j<LATTICE_DIM;j++)
{
    w_transp[j] = w[indx[j]];
    wx[j] = w_transp[j]*cv_por[j]; }
```

La operación de transposición y aplicación de ponderaciones para generar un producto intermedio basándose en el vector de entrada clasificado y las ponderaciones transpuestas se muestra en la Figura 6 mediante la etapa 505

En algunas realizaciones, el determinador de vector de código puede determinar los componentes de distancia sum1 y sum2 para un primer valor de escala scale[0].

Esta operación puede dividirse en las etapas de:

En primer lugar, inicializar la escala y cuadrado de los valores de escala para un primer valor de escala scale[0].

La operación de inicialización de la escala y cuadrado de los valores de escala se muestran en la Figura 6 mediante la etapa 506.

En segundo lugar, seleccionar un vector guía a partir de la matriz de clases guía. Esto se muestra en el ejemplo de matriz anterior como la matriz pl_crt.

La operación de seleccionar un vector guía se muestra en la Figura 6 mediante la etapa 507.

En tercer lugar, generar valores de distancia intermedios sum1 y sum2 basándose en valores intermedios y el vector guía seleccionado.

La operación de generación de valores de distancia intermedios basándose en el vector guía seleccionado se muestra en la Figura 6 mediante la etapa 509.

En cuarto lugar, comprobar las condiciones de paridad en las que el vector guía no alcanza la 7ª posición y corregir el valor de sum1 en el que el número de signos menos en el vector de entrada difieren de la restricción dada en la paridad de clase guía.

La operación de comprobación de las condiciones de paridad en las que el vector guía no alcanza la 7ª posición y corrección del valor de sum1 donde el número de signos menos en el vector de entrada difieren de la restricción dada en la paridad de clase guía se muestra en la Figura 6 mediante la etapa 511.

En quinto lugar, determinar la distancia o valor de error a partir de los valores sum1 y sum2 y a continuación donde la distancia de vector guía actual es la menor indicar el índice del vector más pequeño.

La operación de determinar la distancia para los vectores guía se muestra en la Figura 6 mediante la etapa 513.

La operación puede entrar en bucle a continuación hasta que todos los vectores guía se han seleccionado.

La operación de comprobación de si todos los vectores guía se han seleccionado y entrado en bucle hacia atrás donde no todos los vectores guía se han seleccionado se muestra en la Figura 6 etapa 514.

Estas etapas pueden mostrarse en el siguiente código

```
for(j=0;j<nc_leadere[0];j++)
{
      suml[j] = 0;
      sum2[j] = 0;
      l = 0;
      while(l<LATTICE_DIM-1)
      {
           p = *pl_crt;
           if (p)
           {
               suml[j] = suml[j] + wx[l] * p;
               sum2[j] = sum2[j] + w_transp[j]*p*p;
               pl_crt++;
               l++;
           }
            else
            {
               pl_crt += LATTICE_DIM-l;
               l = LATTICE_DIM;
            }
       }
       if (l = LATTICE_DIM+ 1 ==0)
       {/* if it went up to 7th position, some leaders
have zeros at the end, so no need for them to check the
parity, because they have null-parity */
     p = *pl_crt;
     if ( pl_par_fx[j] ) /* if non-zero parity */
           {
               if ( sig -pl_par_fx[j] != 0 ) /* if number
of minus signs in the input vector different from the
constraint given by the leader class parity */
          {
          suml[j] = suml[j] - wx[l]* p; /* here is
subtraction */
          sum2[j] = sum2[j] + w_transp[l] *p*p;
          pl_crt++;
               }
          else
```

```
            {
             sum1[j] = sum1[j] + wx[l] * p;
             sum2[j] = sum2[j] + w_transp[l]*p*p;
             pl_cnt++;
             }
        }
        else
        {
             sum1[j] = sum1[j] + wx[l]* p;
             sum2[j] = sum2[j] + w_transp[l] *p*p;
             pl_cnt++;
              }
          }
          tmp_dist = sum2[j]*s2 -sum1[j]*s;
          if (tmp_dist < min_dist )
          {
              min_dist = tmp_dist;
              best_idx = j;
          }
    }  /* end of } loop */
```

A continuación, en algunas realizaciones, el determinador de vector de código puede configurarse para usar los valores sum1 y sum2 para determinar distancias de distorsión para otras escalas. Se hace adicionalmente una operación similar de comprobación de un 'mejor' valor de escala.

La operación de determinación de distancias de distorsión para otras escalas se muestra en la Figura 6 mediante la etapa 515.

La operación de determinar la distorsión distancia puede para las otras escalas usando los valores sum1 y sum2 puede implementarse usando el siguiente código de ejemplo

```
for (k=1; k<no_scales; k++)
{s = scale [k];
    s2 = s*s;
    /* and now use the sum1, sum2 values calculated above
    to calculate distortion for the other scales */
    for (j=0; j<no_leaders [j] ; j++)
    {
          tmp_dist = sum2[j]*s2 -sum1[j]*s;
          if (tmp_dist < min_dist)
               {min_dist = tmp_dist;
               best_scale = k;
               best_idx = j ;
                    }
    }
}
```

Además, en algunas realizaciones, puede configurarse el determinador de vector de código, una vez que se encuentran la mejor clase guía y mejor escala, para calcular el vector de código resultante 'cv_out'.

La operación de realizar una transposición inversa para calcular el vector de código se muestra en la Figura 6 mediante la etapa 517.

En algunas realizaciones, la operación de cálculo del vector de código puede implementarse mediante el siguiente código de ejemplo.

```
/* Inverse permutation */
for(j=0; j<LATTICE_DIM;j++)
 {
       id[indx[j]] = j;
 }


for (j=0 ; j<LATTICE_DIM; j++)
 {
    cv_out [j] = sign [j]
    *pl_fx[best_idx*LATTICE_DIM+id[j]]) ;}
if (pl_par_fx[best_idx])
 {
    if (sig -pl_par_fx[best_idx] ! = 0)
     {
          cv_out[smallest] = -cv_out[smallest]; } }
```

En algunas realizaciones, el cálculo de las variables sum1 y sum2 se hace hasta el número de guías del primer truncamiento (no_leaders[0]), significando que el número de guías debería ordenarse de forma decreciente y sus correspondientes escalas ordenadas en consecuencia.

En tales realizaciones, se produce una reducción de complejidad adicional porque el número máximo de guías para una estructura no necesita calcularse, sino que se sabe que está en la primera posición.

Se entendería que la mayor parte de la reducción viene del hecho de que únicamente el vector guía vencedor tiene que transponerse, no todos. El cálculo se hace sobre valores positivos (tanto vector guía como vector de entrada están en valores absolutos) que es válido siempre que el componente de vector de entrada y el cuantificado tengan el mismo signo.

Una diferencia en signo interviene cuando existe una restricción de paridad (número impar o par de componentes negativos) en el vector guía considerado y el vector de entrada no respeta esta restricción. En este caso el signo de valor cuantificado de los componentes de vector de entrada más pequeño tiene su signo invertido. El componente de vector de entrada más pequeño corresponde al último componente en el espacio transpuesto. Esto es por lo que el primer bucle para calcular sum1 y sum2 es “while(I<LATTICE-DIM-1)”. En realidad, espacio no transpuesto esto corresponde a smallest = indx[LATTICE_DIM-1]. LATTICE_DIM es la dimensión de la rejilla considerada.

Aunque los ejemplos anteriores describen realizaciones de la aplicación que operan dentro de un códec dentro de un aparato 10, se apreciaría que la invención como se describe a continuación puede implementarse como parte de cualquier códec de audio (o voz), incluyendo cualquier códec de audio (o voz) de tasa variable/tasa adaptativa. Por lo tanto, por ejemplo, realizaciones de la aplicación pueden implementarse en un códec de audio que puede implementar codificación de audio a través de trayectorias de comunicación fijas o por cable.

Por lo tanto, equipo de usuario puede comprender un códec de audio tal como los descritos en las realizaciones de la aplicación anterior.

Deberá apreciarse que la expresión equipo de usuario se pretende que cubra cualquier tipo adecuado de equipo de usuario inalámbrico, tal como teléfonos móviles, dispositivos de procesamiento de datos portátiles o exploradores de web portátiles.

Además, elementos de una red móvil pública terrestre (PLMN) también pueden comprender códecs de audio como se ha descrito anteriormente.

En general, las diversas realizaciones de la aplicación pueden implementarse en hardware o circuitos de fin especial, software, lógica o cualquier combinación de los mismos. Por ejemplo, algunos aspectos pueden implementarse en hardware, mientras que otros aspectos pueden implementarse en firmware o software que puede ejecutarse por un controlador, microprocesador u otro dispositivo informático, aunque la invención no está limitada a lo mismo. Mientras diversos aspectos de la aplicación pueden ilustrarse y describirse como diagramas de bloque, diagramas de flujo, o usando alguna otra representación pictórica, se entiende bien que estos bloques, aparato, sistemas, técnicas o métodos descritos en la presente memoria pueden implementare, como ejemplos no limitantes, en hardware, software, firmware, circuitos o lógica de fin especial, hardware de fin general o controlador u otros dispositivos informáticos o alguna combinación de los mismos.

Las realizaciones de esta solicitud pueden implementarse mediante software informático ejecutable por un procesador de datos del dispositivo móvil, tal como en la entidad de procesador, o mediante hardware o mediante una combinación de software y hardware. Además en este sentido debería observarse que cualesquiera bloques del flujo lógico como en las figuras pueden representar etapas de programa, o circuitos, bloques y funciones de lógica interconectados, o una combinación de etapas de programa y circuitos, bloques y funciones de lógica.

La memoria puede ser de cualquier tipo adecuado al entorno técnico local y puede implementarse usando cualquier tecnología de almacenamiento de datos adecuada, tal como dispositivos de memoria basada en semiconductores, dispositivos y sistemas de memoria magnética, dispositivos y sistemas de memoria óptica, memoria fija y memoria extraíble. Los procesadores de datos pueden ser de cualquier tipo adecuado al entorno técnico local, y pueden incluir uno o más de ordenadores de fin general, ordenadores de fin especial, microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), circuitos de nivel de puertas y procesadores basados en arquitectura de procesador de múltiples núcleos, como ejemplos no limitantes.

Realizaciones de la aplicación pueden practicarse en diversos componentes tales como módulos de circuito integrado. El diseño de circuitos integrados es en términos generales un proceso altamente automatizado. Están disponibles herramientas de software complejas y potentes para convertir un diseño de nivel lógico en un diseño de circuito de semiconductores listo para grabarse y formarse en un sustrato de semiconductores.

Programas, tales como aquellos proporcionados por Sypnopsys, Inc. de Mountain View, California y Cadence Design, de San José, California encaminan automáticamente conductores y localizan componentes en un chip de semiconductores usando reglas bien establecidas de diseño así como bibliotecas de módulos de diseño prealmacenados. Una vez que el diseño para un circuito de semiconductores se ha completado, el diseño resultante, en un formato electrónico normalizado (por ejemplo, Opus, GDSII o similares) puede transmitirse a una instalación de fabricación de semiconductores o “fab” para fabricación.

Como se usa en esta solicitud, el término 'circuitería' hace referencia a todo lo siguiente:

(a) implementaciones de circuito únicamente de hardware (tal como implementaciones en únicamente circuitería analógica y/o digital) y

(b) a combinaciones de circuitos y software (y/o firmware), tales como: (i) a una combinación de procesador o procesadores o (ii) a porciones de procesador o procesadores/software (incluyendo procesador o procesadores de señales digitales), software, y memoria o memorias que trabajan juntas para provocar que un aparato, tal como un teléfono móvil o servidor, realice diversas funciones y

(c) a circuitos, tal como un microprocesador o microprocesadores o una porción de un microprocesador o microprocesadores, que requieren software o firmware para operación, incluso si el software o firmware no está físicamente presente.

Esta definición de 'circuitería' se aplica a todos los usos de este término en esta solicitud, incluyendo cualquier reivindicación. Como un ejemplo adicional, como se usa en esta solicitud, el término 'circuitería' también cubriría una implementación de meramente un procesador (o múltiples procesadores) o porción de un procesador y su (o sus) software y/o firmware adjunto. El término 'circuitería' también cubriría, por ejemplo, y si es aplicable al elemento de reivindicación particular, un circuito integrado de banda base o circuito integrado de procesador de aplicaciones para un teléfono móvil o circuito integrado similar en servidor, un dispositivo de red celular u otro dispositivo de red.

La descripción anterior ha proporcionado por medio de ejemplos ejemplares y no limitantes una descripción completa e informativa de la realización ejemplar de esta invención. Sin embargo, pueden hacerse evidentes diversas modificaciones y adaptaciones para los expertos en la técnica en vista de la descripción anterior, cuando se leen en conjunto con los dibujos adjuntos y las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método implementado por procesador para codificar al menos una señal de audio, en donde el método comprende:

generar al menos un vector de parámetros que definen la al menos una señal de audio;
clasificar componentes de valor absoluto del al menos un vector de parámetros según un orden descendente de los valores absolutos de los componentes del al menos un vector de parámetros para generar un al menos un vector ordenado asociado de parámetros;
seleccionar de una lista de clases guía al menos un vector de código potencial;
realizar, para cada uno del al menos un vector de código potencial seleccionado individualmente y para cada uno del al menos un vector de parámetros ordenados individualmente, una etapa de determinación de una distancia entre el vector de código potencial único y el vector ordenado único de parámetros, en donde la etapa de determinación comprende:

(a) generar un primer y un segundo valor de distancia intermedia, respectivamente, en donde el primer valor de distancia intermedia se obtiene mediante la suma de los productos de los correspondientes componentes del vector de código potencial único y el vector ordenado único de parámetros y el segundo valor de distancia intermedia se obtiene mediante la suma de los cuadrados de los componentes del vector de código potencial único;
(b1) actualizar el primer valor de distancia intermedia restando el producto de un último componente del vector de código potencial único y un último componente del vector ordenado único de parámetros del primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de condiciones de cuando el vector de código potencial único es de paridad distinta de cero y cuando el número de signos menos de los componentes del vector único de parámetros difiere de la restricción de la paridad de clase guía asociada con el vector de código potencial único;
(b2) actualizar el primer valor de distancia intermedia añadiendo el producto de un último componente del vector de código potencial único y un último componente del vector ordenado único de parámetros al primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de condiciones de cuando el vector de código potencial único es de paridad distinta de cero y cuando el número de signos menos de los componentes del vector único de parámetros no difiere de la restricción de la paridad de clase guía asociada con el vector de código potencial único;
(b3) actualizar el primer valor de distancia intermedia añadiendo el producto del último componente del vector de código potencial único y el último componente del vector ordenado único de parámetros al primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de una condición de cuando el vector de código potencial único no es de paridad distinta de cero;
(c) determinar la distancia entre el vector de código potencial único y el vector ordenado único de parámetros restando el primer valor de distancia intermedia multiplicado por un factor de escala del segundo valor de distancia intermedia multiplicado por el factor de escala al cuadrado;

determinar la mejor clase guía asociada con el vector de código potencial único que genera la menor distancia asociada; y
clasificar componentes de la mejor clase guía por el orden inverso del orden descendente de valores absolutos de los componentes del vector único de parámetros para generar un vector cuantificado en rejilla de salida.

2. El método según la reivindicación 1, comprendiendo adicionalmente:

seleccionar el factor de escala de una pluralidad de factores de escala; y
aplicar el factor de escala al vector de código cuantificado en rejilla de salida.

3. El método según cualquiera de las reivindicaciones 1 y 2, en donde generar un primer vector de parámetros que definen al menos una señal de audio comprende:

dividir la al menos una señal de audio en tramas de tiempo; y

determinar un vector de parámetros de frecuencia espectral de línea asociados con al menos una de las tramas de tiempo de señal de audio.

4. El método según cualquiera de las reivindicaciones 1 a 3, en donde clasificar componentes de valor absoluto del al menos un vector de parámetros comprende además:

determinar ponderaciones para una determinación de distancia ponderada;
clasificar las ponderaciones basándose en el orden descendente basándose en los valores absolutos de los componentes del al menos un vector de parámetros para generar un vector de ponderación clasificado; y
aplicar el vector de ponderación clasificado al al menos un vector ordenado de parámetros.

5. Un aparato que comprende hardware de procesamiento para implementar la codificación de al menos una señal de audio, en donde el hardware de procesamiento se configura para:

generar un vector de parámetros que definen la al menos una señal de audio;
clasificar componentes de valor absoluto del vector de parámetros según un orden descendente de los valores absolutos de los componentes del vector de parámetros para generar un vector ordenado asociado de parámetros;
seleccionar de una lista de clases guía vectores de códigos potenciales hasta el número de clases guía de un primer truncamiento, en donde en la lista de clases guía cada clase guía se define como un vector guía correspondiente, y en donde el primer truncamiento es el truncamiento con el número máximo de clases guía;
realizar, para cada uno de los vectores de códigos potenciales seleccionados individualmente y para el vector de parámetros ordenados, una etapa de determinación de una distancia entre el vector de código potencial y el vector ordenado de parámetros, en donde la etapa de determinación se realiza por el hardware de procesamiento configurándose para:

(a) generar un primer y un segundo valor de distancia intermedia, respectivamente, en donde el primer valor de distancia intermedia se obtiene mediante la suma de los productos de los correspondientes componentes del vector de código potencial único y el vector ordenado de parámetros y el segundo valor de distancia intermedia se obtiene mediante la suma de los cuadrados de los componentes del vector de código potencial único;
(b1) actualizar el primer valor de distancia intermedia restando el producto de un último componente del vector de código potencial único y un último componente del vector ordenado de parámetros del primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de condiciones de cuando el vector de código potencial único es de paridad distinta de cero y cuando el número de signos menos de los componentes del vector de parámetros difiere de la restricción de la paridad de clase guía asociada con el vector de código potencial único;
(b2) actualizar el primer valor de distancia intermedia añadiendo el producto de un último componente del vector de código potencial único y un último componente del vector ordenado de parámetros al primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de condiciones de cuando el vector de código potencial único es de paridad distinta de cero y cuando el número de signos menos de los componentes del vector de parámetros no difiere de la restricción de la paridad de clase guía asociada con el vector de código potencial único;
(b3) actualizar el primer valor de distancia intermedia añadiendo el producto del último componente del vector de código potencial único y el último componente del vector ordenado de parámetros al primer valor de distancia intermedia y actualizar el segundo valor de distancia intermedia añadiendo el cuadrado del último componente del vector de código potencial único al segundo valor de distancia intermedia dependiendo de una condición de cuando el vector de código potencial único no es de paridad distinta de cero;
(c) determinar la distancia entre el vector de código potencial y el vector ordenado único de parámetros restando el primer valor de distancia intermedia multiplicado por un factor de escala del segundo valor de distancia intermedia multiplicado por el primer factor de escala al cuadrado;

utilizar los respectivos valores de distancia intermedios primero y segundo, calculados para determinar las distancias entre los respectivos vectores de códigos potenciales únicos y el vector ordenado de parámetros

para el primer factor de escala, para determinar las distancias entre los respectivos vectores de códigos potenciales únicos y el vector ordenado de parámetros para otros factores de escala;
determinar el mejor factor de escala y la mejor clase guía asociados con el único vector de código potencial que genere la menor distancia asociada; y
clasificar los componentes de la mejor clase guía, definida como su correspondiente vector guía en la lista de clases guías, por el orden inverso del orden descendente de los valores absolutos de los componentes del vector de parámetros para generar un vector de código cuantificado en rejilla de salida.

6. El aparato según la reivindicación 5, en donde el hardware de procesamiento se configura adicionalmente para:

seleccionar el primer factor de escala de una pluralidad de factores de escala; en donde se asignan diferentes factores de escala de la pluralidad de factores de escala a los truncamientos correspondientes, en donde los truncamientos se ordenan de forma decreciente según su número de clases principales, de tal modo que el primer truncamiento es el truncamiento con el número máximo de clases principales, y en donde sus factores de escala correspondientes se ordenan en consecuencia, de modo que el primer factor de escala es el factor de escala correspondiente al primer truncamiento ; y
aplicar el mejor factor de escala al vector de código cuantificado en rejilla de salida.

7. El aparato según cualquiera de las reivindicaciones 5 y 6, en donde el hardware de procesamiento configurado para generar el vector de parámetros que definen al menos una señal de audio se configura adicionalmente para:

dividir la al menos una señal de audio en tramas de tiempo; y
determinar un vector de parámetros de frecuencia espectral de línea asociados con al menos una de las tramas de tiempo de señal de audio.

8. El aparato según cualquiera de las reivindicaciones 5 a 7, en donde el hardware de procesamiento configurado para clasificar componentes de valor absoluto del vector de parámetros se configura adicionalmente para:

determinar ponderaciones para una determinación de distancia ponderada;
clasificar las ponderaciones basándose en el orden descendente basándose en los valores absolutos de los componentes del vector de parámetros para generar un vector de ponderación clasificado; y
aplicar el vector de ponderación clasificado al vector ordenado de parámetros.

Figura 1

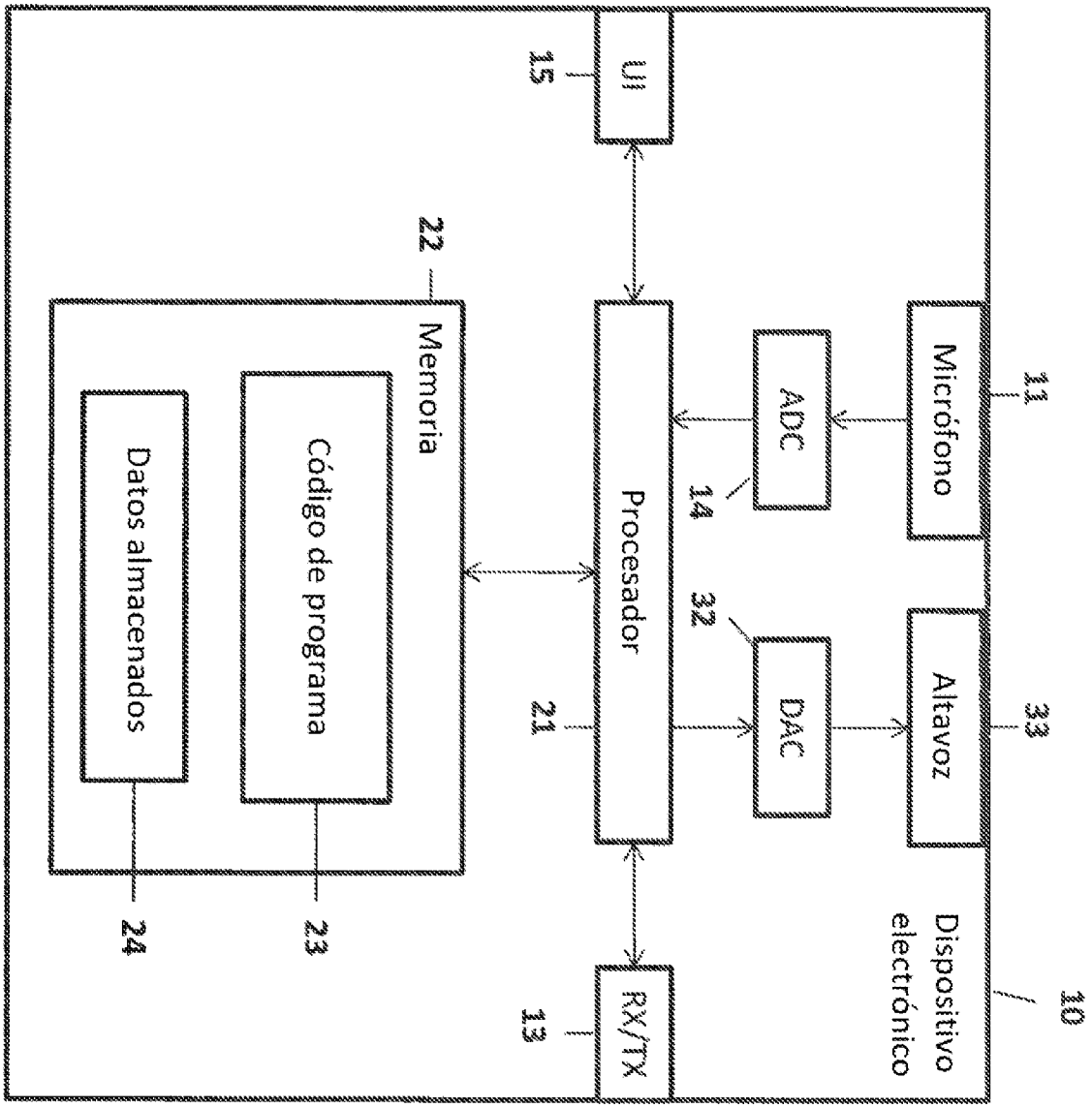

Dispositivo electrónico
10
Micrófono
11
Altavoz
33
ADC
14
DAC
32
Procesador
21
UI
15
RX/TX
13
Memoria
22
Código de programa
23
Datos almacenados
24

Figura 2

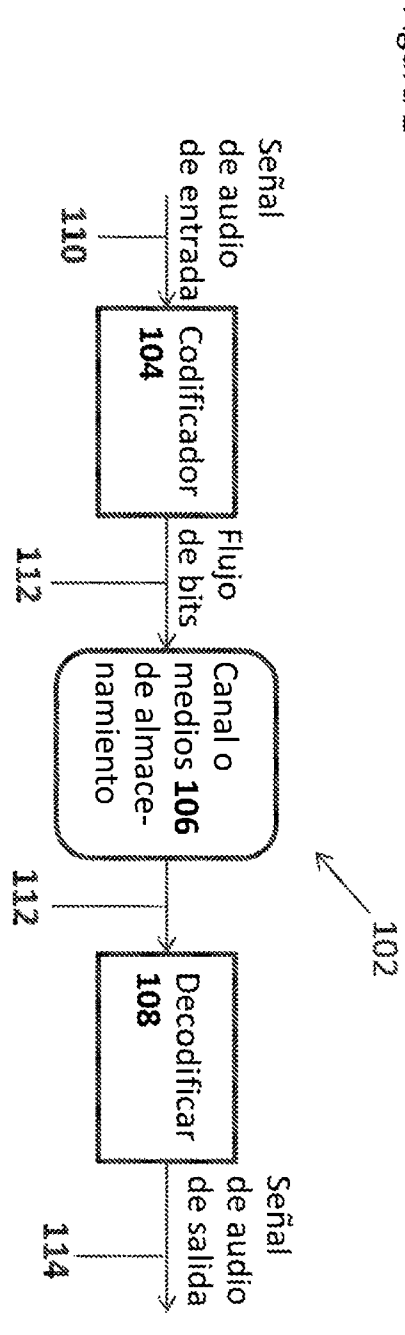


Figura 3

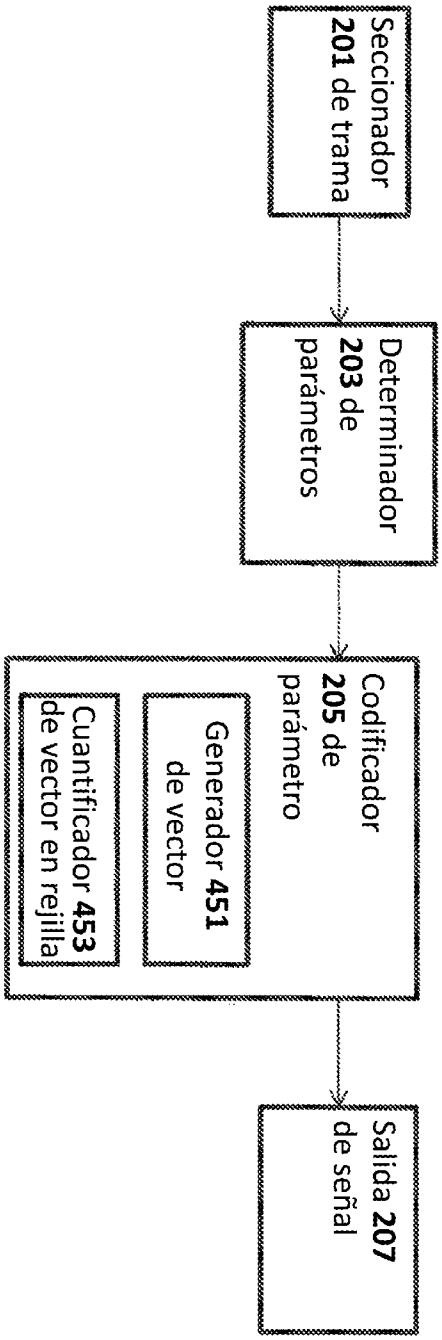


Figura 4

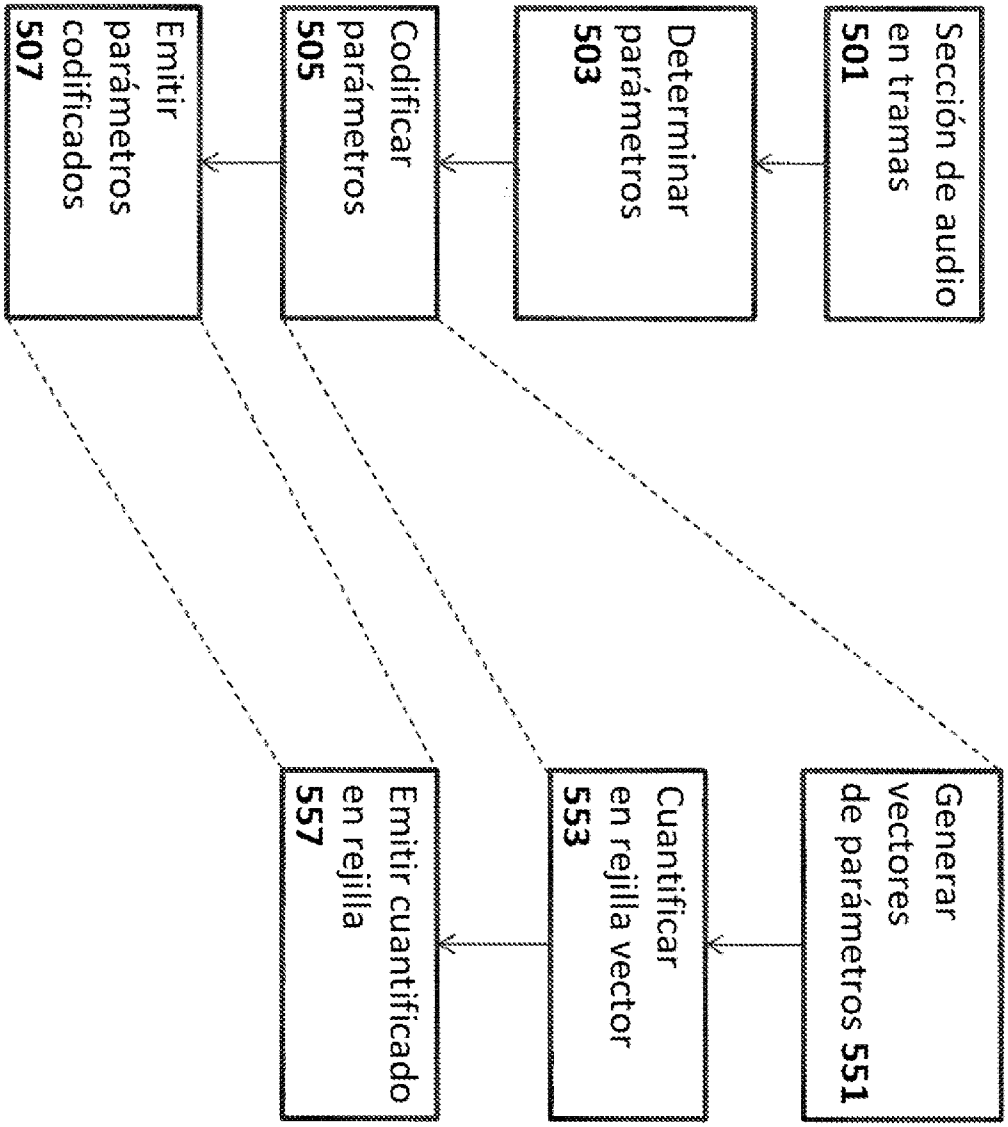

Sección de audio en tramas 501
Determinar parámetros 503
Codificar parámetros 505
Emitir parámetros codificados 507
Generar vectores de parámetros 551
Cuantificar en rejilla vector 553
Emitir cuantificado en rejilla 557

Figura 5

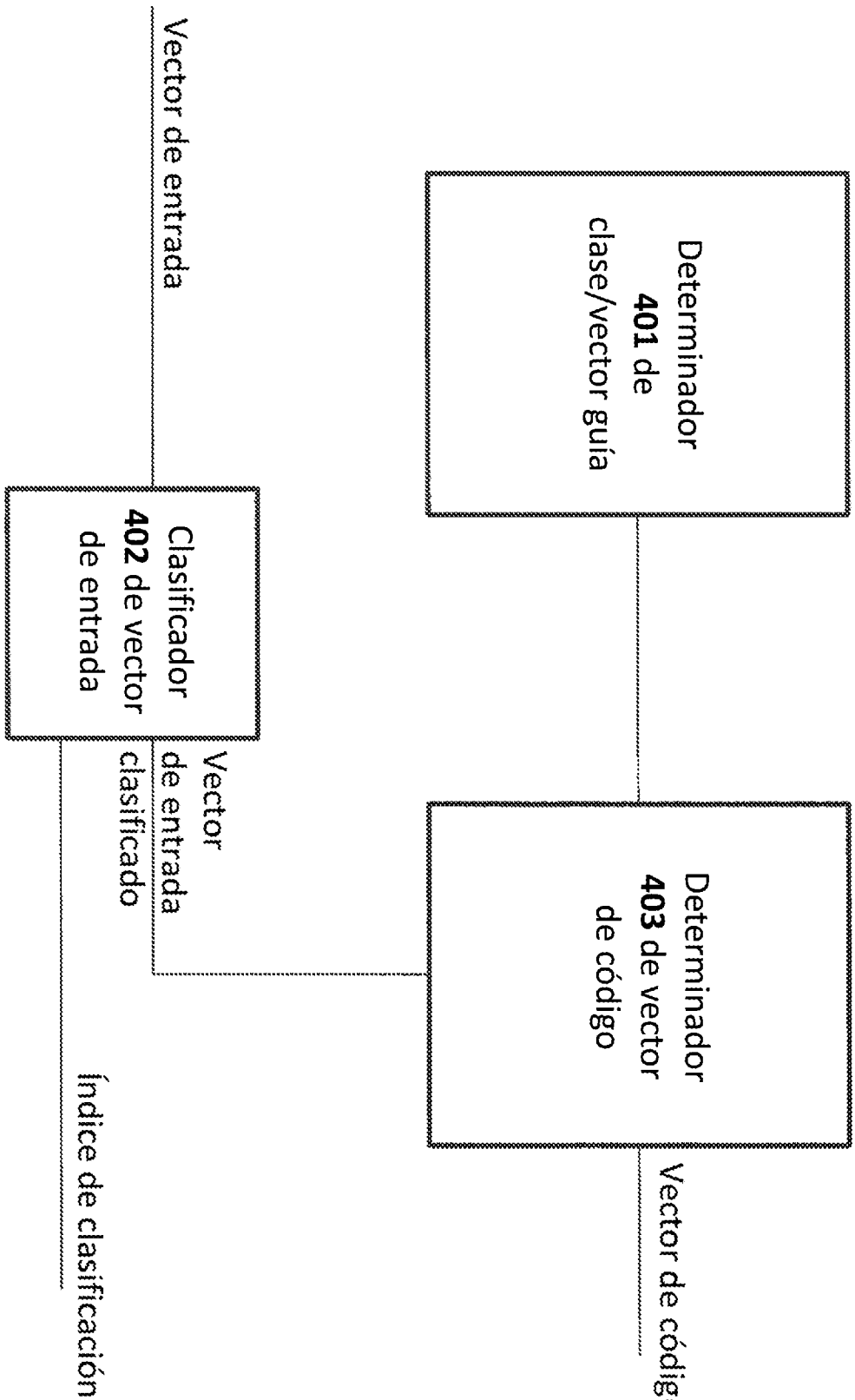
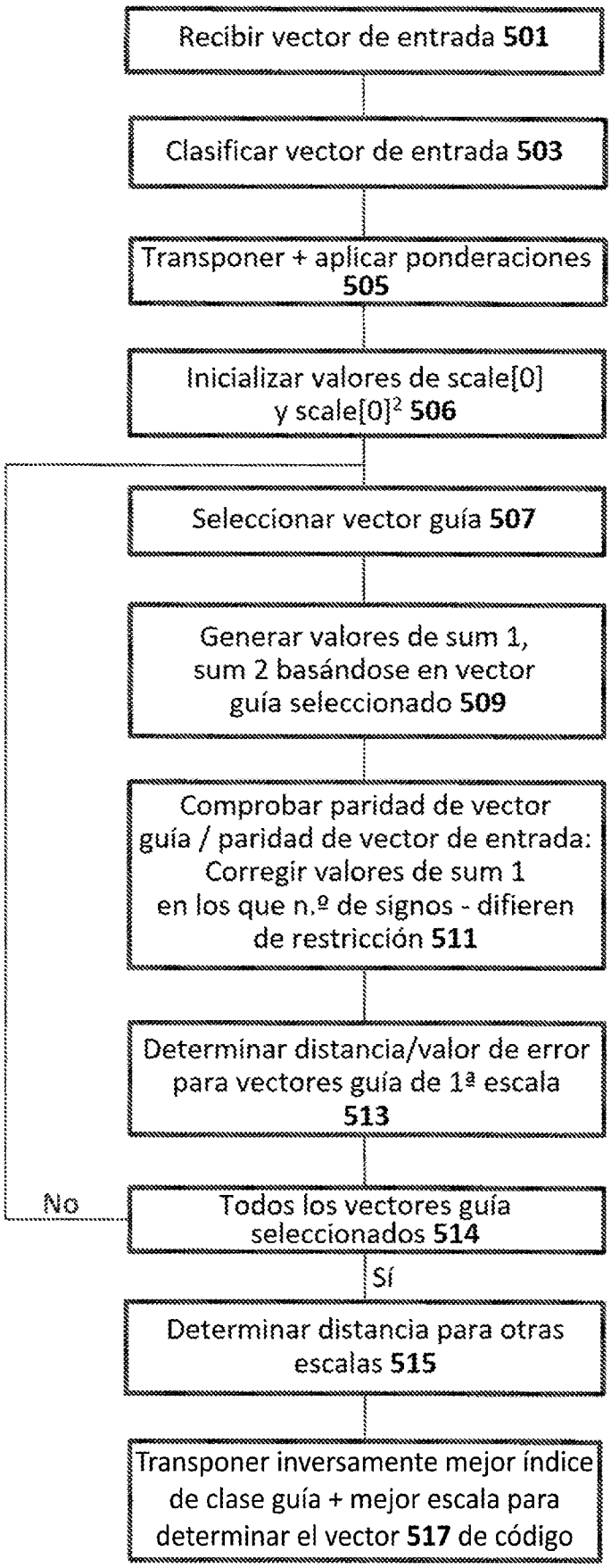

Determinador
401 de
clase/vector guía
Vector de entrada
Clasificador
402 de vector
de entrada
Vector
de entrada
clasificado
Determinador
403 de vector
de código
Vector de código
Índice de clasificación

Figura 6


Recibir vector de entrada 501
Clasificar vector de entrada 503
Transponer + aplicar ponderaciones 505
Inicializar valores de scale[0] y scale[0]² 506
Seleccionar vector guía 507
Generar valores de sum 1, sum 2 basándose en vector guía seleccionado 509
Comprobar paridad de vector guía / paridad de vector de entrada: Corregir valores de sum 1 en los que n.º de signos - difieren de restricción 511
Determinar distancia/valor de error para vectores guía de 1ª escala 513
No
Todos los vectores guía seleccionados 514
Sí
Determinar distancia para otras escalas 515
Transponer inversamente mejor índice de clase guía + mejor escala para determinar el vector 517 de código