

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 627 212**

51 Int. Cl.:

**G10L 19/24** (2013.01)

**G10L 19/02** (2013.01)

**G10L 19/038** (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **14.07.2006 PCT/US2006/027238**

87 Fecha y número de publicación internacional: **25.01.2007 WO07011657**

96 Fecha de presentación y número de la solicitud europea: **14.07.2006 E 06787180 (6)**

97 Fecha y número de publicación de la concesión europea: **01.03.2017 EP 1905011**

54 Título: **Modificación de palabras de código en diccionario usado para codificación eficaz de datos espectrales de medio digital**

30 Prioridad:

**15.07.2005 US 183084**

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

**27.07.2017**

73 Titular/es:

**Microsoft Technology Licensing, LLC (100.0%)  
ONE MICROSOFT WAY  
REDMOND, WA 98052-6399, US**

72 Inventor/es:

**CHEN, WEI-GE;  
MEHROTRA, SANJEEV y  
KOISHIDA, KAZUHITO**

74 Agente/Representante:

**CARVAJAL Y URQUIJO, Isabel**

ES 2 627 212 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**

Modificación de palabras de código en diccionario usado para codificación eficaz de datos espectrales de medio digital.

Campo técnico

5 La tecnología se refiere en general a codificación de datos espectrales representando ciertas porciones de los datos espectrales según versiones modificadas de otras porciones previamente codificadas.

Antecedentes

10 La codificación de audio utiliza técnicas de codificación que aprovechan diversos modelos perceptuales del oído humano. Por ejemplo, muchos tonos más débiles cerca de unos fuertes se enmascaran por lo que no necesitan codificarse. En codificación de audio perceptual tradicional, esto se aprovecha como cuantificación adaptativa de datos de frecuencia diferente. Los datos de frecuencia perceptualmente importantes se les asignan más bits y por lo tanto cuantificación más precisa y viceversa.

15 La codificación perceptual, sin embargo, puede llevarse a un sentido más amplio. Por ejemplo, algunas partes del espectro pueden codificarse con ruido apropiadamente conformado. Cuando se toma este enfoque, la señal codificada puede no tener como objetivo representar una versión exacta o casi exacta de la original. En su lugar la meta es hacer que suene similar y agradable cuando se compara con el original.

20 Todos estos efectos perceptuales pueden usarse para reducir la tasa de bits necesaria para codificación de señales de audio. Esto es debido a que algunos componentes de frecuencia no necesitan representarse de manera precisa como presentes en la señal original, sino que pueden no codificarse o sustituirse con algo que proporcione el mismo efecto perceptual que en el original.

25 En el artículo por Kwon, S.Y. et al.: "A high quality BI-CELP speech coder at 8 kbit/s and below", IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97, Múnich, Alemania 21-24 de abril de 1997, Los Alamitos, CA, Estados Unidos, IEEE Comput. Soc; Estados Unidos, Estados Unidos, vol. 2, 21 de abril de 1997 (21-04-1997), páginas 759-762, DOI: 10.1109/ICASSP.1997.596034, ISBN: 978-0-8186-7919-3, se desvela un método de codificación de voz, que está basado en un modelo de Predicción Lineal con Excitación por Código (CELP) que emplea vectores de excitación combinados desde dos libros de código. Uno de los vectores de excitación proviene del libro de códigos de línea de base y el otro proviene del libro de códigos implicado. En este método, el índice del libro de códigos de línea de base se codifica y transmite al receptor mientras que el índice del libro de códigos implicado se extrae desde la voz sintetizada. Además, en el documento US 6.073.092 A, se desvela un método para codificación de voz usando predicción lineal con excitación por código, que produce voz de calidad de larga distancia a tasas de datos entre 4 y 16 kbit/s. Se usa una serie de libros de códigos de línea de base, implicados y adaptativos, comprendidos de libros de código de pulso y aleatorios, con vectores de juego asociados para caracterizar la voz. Adicionalmente, en el artículo por Epps J. et al: "A new technique for wideband enhancement of coded narrowband speech", Speech Coding Processings, 1999 IEEE Workshop en Porvoo, Finlandia 20-23 de junio de 1999, Piscataway, NJ, Estados Unidos, IEEE, US, 20 de junio de 1999 (20-06-1999), páginas 174-176, DOI: 10.1109/SCFT.1999.781522, ISBN: 978-0-7803-5651-1, se desvela una técnica para predicción de envolvente espectral de banda alta basándose en mapeo de libro de códigos con libros de códigos divididos por sonorización.

Sumario

40 Una técnica de codificación/decodificación de audio descrita en el presente documento utiliza el hecho de que algunos componentes de frecuencia pueden representarse perceptualmente bien, o parcialmente, usando ruido conformado, o versiones conformadas de otros componentes de frecuencia, o la combinación de ambos. Más particularmente, algunas bandas de frecuencia pueden representarse perceptualmente bien como una versión conformada de otras bandas que ya se han codificado. Incluso aunque el espectro real pueda desviarse de su versión sintética, aún es una representación perceptualmente buena que puede usarse para reducir significativamente la tasa de bits de la codificación de señal de audio sin reducir la calidad.

50 Se describen diversas características opcionales para modificar los vectores de código (por ejemplo, palabras de código) en el libro de códigos de acuerdo con algunas reglas que permiten que el vector de código represente mejor los datos de sub-banda. La modificación puede consistir en cualquiera de una transformación lineal o no lineal, o representando el vector de código como una combinación de los otros dos vectores de código. En el caso de una combinación, la modificación puede proporcionarse tomando porciones de un vector de código y combinándolo con porciones de otros vectores de código.

Una palabra de código proviene de una banda base, un libro de códigos fijado y/o una palabra de código generada aleatoriamente. Adicionalmente, una palabra de código puede provenir también de una banda que se codificó previamente por cualquiera de un codificador de banda base o codificador de banda extendida. Las referencias a palabras de código en el presente documento, incluyen todas estas fuentes potenciales para palabras de código, aunque cualquier realización particular puede usar únicamente un subconjunto de estas fuentes para palabras de código. Se realizan diversas transformaciones lineales o no lineales en una o más palabras de código en una biblioteca para obtener un conjunto mayor o más diverso de formas para identificar una mejor forma para hacer coincidir un vector que se esté codificando. En un ejemplo, una palabra de código se invierte en orden de coeficiente para obtener otra palabra de código para coincidencia de forma. En otro ejemplo, una varianza de la palabra de código se reduce usando exponenciación de coeficientes con un exponente menor que uno. De manera similar, una varianza de la palabra de código se exagera usando un exponente mayor que uno. En otro ejemplo, los coeficientes de una palabra de código se niegan. Por supuesto, pueden realizarse muchas otras transformaciones lineales y no lineales en una o más palabras de código para proporcionar un universo más grande o más diverso para sub-bandas coincidentes u otros vectores.

En otro ejemplo, se realiza una búsqueda exhaustiva junto con una banda base y/u otros libros de código para hallar una palabra de código mejor coincidente. Por ejemplo, se realiza una búsqueda que comprende una búsqueda exhaustiva de una biblioteca de palabra de código, que incluye todas las combinaciones de transformación exponencial ( $p=0,5, 1,0, 2,0$ ), transformación de signo (+/-) y transformación de dirección (hacia delante/inversa). De manera similar, esta búsqueda exhaustiva puede realizar a lo largo del espectro de libro de códigos de ruido, otros libros de código o vectores de ruido aleatorio.

En general, puede proporcionarse una coincidencia estrecha determinando una varianza más baja entre la subbanda que se esté codificando y una palabra de código transformada. Un identificador de la palabra de código y transformación, junto con otra información tal como un factor de escala, se codifican en el flujo de bits y se proporcionan al decodificador.

En otro ejemplo, dos o más palabras de código se combinan para proporcionar un modelo para codificar. Por ejemplo, se proporcionan dos palabras de código  $b$  y  $n$ ,  $b = \langle b_0, b_1 \dots b_u \rangle$  y  $n = \langle n_0, n_1 \dots n_u \rangle$  para describir mejor una subbanda que se está codificando. El vector  $b$  puede provenir desde la banda base, un libro de códigos de ruido o una biblioteca, y el vector  $n$  puede provenir de manera similar desde cualquier fuente de este tipo. Se proporciona una regla para intercalar coeficientes desde cada dos o más palabras de código  $b$  y  $n$ , de manera que el decodificador conoce implícita o explícitamente qué coeficiente tomar desde las palabras de código  $b$  y  $n$ . La regla puede proporcionarse en el flujo de bits o puede conocerse por el decodificador implícitamente. Como alternativa, "b" puede ser la codificación real que usa la codificación de forma de onda en lugar de una palabra de código.

Por lo tanto, un codificador puede enviar dos o más identificadores de palabra de código, y opcionalmente, una regla para decodificar qué coeficientes tomar para crear la subbanda. El codificador enviará también información de factor de escala para palabras de código, y opcionalmente, si fuera relevante, cualquier otra información de transformación de palabra de código.

Se harán evidentes características y ventajas adicionales de la invención a partir de la siguiente descripción detallada de las realizaciones que continúa con referencia a los dibujos adjuntos.

Breve descripción de los dibujos

Las Figuras 1 y 2 son un diagrama de bloques de un codificador y decodificador de audio en el que pueden incorporarse las presentes técnicas de codificación.

La Figura 3 es un diagrama de bloques de un codificador de banda base y codificador de banda extendida que implementan la codificación de audio eficaz usando palabras de código modificadas y o segmentación de frecuencia variable que pueden incorporarse en el codificador de audio general de la Figura 1.

La Figura 4 es un diagrama de flujo de bandas de codificación con la codificación de audio eficaz que usa el codificador de banda extendida de la Figura 3.

La Figura 5 es un diagrama de bloques de un decodificador de banda base, un decodificador de configuración de banda extendida y decodificador de banda extendida que pueden incorporarse en el decodificador de audio general de la Figura 2.

La Figura 6 es un diagrama de flujo de bandas de decodificación con la codificación de audio eficaz que usa el decodificador de banda extendida de la Figura 5.

La Figura 7 es un gráfico que representa un conjunto de coeficientes espectrales.

La Figura 8 es un gráfico de una palabra de código y diversas transformaciones lineales y no lineales de la palabra de código.

La Figura 9 es un gráfico de un vector ejemplar que no representa picos de manera distinta.

5 La Figura 10 es un gráfico de la Figura 9 con distintos picos creados mediante la modificación de palabra de código mediante transformación exponencial.

La Figura 11 es un gráfico de una palabra de código según se compara a la subbanda que está modelando.

La Figura 12 es un gráfico de una palabra de código de subbanda transformada según se compara con la subbanda que está modelando.

10 La Figura 13 es un gráfico de una palabra de código, una subbanda a codificarse mediante la palabra de código, una versión cambiada de escala de la palabra de código, y una versión modificada de la palabra de código.

La Figura 14 es un diagrama de una serie ejemplar de transformaciones de tamaño de subbanda de división y unión.

La Figura 15 es un diagrama de bloques de un entorno informático adecuado para implementar el codificador/decodificador de audio de la Figura 1 o 2.

#### Descripción detallada

15 La siguiente descripción detallada trata realizaciones de codificador/decodificador con codificación/decodificación de audio de datos espectrales de audio que usan modificación de palabras de código y/o modificación de una segmentación de frecuencia por defecto. Esta codificación/decodificación de audio representa algunos componentes de frecuencia que usan ruido conformado, o versiones conformadas de otros componentes de frecuencia o la combinación de ambos. Más particularmente, algunas bandas de frecuencia se representan como una versión conformada o transformación de otras bandas. Esto a menudo permite una reducción en tasa de bits a una calidad dada o una mejora en calidad a una tasa de bits dada. Opcionalmente, una configuración de frecuencia de subbanda inicial puede modificarse basándose en la tonalidad, energía o forma de los datos de audio.

20

#### Breve vista general

25 En la solicitud de patente, "Efficient coding of digital media spectral data using wide-sense perceptual similarity", Solicitud de Patente de Estados Unidos N. ° 10/882.801, presentada el 29 de junio 2004, se proporciona un algoritmo que permite la codificación de datos espectrales representando ciertas porciones de los datos espectrales como una versión cambiada de escala de un vector de código, donde el vector de código se elige desde cualquiera de un libro de códigos predeterminado fijo (por ejemplo, un libro de códigos de ruido), o un libro de códigos tomado desde una banda base (por ejemplo, un libro de códigos de banda base). Cuando el libro de códigos se crea de manera adaptativa, puede consistir de datos espectrales previamente codificados.

30

35 Se describen diversas características opcionales para modificar los vectores de código en el libro de códigos de acuerdo con algunas reglas que permiten que el vector de código represente mejor los datos que están representando. La modificación puede consistir en cualquiera de una transformación lineal o no lineal, o que represente el vector de código como una combinación de dos o más otros vectores de código originales o modificados. En el caso de una combinación, la modificación puede proporcionarse tomando las porciones de un vector de código y combinándolas con porciones de otros vectores de código.

40 Cuando se usa modificación de vector de código, tienen que enviarse bits de modo que el decodificador pueda aplicar la transformación para formar un nuevo vector de código. A pesar de los bits adicionales, la modificación de palabra de código es aún una codificación más eficaz para representar porciones de los datos espectrales que la codificación de forma de onda real de esa porción.

45 La tecnología descrita se refiere a mejorar la calidad de codificación de audio, y puede aplicarse también a otra codificación de multimedia tal como imágenes, vídeo y voz. Una mejora perceptual está disponible cuando se codifica audio, especialmente cuando la porción del espectro usada para formar el libro de códigos (típicamente la banda baja) tiene características diferentes que la porción que se está codificando usando ese libro de códigos (típicamente la banda alta). Por ejemplo, si la banda inferior tiene "picos" y por lo tanto tiene valores que están lejos de la media, y la banda alta no, o viceversa, entonces puede usarse esta técnica para codificar mejor la banda alta usando la banda baja como un libro de códigos.

Un vector es una subbanda de datos espectrales. Si los tamaños de subbanda son variables para una implementación dada, esto proporciona la oportunidad de dimensionar subbandas para mejorar la eficacia de

codificación. A menudo, las subbandas que tienen características similares pueden unirse con muy poco efecto sobre la calidad, mientras que las subbandas con datos altamente variables pueden representarse mejor si se divide una subbanda. Se describen diversos métodos para medir la tonalidad, energía o forma de una subbanda. Estas diversas mediciones se analizan a la luz de tomar decisiones de cuándo dividir o unir subbandas. Sin embargo, las subbandas más pequeñas (divididas) requieren más subbandas para representar los mismos datos espectrales. Por lo tanto, los tamaños de subbanda más pequeños requieren más bits para codificar la información. En casos donde se emplean tamaños de subbanda variables, se proporciona una configuración de subbanda para codificación eficaz de los datos espectrales, mientras se considera tanto los datos requeridos para codificar las subbandas como los datos requeridos para enviar la configuración de subbanda a un decodificador. Los siguientes párrafos continúan a través de ejemplos más generalizados a ejemplos más específicos.

#### Codificador y decodificador de audio generalizado

Las Figuras 1 y 2 son diagramas de bloques de un codificador (100) de audio generalizado y decodificador (200) de audio generalizado, en el que las técnicas descritas en el presente documento para codificación/decodificación de audio de datos espectrales de audio usan modificación de palabras de código y/o modificaciones de una segmentación de frecuencia inicial. Las relaciones mostradas entre módulos en el codificador y decodificar indican el flujo principal de información en el codificador y decodificador; otras relaciones no se muestran por motivos de simplicidad. Dependiendo de la implementación y el tipo de compresión deseada, los módulos del codificador o decodificador pueden añadirse, omitirse, dividirse en múltiples módulos, combinarse con otros módulos y/o sustituirse por módulos similares. En realizaciones alternativas, los codificadores o decodificadores con diferentes módulos y/u otras configuraciones de módulos miden calidad de audio perceptual.

Se describen detalles adicionales de un codificador/decodificador de audio en el que puede incorporarse la codificación/decodificación de datos espectrales de audio de similitud perceptual de sentido amplio en las siguientes solicitudes de Patente de Estados Unidos: Solicitud de Patente de Estados Unidos N. ° 10/882.801, presentada el 29/6/2004; Solicitud de Patente de Estados Unidos N. ° 10/020.708, presentada el 14/12/2001; Solicitud de Patente de Estados Unidos N. ° 10/016.918, presentada el 14/12/2001; Solicitud de Patente de Estados Unidos N. ° 10/017.702, presentada el 14/12/2001; Solicitud de Patente de Estados Unidos N. ° 10/017.861, presentada el 14/12/2001; y la Solicitud de Patente de Estados Unidos N. ° 10/017.694, presentada el 14/12/2001.

#### Codificador de audio generalizado ejemplar

El codificador (100) de audio generalizado incluye un transformador (110) de frecuencia, un transformador (120) de múltiples canales, un modelador (130) de percepción, un ponderador (140), un cuantificador (150), un codificador (160) por entropía, un controlador (170) de tasa/calidad y un multiplexor de flujo de bits ["MUX"] (180).

El codificador (100) recibe unas series de tiempo de muestras (105) de audio de entrada. Para entrada con múltiples canales (por ejemplo, modo estéreo), el codificador (100) procesa los canales de manera independiente, y puede funcionar con canales codificados conjuntamente siguiendo el transformador (120) de múltiples canales. El codificador (100) comprime las muestras (105) de audio y multiplexa información producida por los diversos módulos del codificador (100) para emitir un flujo de bits (195) en un formato tal como Windows Media Audio ["WMA"] o Formato de Flujo Continuo Avanzado ["ASF"]. Como alternativa, el codificador (100) funciona con otros formatos de entrada y/o salida.

El transformador (110) de frecuencia recibe las muestras (105) de audio y las convierte en datos en el dominio de la frecuencia. El transformador (110) de frecuencia divide las muestras (105) de audio en bloques, que pueden tener tamaño variable para permitir resolución temporal variable. Los bloques pequeños permiten mayor conservación de detalle de tiempo en segmentos de transición cortos pero activos en las muestras (105) de audio de entrada, pero sacrifican alguna resolución de frecuencia. En contraste, los bloques grandes tienen mejor resolución de frecuencia y pero resolución de tiempo, y normalmente permiten mayor eficacia de compresión en segmentos más largos y menos activos. Los bloques pueden solapar para reducir discontinuidades perceptibles entre bloques que podrían introducirse de otra manera por cuantificación posterior. El transformador (110) de frecuencia emite bloques de datos de coeficiente de frecuencia al transformador (120) de múltiples canales y emite información secundaria tal como tamaños de bloque al MUX (180). El transformador (110) de frecuencia emite tanto los datos de coeficiente de frecuencia como la información secundaria al modelador (130) de percepción.

El transformador (110) de frecuencia subdivide una trama de muestras (105) de audio de entrada en bloques de subtrama solapantes con tamaño variable en el tiempo y aplica una MLT variable en el tiempo a los bloques de subtrama. Los tamaños de subtrama ejemplares incluyen 128, 256, 512, 1024, 2048 y 4096 muestras. La MLT opera como una DCT modulada por una función de ventana de tiempo, donde la función de ventana es variable en el tiempo y depende de la secuencia de tamaños de subtrama. La MLT transforma un bloque solapante dado de muestras  $x[n], 0 \leq n < \text{tamaño\_subtrama}$  en un bloque de coeficientes de frecuencia  $X[k], 0 \leq k < \text{tamaño\_subtrama}/2$ . El transformador (110) de frecuencia puede emitir también estimaciones de la complejidad de tramas futuras al controlador (170) de tasa/calidad. Realizaciones alternativas usan otras variedades de MLT. En aún otras

realizaciones alternativas, el transformador (110) de frecuencia aplica una DCT, FFT u otro tipo de transformación de frecuencia solapada o no solapada, modulada o no modulada, o usa codificación de subbanda o de ondícula.

5 Para datos de audio de múltiples canales, los múltiples canales de datos de coeficiente de frecuencia producidos por el transformador (110) de frecuencia a menudo están correlacionados. Para aprovechar esta correlación, el transformador (120) de múltiples canales puede convertir los múltiples canales codificados independientemente originales en canales codificados conjuntamente. Por ejemplo, si la entrada es modo estéreo, el transformador (120) de múltiples canales puede convertir los canales izquierdo y derecho en canales de suma y diferencia:

$$X_{sum}[k] = \frac{X_{izquierda}[k] + X_{derecha}[k]}{2} \quad (1)$$

$$X_{dif}[k] = \frac{X_{izquierda}[k] - X_{derecha}[k]}{2} \quad (2)$$

10 O, el transformador (120) de múltiples canales puede pasar los canales izquierdo y derecho a través de canales codificados independientemente. Más en general, para un número de canales de entrada mayor de uno, el transformador (120) de múltiples canales pasa canales codificados independientemente originales sin cambios o convierte los canales originales en canales codificados conjuntamente. La decisión de usar canales codificados independiente o conjuntamente puede predeterminarse, o la decisión puede realizarse de manera adaptativa en una base bloque a bloque u otra durante la codificación. El transformador (120) de múltiples canales produce información secundaria al MUX (180) que indica el modo de transformación de canal usado.

15 El modelador (130) de percepción modela propiedades del sistema auditivo humano para mejorar la calidad de la señal de audio reconstruida para una tasa de bits dada. El modelador (130) de percepción calcula el patrón de excitación de un bloque de coeficientes de frecuencia de tamaño variable. En primer lugar, el modelador (130) de percepción normaliza el tamaño y escala de amplitud del bloque. Esto posibilita el efecto escala temporal posterior y establece una escala coherente para medidas de canal. Opcionalmente, el modelador (130) de percepción atenúa los coeficientes a ciertas frecuencias para modelar la función de transferencia del oído externo/medio. El modelador (130) de percepción calcula la energía de los coeficientes en el bloque y agrega las energías por 25 bandas críticas. Como alternativa, el modelador (130) de percepción usa otro número de bandas críticas (por ejemplo, 55 o 109). Los intervalos de frecuencia para las bandas críticas son dependientes de la implementación y se conocen bien numerosas opciones. Por ejemplo, véase el documento ITU-R BS 1387 o una referencia mencionada en el mismo. El modelador (130) de percepción procesa las energías de banda para contabilizar el enmascaramiento simultáneo y temporal. En realizaciones alternativas, el modelador (130) de percepción procesa los datos de audio de acuerdo con un modelo auditivo diferente, tal como el descrito o mencionado en el documento ITU-R BS 1387.

20 El ponderador (140) genera factores de ponderación (denominados como alternativa una matriz de cuantificación) basándose en el patrón de excitación recibido desde el modelador (130) de percepción y aplica los factores de ponderación a los datos recibidos desde el transformador (120) de múltiples canales. Los factores de ponderación incluyen un peso para cada una de múltiples bandas de cuantificación en los datos de audio. Las bandas de cuantificación pueden ser las mismas o diferentes en número o posición desde las bandas críticas usadas en cualquier otro lugar en el codificador (100). Los factores de ponderación indican proporciones en las que el ruido se expande a través de las bandas de cuantificación, con el objetivo de minimizar la capacidad audible del ruido poniendo más ruido en bandas donde es menos audible y viceversa. Los factores de ponderación pueden variar en amplitudes y número de bandas de cuantificación de bloque a bloque. En una implementación, el número de bandas de cuantificación varía de acuerdo con el tamaño de bloque; bloques más pequeños tienen menos bandas de cuantificación que bloques más grandes. Por ejemplo, bloques con 128 coeficientes tienen 13 bandas de cuantificación, bloques con 256 coeficientes tienen 15 bandas de cuantificación, hasta 25 bandas de cuantificación para bloques con 2048 coeficientes. Estas proporciones de bloque-banda son únicamente ejemplares. El ponderador (140) genera un conjunto de factores de ponderación para cada canal de datos de audio de múltiples canales en canales codificados independiente o conjuntamente, o genera un único conjunto de factores de ponderación para canales codificados conjuntamente. En realizaciones alternativas, el ponderador (140) genera los factores de ponderación desde información distinta de, o además de, los patrones de excitación.

25 El ponderador (140) emite bloques ponderados de datos de coeficiente al cuantificador (150) y emite información secundaria tal como el conjunto de factores de ponderación al MUX (180). El ponderador (140) puede emitir también los factores de ponderación al controlador (140) de tasa/calidad o a otros módulos en el codificador (100). El conjunto de factores de ponderación puede comprimirse para representación más eficaz. Si los factores de ponderación están comprimidos con pérdidas, los factores de ponderación reconstruidos se usan típicamente para ponderar los bloques de datos de coeficientes. Si la información de audio en una banda de un bloque se elimina completamente por alguna razón (por ejemplo, sustitución de ruido o truncación de banda), el codificador (100) puede mejorar adicionalmente la compresión de la matriz de cuantificación para el bloque.

5 El cuantificador (150) cuantifica la salida del ponderador (140), produciendo datos de coeficiente cuantificados al codificador (160) por entropía e información secundaria que incluye el tamaño de etapa de cuantificación al MUX (180). La cuantificación introduce pérdida de información irreversible, pero también permite que el codificador (100) regule la tasa de bits del flujo de bits (195) de salida en conjunto con el controlador (170) de tasa/calidad. En la Figura 1, el cuantificador (150) es un cuantificador escalar uniforme adaptativo. El cuantificador (150) aplica el mismo tamaño de etapa de cuantificación a cada coeficiente de frecuencia, pero el mismo tamaño de etapa de cuantificación puede cambiar de una iteración a la siguiente para afectar a la tasa de bits de la salida del codificador (160) por entropía. En realizaciones alternativas, el cuantificador es un cuantificador no uniforme, un cuantificador vectorial y/o un cuantificador no adaptativo.

10 El codificador (160) por entropía comprime de manera sin pérdidas datos de coeficiente cuantificados recibidos desde el cuantificador (150). Por ejemplo, el codificador (160) por entropía usa codificación por longitud de serie de múltiples niveles, codificación de longitud variable a variable, codificación por longitud de serie, codificación Huffman, codificación de diccionario, codificación aritmética, codificación LZ, una combinación de las anteriores o alguna otra técnica de codificación.

15 El controlador (170) de tasa/calidad funciona con el cuantificador (150) para regular la tasa de bits y calidad de la salida del codificador (100). El controlador (170) de tasa/calidad recibe información desde otros módulos del codificador (100). En una implementación, el controlador (170) de tasa/calidad recibe estimaciones de complejidad futura desde el transformador (110) de frecuencia, tasa de muestreo, información de tamaño de bloque, el patrón de excitación de datos de audio original desde el modelador (130) de percepción, factores de ponderación desde el ponderador (140), un bloque de información de audio cuantificada en alguna forma (por ejemplo, cuantificada, reconstruida o codificada), y almacena en memoria intermedia información de estado desde el MUX (180). El controlador (170) de tasa/calidad puede incluir un cuantificador inverso, un ponderador inverso, un transformador de múltiples canales inverso, y, potencialmente, un decodificador por entropía y otros módulos, para reconstruir los datos de audio de una forma cuantificada.

20 El controlador (170) de tasa/calidad procesa la información para determinar un tamaño de etapa de cuantificación deseado dadas condiciones actuales y emite el tamaño de etapa de cuantificación al cuantificador (150). El controlador (170) de tasa/calidad a continuación mide la calidad de un bloque de datos de audio reconstruidos como cuantificados con el tamaño de etapa de cuantificación, como se describe a continuación. Usando la calidad medida así como información de tasa de bits, el controlador (170) de tasa/calidad ajusta el tamaño de etapa de cuantificación con el objetivo de satisfacer restricciones de tasa de bits y calidad, tanto de manera instantánea como a largo plazo. En realizaciones alternativas, el controlador (170) de tasa/calidad funciona con información diferente o adicional o aplica diferentes técnicas para regular la calidad y tasa de bits.

25 En conjunto con el controlador (170) de tasa/calidad, el codificador (100) puede aplicar sustitución de ruido, truncación de banda y/o rematrización de múltiples canales a un bloque de datos de audio. A tasas de bits bajas y medias, el codificador (100) de audio puede usar sustitución de ruido para transportar información en ciertas bandas. En truncación de banda, si la calidad medida para un bloque indica calidad pobre, el codificador (100) puede eliminar completamente los coeficientes en ciertas bandas (normalmente de frecuencia más alta) para mejorar la calidad global en las bandas restantes. En rematrización de múltiples canales, para datos de audio de múltiples canales de baja tasa de bits en canales codificados conjuntamente, el codificador (100) puede suprimir información en ciertos canales (por ejemplo, el canal de diferencia) para mejorar la calidad del canal o canales restantes (por ejemplo, el canal de suma).

30 El MUX (180) multiplexa la información secundaria recibida desde los otros módulos del codificador (100) de audio junto con los datos codificados por entropía recibidos desde el codificador (160) por entropía. El MUX (180) emite la información en WMA o en otro formato que reconoce un decodificador de audio.

35 El MUX (180) incluye una memoria intermedia virtual que almacena el flujo de bits (195) para emitirse mediante el codificador (100). La memoria intermedia virtual almacena una duración predeterminada de información de audio (por ejemplo, 5 segundos para enviar por flujo continuo audio) para suavizar sobre fluctuaciones a corto plazo en la tasa de bits debido a cambios de complejidad en el audio. La memoria intermedia virtual a continuación emite datos a una tasa de bits relativamente constante. La amplitud actual de la memoria intermedia, la tasa de cambio de amplitud de la memoria intermedia y otras características de la memoria intermedia pueden usarse mediante el controlador (170) de tasa/calidad para regular la calidad y tasa de bits.

#### Decodificador de audio generalizado ejemplar

40 Con referencia a la Figura 2, el decodificador (200) de audio generalizado incluye un demultiplexor de flujo de bits ["DEMUX"] (210), un decodificador (220) por entropía, un cuantificador (230) inverso, un generador (240) de ruido, un ponderador (250) inverso, un transformador (260) de múltiples canales inverso y un transformador (270) de frecuencia inverso. El decodificador (200) es más simple de lo que lo es el codificador (100) puesto que el decodificador (200) no incluye módulos para control de tasa/calidad.

5 El decodificador (200) recibe un flujo de bits (205) de datos de audio comprimidos en WMA u otro formato. El flujo de bits (205) incluye datos codificados por entropía así como información secundaria desde la que el decodificador (200) reconstruye muestras (295) de audio. Para datos de audio con múltiples canales, el decodificador (200) procesa cada canal de manera independiente, y puede funcionar con canales codificados conjuntamente antes del transformador (260) de múltiples canales inverso.

El DEMUX (210) analiza información en el flujo de bits (205) y envía información a los módulos del decodificador (200). El DEMUX (210) incluye una o más memorias intermedias para compensar variaciones a corto plazo en la tasa de bits debido a fluctuaciones en complejidad del audio, inestabilidad de red y/u otros factores.

10 El decodificador (220) por entropía descomprime de manera sin pérdidas códigos de entropía recibidos desde el DEMUX (210), produciendo datos de coeficiente de frecuencia cuantificados. El decodificador (220) por entropía típicamente aplica la inversa de la técnica de codificación por entropía usada en el codificador.

15 El cuantificador (230) inverso recibe un tamaño de etapa de cuantificación desde el DEMUX (210) y recibe datos de coeficiente de frecuencia cuantificados desde el decodificador (220) por entropía. El cuantificador (230) inverso aplica el tamaño de etapa de cuantificación a los datos de coeficiente de frecuencia cuantificados para reconstruir parcialmente los datos de coeficiente de frecuencia. En realizaciones alternativas, el cuantificador inverso aplica la inversa de alguna otra técnica de cuantificación usada en el codificador.

El generador (240) de ruido recibe desde el DEMUX (210) la indicación de qué bandas en un bloque de datos están sustituidas en ruido así como cualquier parámetro para la forma del ruido. El generador (240) de ruido genera los patrones para las bandas indicadas y pasa la información al ponderador (250) inverso.

20 El ponderador (250) inverso recibe los factores de ponderación desde el DEMUX (210), patrones para cualquier banda de ruido sustituido desde el generador (240) de ruido, y los datos de coeficiente de frecuencia parcialmente reconstruidos desde el cuantificador (230) inverso. Según sea necesario, el ponderador (250) inverso descomprime los factores de ponderación. El ponderador (250) inverso aplica los factores de ponderación a los datos de coeficiente de frecuencia parcialmente reconstruidos para bandas que no se han sustituido en ruido. El ponderador  
25 (250) inverso a continuación añade los patrones de ruido recibidos desde el generador (240) de ruido.

30 El transformador (260) de múltiples canales inverso recibe los datos de coeficiente de frecuencia reconstruidos desde el ponderador (250) inverso e información de modo de transformación de canal desde el DEMUX (210). Si los datos de múltiples canales están en canales codificados independientemente, el transformador (260) de múltiples canales inverso pasa los canales a través. Si los datos de múltiples canales están en canales codificados conjuntamente, el transformador (260) de múltiples canales inverso convierte los datos en canales codificados independientemente. Si se desea, el decodificador (200) puede medir la calidad de los datos de coeficiente de frecuencia reconstruidos en este punto.

35 El transformador (270) de frecuencia inverso recibe los datos de coeficiente de frecuencia emitidos mediante el transformador (260) de múltiples canales así como información secundaria tal como tamaños de bloque desde el DEMUX (210). El transformador (270) de frecuencia inverso aplica la inversa de la transformación de frecuencia usada en el codificador y emite bloques, de muestras (295) de audio reconstruidas.

Codificación/decodificación ejemplar con palabras de código modificadas y similitud perceptual de sentido amplio

40 La Figura 3 ilustra una implementación de un codificador (300) de audio que usa codificación con configuración de subbanda adaptativa y/o palabras de código modificadas tal como, con similitud perceptual de sentido amplio, que puede incorporarse en el proceso de codificación/decodificación de audio global del codificador (100) y decodificador (200) de audio generalizado de las Figuras 1 y 2. En esta implementación, el codificador (300) de audio realiza una descomposición espectral en la transformación (320), usando cualquiera de una transformación de subbanda o una transformación ortogonal solapada tal como MDCT o MLT, para producir un conjunto de coeficientes espectrales para cada bloque de entrada de la señal de audio. Como es conocido de manera convencional, el codificador de  
45 audio codifica estos coeficientes espectrales para enviar en el flujo de bits de salida al decodificador. La codificación de los valores de estos coeficientes espectrales constituye la mayoría de la tasa de bits usada en un códec de audio. A tasas de bits bajas, el codificador (300) de audio selecciona codificar menos de los coeficientes espectrales usando un codificador (340) de banda base (es decir, un número de coeficientes que pueden codificarse dentro de un porcentaje del ancho de banda de los coeficientes espectrales emitidos desde el transformador (110) de  
50 frecuencia), tal como una porción inferior o banda base del espectro. El codificador (340) de banda base codifica estos coeficientes espectrales de banda base usando una sintaxis de codificación conocida de manera convencional, como se describe para el codificador de audio generalizado anterior. Esto daría como resultado en general que el audio reconstruido sonara apagado o con filtro de paso bajo.

55 El codificador (300) de audio evita el efecto de apagado/paso bajo codificando también los coeficientes espectrales omitidos usando configuración de subbanda adaptativa y/o palabras de código modificadas con similitud perceptual



de sentido amplio. Los coeficientes espectrales (denominados en el presente documento como los “coeficientes espectrales de banda extendida”) que se omitieron de la codificación con el codificador (340) de banda base se codifican mediante el codificador (350) de banda extendida como ruido conformado, o versiones conformadas de otros componentes de frecuencia, o dos o más combinaciones de los dos. Más específicamente, los coeficientes espectrales de banda extendida se dividen en un número de subbandas de diversos tamaños y potencialmente diferentes (por ejemplo, de típicamente 16, 32, 64, 128, 256, ..., etc., coeficientes espectrales), que se codifican como ruido conformado o versiones conformadas de otros componentes de frecuencia. Esto añade una versión perceptualmente placentera del coeficiente espectral faltante para proporcionar un sonido completamente más rico. Incluso aunque el espectro real pueda desviarse de la versión sintética resultante de esta codificación, esta codificación de banda extendida proporciona un efecto perceptual similar que en el original.

En algunas implementaciones, la anchura de la banda base (es decir, número de coeficientes espectrales de banda base codificados usando el codificador 340 de banda base) así como el tamaño o número de bandas extendidas puede variarse de una configuración por defecto o inicial. En tal caso, la anchura de la banda base y/o número (o tamaño) de bandas extendidas codificadas usando el codificador (350) de banda base extendida puede codificarse (360) en el flujo (195) de salida.

Si es deseable, el particionamiento del flujo de bits entre los coeficientes espectrales de banda base y coeficientes de banda extendida en el codificador (300) de audio se hace para asegurar compatibilidad hacia atrás con decodificadores existentes basándose en la sintaxis de codificación del codificador de banda base, de manera que tal decodificador existente pueda decodificar la porción de banda base codificada mientras se ignora la porción extendida. El resultado es que los decodificadores más modernos tienen la capacidad para representar el espectro completo cubierto por el flujo de bits codificado de banda extendida, mientras que los decodificadores más antiguos pueden representar la porción que el codificador elige para codificar con la sintaxis existente. El límite de frecuencia (por ejemplo, el límite entre la porción de banda base y extendida) puede ser flexible y variable en el tiempo. Puede decidirse mediante el codificador basándose en características de señal y enviarse explícitamente al decodificador, o puede ser una función del espectro decodificado, por lo que no necesita enviarse. Puesto que los decodificadores existentes pueden decodificar únicamente la porción que se codifica usando el códec existente (banda base), esto significa que la porción inferior del espectro (por ejemplo, banda base) se codifica con el códec existente y la porción superior se codifica usando la codificación de banda extendida con palabras de código modificadas usando similitud perceptual de sentido amplio

En otras implementaciones donde tal compatibilidad hacia atrás no sea necesaria, el codificador tiene la libertad de elegir entre la codificación de banda base convencional y la banda extendida (con palabras de código modificadas y enfoque de similitud perceptual de sentido amplio) solamente basándose en características de señal y el coste de codificar sin considerar la localización de límite de frecuencia. Por ejemplo, aunque es altamente improbable en señales naturales, puede ser mejor codificar la frecuencia superior con el códec tradicional y la porción inferior usando el códec extendido.

#### Método de codificación ejemplar

La Figura 4 es un diagrama de flujo que representa un proceso (400) de codificación de audio realizado mediante el codificador (350) de banda base extendida de la Figura 3 para codificar los coeficientes espectrales de banda extendida. En este proceso (400) de codificación de audio, el codificador (350) de banda base extendida divide los coeficientes espectrales de banda extendida en un número de subbandas. En una implementación típica, estas subbandas en general consisten en 64 o 128 coeficientes espectrales cada una. Como alternativa, pueden usarse otras subbandas de tamaño (por ejemplo, 16, 32 u otros números de coeficientes espectrales). Si un codificador de banda extendida proporciona la posibilidad de modificar el tamaño de las subbandas, un proceso (360) de configuración de banda extendida modifica las subbandas y codifica la configuración de banda extendida. Las subbandas pueden ser disjuntas o pueden solaparse (usando generación de ventanas). Con subbandas solapantes, se codifican más bandas. Por ejemplo, si tienen que codificarse 128 coeficientes espectrales usando el codificador de banda base extendida con subbandas de tamaño 64, el método usará dos bandas disjuntas para codificar los coeficientes, codificando los coeficientes 0 a 63 como una subbanda y los coeficientes 64 a 127 como la otra. Como alternativa, pueden usarse tres bandas solapantes con el 50 % de solapamiento, codificando 0 a 63 como una banda, 32 a 95 como otra banda y 64 a 127 como la tercera banda. Se analizarán más tarde diversos otros métodos dinámicos para segmentación de frecuencia de subbandas en esta memoria descriptiva.

Para cada una de estas subbandas fijas o dinámicamente optimizadas, el codificador (350) de banda base extendida codifica la banda usando dos parámetros. Un parámetro (“parámetro de escala”) es un factor de escala que representa la energía total en la banda. El otro parámetro (“parámetro de forma”, en general en forma de un vector de movimiento) se usa para representar la forma del espectro en la banda. Opcionalmente, como se analizará, el parámetro de forma requerirá uno o más bits de transformación de forma que indican un exponente, una dirección de vector (por ejemplo, hacia delante/inversa) y/o una transformación de signo de coeficiente.

Como se ilustra en el diagrama de flujo de la Figura 4, el codificador (350) de banda base extendida realiza el

proceso (400) para cada subbanda de la banda extendida. En primer lugar (en 420), el codificador (350) de banda base extendida calcula el factor de escala. En una implementación, el factor de escala es simplemente el valor de rms (media cuadrática) de los coeficientes dentro de la subbanda actual. Se halla tomando la raíz cuadrada del valor promedio cuadrado de todos los coeficientes. El valor promedio cuadrado se halla tomando la suma del valor cuadrado de todos los coeficientes en la subbanda, y dividiendo por el número de coeficientes.

El codificador (350) de banda base extendida a continuación determina el parámetro de forma. El parámetro de forma normalmente es un vector de movimiento que indica simplemente copiar sobre una versión normalizada del espectro de una porción del espectro que ya se ha codificado (es decir, una porción de los coeficientes espectrales de banda base codificados con el codificador de banda base). En ciertos casos, el parámetro de forma puede especificar en su lugar un vector de ruido aleatorio normalizado o simplemente un vector para una forma espectral desde un libro de códigos fijado. Copiar la forma desde otra porción del espectro es útil en audio puesto que típicamente en muchas señales tonales, hay componentes armónicos que se repiten a través de todo el espectro. El uso de ruido o algún otro libro de códigos fijado permite una codificación de baja tasa de bits de aquellos componentes que no están bien representados en la porción codificada de banda base del espectro. Por consiguiente, el proceso (400) proporciona un método de codificación que es esencialmente una codificación de cuantificación de vector de forma de ganancia de estas bandas, donde el vector es la banda de frecuencia de coeficientes espectrales, y el libro de códigos se toma del espectro previamente codificado y puede incluir otros vectores fijados o vectores de ruido aleatorio, también. Es decir, cada subbanda codificada mediante el codificador de banda base extendida se representa como  $a^*X$ , donde 'a' es un parámetro de escala y 'X' es un vector representado por el parámetro de forma, y puede ser una versión normalizada de (cualquier) coeficiente espectral previamente codificado, un vector desde un libro de códigos fijado o un vector de ruido aleatorio. También, si esta porción copiada del espectro se añade a una codificación tradicional de esa misma porción, entonces esta adición es una codificación residual. Esto podría ser útil si una codificación tradicional de la señal proporciona una representación de base (por ejemplo, codificación del suelo espectral) que es fácil de codificar con unos pocos bits, y el resto se codifica con el nuevo algoritmo.

Más específicamente, en la acción (430), el codificador (350) de banda base extendida busca los coeficientes espectrales de banda base (u otra previamente codificada) para un vector en la banda base de coeficientes espectrales que tienen una forma similar como la subbanda actual. Como se ha establecido anteriormente, una "palabra de código desde la banda base" también incluye fuentes fuera de la banda base presente. El codificador de banda base extendida determina qué porción de la banda base (u otra banda previa) es más similar a la subbanda actual usando una comparación de media mínima cuadrática a una versión normalizada de cada porción de la banda base. Opcionalmente, se aplica una transformación (431) lineal o no lineal a una o más porciones del espectro en la banda base (u otra banda previa) para crear un universo mayor de formas para coincidencia. De nuevo, la banda base incluye la biblioteca y otras bandas previas cuando se analizan fuentes para palabras de código. Opcionalmente, el codificador de banda extendida realiza una o más transformaciones lineales o no lineales en la banda base y/o libro de códigos fijados para proporcionar una biblioteca más grande de formas disponibles para coincidencia. Por ejemplo, considérese un caso en el que hay 256 coeficientes espectrales producidos por la transformación (320) desde un bloque de entrada, las subbandas de banda extendida (en este ejemplo) son cada una 16 coeficientes espectrales en anchura, y el codificador de banda base codifica los primeros 128 coeficientes espectrales (numerados de 0 a 127) como la banda base. A continuación, la búsqueda realiza una comparación de media mínima cuadrática de los 16 coeficientes espectrales normalizados en cada banda extendida a una versión normalizada de cada una de 16 porciones de coeficientes espectrales de la banda base (o cualquier banda previamente codificada) comenzando en las posiciones de coeficiente 0 a 111 (es decir, un total de 112 posibles formas espectrales diferentes codificadas en la banda base en este caso). La porción de banda base que tiene el valor de media cuadrática mínima más bajo se considera la más cercana (más similar) en forma a la banda extendida actual. Opcionalmente, la búsqueda realiza la comparación de media cuadrática mínima en las transformaciones (431) lineales o no lineales de la banda base (u otras bandas). En la acción (432), el codificador de banda base extendida comprueba si esta banda más similar de los coeficientes espectrales de banda base es suficientemente cercana en forma a la banda extendida actual (por ejemplo, el valor de media cuadrática mínima es inferior a un umbral preseleccionado). En caso afirmativo, a continuación el codificador de banda base extendida determina un vector de movimiento que apunta a esta banda coincidente más cercana de coeficientes espectrales de banda base en la acción (434) y opcionalmente, información acerca de una transformación lineal o no lineal en el vector de movimiento mejor coincidente. El vector de movimiento puede ser la posición de coeficiente de inicio en la banda base (por ejemplo, 0 a 111 en el ejemplo). Pueden usarse otros métodos (tal como comprobar tonalidad frente a no tonalidad) para observar si la banda más similar de los coeficientes de banda base (u otras bandas) está suficientemente cercana en forma a la banda extendida actual.

Si no se halla porción suficientemente similar de la banda base, el codificador de banda base extendida a continuación mira un libro de códigos (440) fijado de formas espectrales para representar la subbanda actual. El codificador de banda base extendida busca en este libro de códigos (440) fijado una forma espectral similar a la de la subbanda actual. Opcionalmente, la búsqueda realiza las comparaciones de media cuadrática mínima en las transformaciones (431) lineales o no lineales del libro de códigos fijado. Si se halla, el codificador de banda base extendida usa su índice en el libro de códigos como el parámetro de forma en la acción (444) y opcionalmente, información acerca de una transformación lineal o no lineal en el mejor índice coincidente en el libro de códigos. De

otra manera, en la acción (450), el codificador de banda base extendida puede determinar también representar la forma de la subbanda actual como un vector de ruido aleatorio normalizado.

5 En implementaciones alternativas, el codificador de banda extendida puede decidir si los coeficientes espectrales pueden representarse usando ruido incluso antes de buscar la mejor forma espectral en la banda base. De esta manera incluso si se hallara una forma espectral suficientemente cercana en la banda base, el codificador de banda base extendida codificará aún esa porción usando ruido aleatorio. Esto puede dar como resultado menos bits cuando se compara con enviar el vector de movimiento que corresponde a una posición en la banda base.

10 En la acción (460), el codificador de banda extendida codifica los parámetros de escala y forma (es decir, factor de cambio de escala y vector de movimiento en esta implementación, y opcionalmente, información de transformación lineal y no lineal) usando codificación predictiva, cuantificación y/o codificación por entropía. En una implementación, por ejemplo, el parámetro de escala se codifica de manera predictiva basándose en la subbanda extendida inmediatamente precedente. (Los factores de cambio de escala de las subbandas de la banda extendida típicamente son similares en valor, de modo que sucesivas sub-bandas típicamente tienen factores de cambio de escala cercanos en valor). En otras palabras, se codifica el valor total del factor de cambio de escala para la primera subbanda de la banda extendida. Las subbandas posteriores se codifican como su diferencia de su valor real de su valor previsto (es decir, siendo el valor previsto el factor de cambio de escala de la subbanda precedente). Para 15 audio de múltiples canales, se codifica la primera subbanda de la banda extendida en cada canal como su valor completo, y se predicen factores de cambio de escala de subbandas posteriores de aquellas de la subbanda precedente en el canal. En implementaciones alternativas, el parámetro de escala también puede predecirse a través de los canales, para más de otra subbanda, desde el espectro de banda base o desde bloques de audio de 20 entrada previos, entre otras variaciones.

El codificador de banda base extendida cuantifica adicionalmente el parámetro de escala usando cuantificación uniforme o no uniforme. En una implementación, se usa una cuantificación no uniforme del parámetro de escala, en la que un logaritmo del factor de cambio de escala se cuantifica uniformemente a 128 segmentos. El valor 25 cuantificado resultante a continuación se codifica por entropía usando codificación Huffman.

Para el parámetro de forma, el codificador de banda base extendida también usa codificación predictiva (que puede predecirse desde la subbanda precedente como para el parámetro de escala), cuantificación a 64 segmentos, y codificación por entropía (por ejemplo, con codificación Huffman).

30 En algunas implementaciones, las subbandas de banda extendida pueden ser variables en tamaño. En tales casos, el codificador de banda base extendida también codifica la configuración de la banda extendida.

Más particularmente, en una implementación de ejemplo, el codificador de banda base extendida codifica los parámetros de escala y de forma como se muestra mediante el listado de pseudocódigo en la Tabla 1. Más de un parámetro de escala o de forma puede enviarse para el caso de múltiples palabras de código.

Tabla 1

```

para cada pieza en flujo de audio
{
  para cada canal en pieza que puede necesitar codificarse (por ejemplo, altavoz
de graves puede no necesitar codificarse)
  {
    1 bit para indicar si el canal está codificado o no.
    8 bits para especificar versión cuantificada de posición de inicio de banda
extendida.
    'n_config' bits para especificar codificación de configuración de banda.
    para cada subbanda a codificar usando codificador de banda extendida
    {
      'n_scale' bits para código de longitud variable para especificar
parámetro de escala (energía en la banda).
      'n_shape' bits para código de longitud variable para especificar
parámetro de forma.
      'n_transformation' bits para parámetros de transformación lineal/no
lineal.
    }
  }
}

```

35

En el listado de código anterior, la codificación para especificar la configuración de banda (es decir, el número de bandas y sus tamaños) depende del número de coeficientes espectrales a codificarse usando el codificador de

banda base extendida. El número de coeficientes codificados usando el codificador de banda base extendida puede encontrarse usando la posición de inicio de la banda extendida y el número total de coeficientes espectrales (número de coeficientes espectrales codificados usando el codificador de banda extendida = número total de coeficientes espectrales - posición de inicio). En un ejemplo, la configuración de banda se codifica a continuación como un índice en el listado de todas las configuraciones posibles permitidas. Este índice se codifica usando un código de longitud fija con  $n\_config = \log_2(\text{número de configuraciones})$  bits. Las configuraciones permitidas son una función de número de coeficientes espectrales a codificarse usando este método. Por ejemplo, si se han de codificar 128 coeficientes, la configuración por defecto es 2 bandas de tamaño 64. Pueden ser posibles otras configuraciones, por ejemplo, la Tabla 2 muestra un listado de configuraciones de banda para 128 coeficientes espectrales.

Tabla 2				
0:	128			
1:	64		64	
2:	64		32	32
3:	32	32	64	
4:	32	32	32	32

Por lo tanto, en este ejemplo, hay 5 posibles configuraciones de banda. En una configuración de este tipo, se elige una configuración por defecto para los coeficientes como que tiene 'n' bandas. A continuación, permitiendo a cada banda dividirse o unirse (únicamente un nivel), hay  $5^{(n/2)}$  posibles configuraciones, que requieren  $(n/2)\log_2(5)$  bits para codificar. En otras implementaciones, puede usarse codificación de longitud variable para codificar la configuración. No se requiere método específico de configuración de banda extendida para beneficiarse de la modificación de palabra de código. Adicionalmente, se analizan más tarde diversos otros métodos para configuración de banda extendida que no requieren ningún método de modificación de palabra de código de este tipo para ser beneficiosos.

Como se ha analizado anteriormente, el factor de escala se codifica usando codificación predictiva, donde la predicción puede tomarse desde factores de escala previamente codificados desde bandas previas en el mismo canal, desde canales previos en la misma pieza, o desde piezas previamente decodificadas. Para una implementación dada, la elección para la predicción puede realizarse mirando qué banda previa (en la misma banda extendida, canal o pieza (bloque de entrada)) proporcionó las correlaciones más altas. En una implementación ejemplo, la banda se codifica predictivamente como sigue:

- Sean los factores de escala en una pieza  $x[i][j]$ , donde  $i$ =índice de canal,  $j$ =índice de banda.
- Para  $i=0 \ \&\& \ j=0$  (primer canal, primera banda), sin predicción.
- Para  $i!=0 \ \&\& \ j=0$  (otros canales, primera banda), predicción es  $x[0][0]$  (primer canal, primera banda)
- Para  $i!=0 \ \&\& \ j!=0$  (otros canales, otras bandas), predicción es  $x[i][j-1]$  (mismo canal, banda previa).

En la tabla de código anterior, el "parámetro de forma" es un vector de movimiento que especifica la localización de la palabra de código previa de coeficientes espectrales, o vector desde el libro de códigos fijado o ruido. Los coeficientes espectrales previos pueden provenir desde el mismo canal, o desde canales previos, o desde piezas previas. El parámetro de forma se codifica usando predicción, donde la predicción se toma desde localizaciones previas desde bandas previas en el mismo canal o canales previos en la misma pieza o desde piezas previas. Cualquier transformación lineal o no lineal puede aplicarse a una forma. El parámetro de "transformación" indica tal información de transformación, índice para información de transformación o etc.

Método de decodificación ejemplar

La Figura 5 muestra un decodificador (500) de audio para el flujo de bits producido por el codificador (300) de audio. En este decodificador, el flujo de bits (205) codificado se demultiplexa (por ejemplo, basándose en la anchura de banda base codificada y configuración de banda extendida) mediante el demultiplexor (210) de flujo de bits en el flujo de código de banda base y flujo de código de banda extendida, que se decodifican en el decodificador (540) de banda base y el decodificador (550) de banda extendida. El decodificador (540) de banda base decodifica los coeficientes espectrales de banda base usando decodificación convencional del códec de banda base. El decodificador (545) de configuración de banda extendida decodifica los tamaños de banda optimizados si se utiliza la optimización desde una configuración de banda por defecto. El decodificador (550) de banda extendida decodifica el flujo de código de banda extendida, que incluye copiar a través de una o más porciones de los coeficientes espectrales de banda base originales o transformados (o cualquier banda o libro de códigos previo) apuntados mediante el vector de movimiento de los parámetros de forma (y cualquier información opcional sobre la transformación lineal o no lineal del coeficiente apuntado mediante el vector de movimiento) y cambiar de escala mediante el factor de cambio de escala del parámetro de escala. Los coeficientes espectrales de banda base y banda extendida se combinan en un único espectro que se convierte mediante transformación 580 inversa para

reconstruir la señal de audio.

La Figura 6 muestra un proceso (600) de decodificación usado en el decodificador (550) de banda extendida de la Figura 5. Para cada subbanda codificada de la banda extendida en el flujo de código de banda extendida (acción (610)), el decodificador de banda extendida decodifica el factor de escala (acción (620)) y el vector de movimiento junto con cualquier información de transformación (acción (630)). El decodificador de banda extendida a continuación copia (acción (640)) la subbanda de banda base, vector de libro de códigos fijado, o vector de ruido aleatorio identificado mediante el vector de movimiento (parámetro de forma y realiza cualquier transformación identificada). El decodificador de banda extendida cambia de escala la banda o vector espectral copiado mediante el factor de cambio de escala para producir los coeficientes espectrales para la subbanda actual de la banda extendida.

Coeficientes espectrales ejemplares

La Figura 7 es un gráfico que representa un conjunto de coeficientes espectrales. Por ejemplo, los coeficientes (700) son una salida de una transformación o una transformación ortogonal solapada tal como MDCT o MCT, para producir un conjunto de coeficientes espectrales para cada bloque de entrada de la señal de audio.

Como se muestra en la Figura 7, una porción de la salida de la transformación denominada la banda base (702) se codifica mediante el codificador de banda base. A continuación la banda extendida (704) se divide en subbandas de tamaños (706) homogéneos o variados. Las formas en la banda base (708) (por ejemplo, formas según se representa mediante una serie de coeficientes) se comparan a formas en la banda extendida (710), y se usa un desplazamiento (712) que representa una forma similar en la banda base para codificar una forma (por ejemplo, subbanda) en la banda extendida de modo que necesitan codificarse y enviarse menos bits al decodificador.

Un tamaño de banda base (702) puede variar, y una banda extendida (704) resultante puede variar basándose en la banda base. La banda extendida puede dividirse en diversos y múltiples tamaños (706) de subbanda de tamaño.

En este ejemplo, un segmento de banda base (desde esta o cualquier banda anterior) se usa para identificar una palabra de código (708) para simular una subbanda en la banda extendida (710). La palabra de código (708) puede transformarse linealmente o transformarse no linealmente para crear otras formas (por ejemplo, otras series de coeficientes) que pueden proporcionar más estrechamente un modelo para el vector (710) que se está codificando.

Por lo tanto, se usan diversos segmentos en la banda base como modelos potenciales (por ejemplo, un libro de códigos, biblioteca o diccionario de palabras de código) para codificar datos en la banda extendida. En lugar de enviar los coeficientes (710) reales en una subbanda en la banda extendida se envía un identificador tal como un desplazamiento (712) de vector de movimiento, al codificador para representar los datos para la banda extendida. Sin embargo, en ocasiones no hay coincidencias cercanas en la banda base para datos que se están modelando en una subbanda. Esto puede ser debido a restricciones de baja tasa de bits que pueden permitir una banda base de tamaño limitado. Como se establece, el tamaño (702) de la banda base en relación a la banda extendida puede variar basándose en recursos informáticos tales como el tiempo, dispositivo de salida o ancho de banda.

En otro ejemplo, se proporciona otro libro de códigos (716) o está disponible para el codificador/decodificador, y se proporciona un identificador de mejor coincidencia como un índice a una palabra de código (718) de coincidencia más cercana en el libro de códigos. Adicionalmente, en casos donde es deseable ruido aleatorio como una palabra de código, una porción del flujo de bits (tal como bits desde la banda base) pueden usarse para sembrar de manera similar un generador de números aleatorios tanto en el codificador como el decodificador.

Estos diversos métodos pueden usarse para crear una biblioteca o diccionario de palabras de código para proporcionar un universo de palabras de código para hacer coincidir una forma, para codificar una subbanda (710) u otro vector, de modo que los mismos coeficientes pueden modelarse mediante un vector (712) de movimiento en lugar de cuantificarse individualmente.

Transformaciones de palabras de código ejemplares

La Figura 8 es un gráfico de una palabra de código y diversas transformaciones lineales y no lineales de la palabra de código. Por ejemplo, una palabra de código (802) proviene de una banda base, un libro de códigos fijado y/o una palabra de código generada aleatoriamente. Se realizan transformaciones lineales o no lineales en una o más palabras de código en una biblioteca para obtener un conjunto mayor o más diverso de formas para identificar una mejor forma para hacer coincidir un vector que se está codificando. En un ejemplo, una palabra de código se invierte (804) en orden de coeficiente para obtener otra palabra de código para coincidencia de forma. Una inversa de una palabra de código que contiene los valores de coeficiente  $< 1, 1,5, 2,2, 3,2 >$  se vuelve  $< 3,2, 2,2, 1,5, 1 >$ . En otro ejemplo, el rango dinámico o varianza de una palabra de código se reduce (806) usando exponenciación con un exponente menor que uno en cada coeficiente. De manera similar, una variación de la palabra de código se exagera (por ejemplo, varianza aumentada) usando un exponente mayor que uno, no mostrado. Por ejemplo, una palabra de

5 código que contiene los coeficientes  $\langle 1, 1, 2, 1, 4, 2, 1 \rangle$  se eleva a la potencia de 2 para crear la palabra de código  $\langle 1, 1, 4, 1, 16, 4, 1 \rangle$ . En otro ejemplo, los coeficientes de una palabra de código  $\langle -1, 1, 2, 3 \rangle$  (802) se niegan  $\langle 1, -1, -2, -3 \rangle$  (808). Por supuesto, pueden realizarse muchas otras transformaciones lineales y no lineales (por ejemplo, 806) en una o más palabras de código para proporcionar un universo o biblioteca mayor o más diverso para hacer coincidir subbandas u otros vectores. Adicionalmente, una o más transformaciones pueden aplicarse también en combinación a las palabras de código para proporcionar mayor diversidad de formas disponibles.

10 En un ejemplo, un codificador en primer lugar determina una palabra de código en la banda base que es una coincidencia más cercana a una subbanda que se está codificando. Por ejemplo, puede usarse una comparación de media mínima cuadrática de coeficientes en la banda base para determinar una mejor coincidencia. Por ejemplo, después de comparar (708) a (710), la comparación mueve un coeficiente hacia abajo del espectro, un coeficiente a la vez, para obtener otra palabra de código para comparar (710). A continuación cuando se encuentra una coincidencia más cercana, en un ejemplo, la forma de la palabra de código de mejor coincidencia se varía mediante transformación no lineal para ver si la coincidencia puede mejorarse. Por ejemplo, usando una transformación de exponente en los coeficientes de una palabra de código de mejor coincidencia puede proporcionar un perfeccionamiento en la coincidencia. Hay dos métodos para hallar la mejor coincidencia de palabra de código y exponente. En el primer método, se halla una mejor palabra de código típicamente usando la distancia euclídea como la métrica (MSE). Después de que se halla la mejor palabra de código, se halla el mejor exponente. El mejor exponente se halla usando uno de los siguientes dos métodos.

20 Un método es intentar todos los exponentes disponibles y observar cuál proporciona la distancia Euclídea mínima, el otro método es probar exponentes y observar cuál exponente proporciona la mejor coincidencia de histograma o función de masa de probabilidad (pmf). La coincidencia de pmf puede calcularse usando el segundo momento sobre la media (la varianza) para la pmf del vector original y para cada uno de los vectores exponentiados. El que tiene la coincidencia más cercana se elige para que sea el mejor exponente.

25 El segundo método de hallar la mejor coincidencia de palabra de código y exponente es hacer una búsqueda exhaustiva usando muchas combinaciones de palabras de código y exponentes.

30 Si, por ejemplo,  $X^{0.5}$  proporciona una mejor comparación que  $X^{1.0}$ , se codifica una subbanda usando el desplazamiento para esa palabra de código en la banda base (712), junto con una transformación (lineal o no lineal)  $x^p$ , donde uno o más bits que indican  $p=0,5$  se envían y se aplican en el decodificador. En este ejemplo, la búsqueda continúa hallando una palabra de código en primer lugar, y a continuación variando con una transformación, pero no se requiere tal orden en la práctica.

35 En otro ejemplo, se realiza una búsqueda exhaustiva a lo largo de la banda base y/o otros libros de código para hallar una mejor coincidencia. Por ejemplo, se realiza una búsqueda que comprende una búsqueda exhaustiva a lo largo de la banda base de todas las combinaciones de (transformación exponencial ( $p=0,5, 1,0, 2,0$ ), transformación de signo (+/-), dirección (hacia delante/inversa). De manera similar, esta búsqueda exhaustiva puede realizarse a lo largo del espectro de libro de códigos de ruido, o palabras de código.

En general, puede proporcionarse una coincidencia cercana determinando una varianza más baja entre la subbanda que se está codificando y la palabra de código y transformación seleccionada para modelar una subbanda. Un identificador o indicación codificada de la palabra de código y/o transformación, junto con otra información tal como un factor de escala, se codifica en el flujo de bits y se proporciona al codificador.

#### 40 Codificación de palabra de código múltiple ejemplar

45 En un ejemplo, se utilizan dos palabras de código diferentes para proporcionar una codificación de subbanda. Por ejemplo, dadas dos palabras de código  $b$  y  $n$  de longitud  $u$ , se proporcionan  $b = \langle b_0, b_1, \dots, b_u \rangle$  y  $n = \langle n_0, n_1, \dots, n_u \rangle$  para describir mejor una subbanda que se está codificando. El vector  $b$  puede provenir de la banda base, cualquier banda anterior, un libro de códigos de ruido o una biblioteca, y el vector  $n$  puede provenir de manera similar desde cualquier tal fuente. Se proporciona una regla para intercalar coeficientes desde cada una de dos o más palabras de código  $b$  y  $n$ , de manera que el decodificador conoce implícita o explícitamente qué coeficiente tomar desde las palabras de código  $b$  y  $n$ . La regla puede proporcionarse en el flujo de bits o puede conocerse por el decodificador implícitamente.

50 La regla y dos o más vectores se usan en el decodificador para crear las subbandas  $s = \langle n_0, b_1, n_2, b_3, b_4, \dots, n_u \rangle$ . Por ejemplo, se establece una regla basándose en el orden de las palabras de códigos enviadas, y un valor de porcentaje "a". El codificador entrega información en el orden  $(b, n, a)$ . El decodificador traduce la información en un requisito para tomar cualquier coeficiente desde el primer vector  $b$  si ese coeficiente es menor que 'a' multiplicado por el valor de coeficiente más alto  $M$  en el vector  $b$ . Por lo tanto, si un coeficiente  $b_1$  es mayor que  $a \cdot M$ , entonces  $b_1$  está en el vector  $s$ , de otra manera  $n_1$  está en  $s$ . Otra regla puede requerir que para que  $b_1$  esté en el vector  $s$ , tiene que ser parte de un grupo de  $T$  coeficientes adyacentes con un valor menor que  $a \cdot M$ . Si se establece un valor por defecto para 'a', entonces 'a' no necesita enviarse al decodificador, puesto que está implícito.

Por lo tanto, un decodificador puede enviar dos o más identificadores de palabra de código, y opcionalmente, una regla para decodificar cuáles coeficientes tomar para crear la subbanda. El codificador enviará también información de factor de escala para palabras de código, y opcionalmente, si fuera relevante, cualquier otra información de transformación de palabra de código puesto que b y/o n pueden transformarse lineal o no linealmente.

- 5 Usando dos o más palabras de código b y n anteriores, un codificador enviaría un identificador (por ejemplo, un vector de movimiento, índice de libro de códigos, etc.) de las palabras de códigos, una regla (por ejemplo, índice para libro de reglas) o la regla será conocida implícitamente tanto por el codificador como el decodificador, cualquier información de transformación adicional (por ejemplo,  $x^p$ ,  $p=0,5$ , suponiendo b o n también requiere transformación adicional), y la información acerca de factores de escala (por ejemplo,  $s_b$ ,  $s_n$ , etc.). La información de factor de escala puede ser también un factor de escala y una relación (por ejemplo,  $s_b$ ,  $s_b/s_n$ , etc.). Con un factor de escala de vector y una relación, el decodificador tendrá suficiente información para calcular el otro factor de escala.

Mejora de banda base ejemplar

- 15 Bajo ciertas condiciones, tales como aplicaciones de baja tasa de bits, la misma banda base puede no codificarse bien (por ejemplo, varios coeficientes cero consecutivos o entremezclados). En un ejemplo de este tipo, la banda base representa picos de buena intensidad, pero no representa bien varianzas sutiles a coeficientes que representan intensidades inferiores entre picos. En un caso de este tipo, los picos de una palabra de código desde la misma la banda base se seleccionan como un primer vector (por ejemplo, b), y los coeficientes cero, o coeficientes relativos muy bajos se sustituyen por un segundo vector (por ejemplo, n) que se asemeja más estrechamente a la baja energía entre picos. Por lo tanto, el método de dos palabras de código puede usarse en la banda base o subbanda de la banda base, para proporcionar mejora de banda base. Como anteriormente, la regla usada para seleccionar el primer o segundo vector, puede ser explícita y enviarse al decodificador, o ser implícita. En algunos casos el segundo vector puede proporcionarse mejor mediante una palabra de código de ruido.

Transformaciones ejemplares

- 25 Una banda base, banda previa y otro libro de códigos proporcionan una biblioteca de coeficientes consecutivos, sirviendo cada coeficiente potencialmente como el primer coeficiente en una serie de coeficientes consecutivos que puede servir como una palabra de código. Una palabra de código de mejor coincidencia en la biblioteca se identifica y envía a un decodificador, junto con un factor de escala, y se usa mediante el decodificador para crear una subbanda en la subbanda extendida.

- 30 Opcionalmente, una o más palabras de código en la biblioteca se transforman para proporcionar un universo mayor de palabras de código disponibles para hallar una mejor coincidencia para una forma que se está codificando. En matemáticas, existe un universo de transformaciones lineales y no lineales para formas, vectores y matrices. Por ejemplo, un vector puede invertirse, negarse a través de un eje, y una forma puede modificarse de otra manera con transformaciones lineales y no lineales tal como aplicando funciones de raíz, exponentes, etc. Se realiza una búsqueda en la biblioteca de palabras de código, que incluye aplicar una o más transformaciones lineales o no lineales en las palabras de código, y se identifica una palabra de código de coincidencia más cercana, junto con cualquier transformación. Un identificador de una mejor coincidencia, palabra de código, un factor de escala y un identificador de transformación se envían a un decodificador. Un decodificador recibe la información y reconstruye una subbanda en la banda extendida.

- 40 Opcionalmente, un codificador selecciona dos o más palabras de código que juntas representan mejor una subbanda que se está codificando y/o mejorando. Se usa una regla para seleccionar o intercalar posiciones de coeficiente individuales en la subbanda que se está codificando. La regla es implícita o explícita. La subbanda que se está codificando puede estar en la banda extendida, o puede ser una subbanda en la banda base que se está mejorando. Las dos o más palabras de código que se están usando pueden provenir de una banda base o cualquier otro libro de códigos, y una o más de las palabras de códigos puede transferirse lineal o no linealmente.

- 45 Coincidencia de envolvente ejemplar

Una señal denominada "una envolvente" (por ejemplo,  $Env(i)$ ) se genera ejecutando una media ponderada en la señal de entrada  $x(i)$  (por ejemplo, audio, vídeo, etc.) como sigue:

$$Env(i) = \sum_{j=-L}^L w(j) | x(i+j) |$$

- 50 donde  $w(j)$  es una función de ponderación (actualmente una forma de triángulo) y L es el número de coeficientes vecinos a reconstruirse en el análisis ponderado. Anteriormente, se analizó el ejemplo de una búsqueda exhaustiva usando un universo de palabras de código de entrada, transformación de exponente (0,5, 1,0, 2,0), negación de

5 coeficiente (signo +/-) y dirección de coeficiente de palabra de código (hacia delante, inversa). En lugar de un mejor 'Q' número de palabras de código se seleccionan en primer lugar (combinaciones de palabra de código, exponente, signo y/o dirección) usando una distancia euclídea entre las envolventes de la subbanda que se está codificando y la palabra de código. Las versiones no cuantificadas originales de las palabras de códigos pueden ser útiles para medir la distancia euclídea de la envolvente. A partir de estos Q candidatos más cercanos determinados basándose en la distancia euclídea, se selecciona una mejor coincidencia. Opcionalmente, después de que se consideren las envolventes, un método (tal como como se ha descrito anteriormente métodos de comparación de palabras de códigos) puede devolver para examinar cuál de los Q candidatos se adapta mejor.

Modificación de palabra de código ejemplar

10 Dado un libro de códigos que consiste en vectores de código, se propone una modificación de los vectores de código en el libro de códigos de manera que pueden representar mejor el vector que se está codificando. La modificación de libro de código/palabra de código puede consistir en cualquier combinación de una o más de las siguientes transformaciones.

- Transformación lineal aplicada a un vector de código.
- 15 • Transformación no lineal aplicada a un vector de código.
- Combinar más de un vector de código para obtener un nuevo vector de código (los vectores que se combinan pueden provenir desde el mismo libro de códigos, diferentes libros de códigos o ser aleatorios).
- Combinar un vector de código con una codificación de base.

20 Se usa la información relacionada con la transformación, si la hubiera, y cuáles vectores de código se usan en la transformación se envían al decodificador en el flujo de bits o se calculan en el decodificador usando el conocimiento que ya tenía (datos que ya ha decodificado). Un vector es típicamente una cierta banda de coeficientes espectrales que sean de codificar.

Se proporcionan tres ejemplos en particular para modificaciones de palabra de código:

- (1) exponenciación aplicada a cada componente del vector (transformación no lineal),
- 25 (2) combinación de dos (o más) vectores para formar un nuevo vector, donde cada uno de los dos vectores se usa para representar porciones del vector que tienen diferentes características, y (3) combinar un vector de código con una codificación de base. En el siguiente análisis, se usará  $v$  para representar el vector a codificar,  $x$  será el vector de código o palabra de código que se está usando para codificar  $v$ , e  $y$  serán el vector de código modificado. El vector  $v$  se codificará usando una aproximación  $v' = Sx$ , donde  $S$  es un factor de escala. El factor de escala usado es
- 30 una versión cuantificada de la relación de potencia entre  $v$  y  $x$ ,

$$S = \frac{Q(\|v\|)}{\|x\|},$$

donde  $Q(.)$  es cuantificación, e  $\|.\|$  representa la norma, que es la potencia en el vector. Se envía una versión cuantificada de la potencia en el vector original. El decodificador calcula el factor de escala para usar dividiendo la potencia en el vector de código.

35 Transformación no lineal ejemplar

Un primer ejemplo consiste en aplicar un exponente a cada componente en el vector de código. La Tabla 3 proporciona una transformación no lineal de una serie de coeficientes en una palabra de código.

Tabla 3								
Palabra de código	1	2	3	2	1	1	2	3
Transformación	1	4	9	4	1	1	4	9

40 En este ejemplo, cada coeficiente en una palabra de código (vector de código) se eleva a la potencia de un exponente dos ( $x^2$ ). En un ejemplo de este tipo, si la forma de la palabra de código transformada es un mejor ajuste



para un vector a codificar, entonces el codificador proporcionará una identificación de la palabra de código y la transformación que conduce a una mejor coincidencia.

5 El exponente puede enviarse al decodificador usando un número fijo de bits, o puede enviarse desde un libro de códigos de exponentes, o puede calcularse implícitamente en el decodificador usando datos previamente observados. Por ejemplo, para un vector L dimensional, sean los componentes del vector de código de orden 'i' en un libro de códigos  $x_i[0], x_i[1], \dots, x_i[L-1]$ . A continuación, la exponenciación aplica un exponente 'p' para modificar el vector para obtener un nuevo vector  $y_i$ ,

$$y_i[j] = (x_i[j])^p, \text{ para } j = 0, 1, \dots, L-1,$$

10 donde 'j' es el índice de componente. Esta transformación no lineal permite que un vector de código que tiene picos se use para codificar un vector que no tiene usando un valor de p que es menor de 1. De manera similar, permite que se use un vector de código no de picos para representar uno con picos usando  $p > 1$ .

La Figura 9 es un gráfico de un vector ejemplar que no representa picos de manera distinta.

La Figura 10 es un gráfico de la Figura 9 con distintos picos creado mediante transformación exponencial.

15 Como un ejemplo, véase la Figura 9 y Figura 10. En la Figura 9, un vector que es bastante aleatorio y se muestra que no tiene picos distintos. Cuando se aplica un exponente  $p=5$ , entonces la Figura 10 representa mejor los picos deseados. De manera similar, si el vector de código original que se mostró en la Figura 10, entonces un exponente  $p=1/5=0,2$ , proporcionaría la Figura 9. El factor de escala del curso se recalcula puesto que la norma (o energía) en el vector de código ha cambiado durante la transformación de x a y. En particular,  $S=Q(\|y\|)/\|x\|$  se usa ahora por el factor de escala. El factor de escala real que se envía  $Q(\|v\|)$  no cambia con el exponente, pero el decodificador  
20 tiene que calcular un factor de escala diferente debido al cambio en la potencia en el vector de código.

Una palabra de código puede tener varios exponentes aplicados en ella, proporcionando cada uno diferentes resultados. El método usado para calcular el mejor exponente es hallar un exponente de manera que el histograma (o función de masa de probabilidad (pmf)) de los valores a través del vector de código coincidan mejor el del vector real. Para hacer esto, se calcula una varianza de los valores de símbolo tanto para el vector como el vector de  
25 código usando exponenciación. Por ejemplo, supóngase que el conjunto de posibles exponentes es  $p_k$ , donde k se usa para indexar el conjunto de posibles exponentes,  $k=0, 1, \dots, P-1$ . Entonces se calcula el segundo momento normalizado sobre de la media para el vector de código resultante de cada posible exponente ( $V_k$ ), y se compara al vector real (V).

$$V_k = \frac{\left( \frac{1}{L} \sum_{j=0}^{L-1} |x[j]|^{2p_k} - \left( \frac{1}{L} \sum_{j=0}^{L-1} |x[j]|^{p_k} \right)^2 \right)}{\frac{1}{L} \sum_{j=0}^{L-1} |x[j]|^{2p_k}}, k = 0, 1, \dots, P-1$$

$$V = \frac{\left( \frac{1}{L} \sum_{j=0}^{L-1} |v[j]|^2 - \left( \frac{1}{L} \sum_{j=0}^{L-1} |v[j]| \right)^2 \right)}{\frac{1}{L} \sum_{j=0}^{L-1} |v[j]|^2}$$

30 Se elige el mejor exponente para minimizar la diferencia entre  $V_k$  y V, y se proporciona mediante  $p_b$ , donde b se define como:

$$b = \arg \min_k (|V - V_k|)$$

Como se ha indicado previamente, un exponente de mejor coincidencia puede hallarse también usando una búsqueda exhaustiva.

Modificación de palabra de código ejemplar mediante combinación

5 Otra transformación combina múltiples vectores para formar un nuevo vector de código. Esto es esencialmente una codificación de múltiples etapas, donde en cada etapa se halla una coincidencia que coincide mejor con la porción más importante del vector no codificado aún. Como un ejemplo para dos vectores, hallamos en primer lugar la mejor coincidencia y a continuación observamos cuál porción del vector se está codificando bien. La segmentación puede enviarse explícitamente, pero esta puede tomar demasiados bits. Por lo tanto, la segmentación se proporciona implícitamente, en un ejemplo, indicando qué porción del vector usar. La porción restante se representa a continuación usando cualquiera de un vector de código aleatorio, u otro vector de código desde un libro de códigos que representa los componentes restantes mejor. Siendo  $x$  un primer vector de código, y siendo  $w$  un segundo vector de código. Especificando el conjunto  $T$  la porción del vector que se considera para codificarse usando el primer vector de código. La cardinalidad del ajuste  $T$  será entre 0 y  $L$ , es decir tendrá entre 0 y  $L$  elementos que representan los índices del vector que se consideran para codificarse usando este primer vector de código. Se proporciona una regla para figurar cuáles componentes se representan bien mediante el primer vector y la regla puede usar métricas, tales como, determinar si un coeficiente potencial es mayor que un cierto porcentaje del coeficiente máximo en el primer vector. Por lo tanto, para cualquier coeficiente en el primer vector que está dentro de un porcentaje del coeficiente más alto en el primer vector, ese coeficiente se tomará desde el primer vector, sino, ese coeficiente de palabra de código se toma desde la segunda palabra de código. Siendo  $M$  el valor máximo en el primer vector de código  $x$ . Entonces el conjunto  $T$  puede definirse usando lo siguiente:

$$T = \{j : x[j] > aM, j = 0, 1, \dots, L - 1\},$$

donde 'a' es alguna constante entre 0 y 1. Por ejemplo, si  $a=0$ , entonces cualquier valor distinto de 0 se considera que pertenece al conjunto  $T$  de vectores codificados. Si  $a=1-\epsilon$ , entonces únicamente se considera el mismo valor máximo para codificarse, si  $\epsilon$  se toma para que sea suficientemente pequeño. Entonces dado el conjunto  $T$ , un conjunto  $N$  es el complementario y el conjunto restante tomado desde el vector  $w$ , como sigue:

$$N = \{j : x[j] \leq aM, j = 0, 1, \dots, L - 1\}.$$

Por lo tanto, se toma un coeficiente  $x[j]$  desde  $x$  o  $w$  dependiendo del valor de  $aM$ . Obsérvese que  $N$  o  $T$  pueden dividirse adicionalmente usando otras reglas similares para obtener más de dos vectores. Dados  $T$  y  $N$  como los conjuntos de índices codificados usando el primer vector de código ( $x$ ) y el segundo vector de código ( $w$ ) respectivamente, se define un nuevo vector:

$$y[j] = \begin{cases} S_x x[j], & \text{if } j \in T \\ S_w w[j], & \text{if } j \in N \end{cases},$$

donde  $S_x$  y  $S_w$  son los factores de escala para  $x$  y  $w$ , respectivamente. Puesto que se envía típicamente un factor de escala para todo el vector de código, que representa una versión cuantificada de la potencia en todo el vector que se está codificando, una relación entre los dos factores de escala ( $S_w/S_x$ ) además del factor de escala para todo el vector de código necesita enviarse en este caso. En general, si se crea un vector usando 'm' vectores de código, entonces tendrían que enviarse 'm' factores de escala que incluyen el del vector entero. Por ejemplo, para el caso de dos vectores, obsérvese que,

$$\|v\|^2 = \frac{1}{L} \sum_{j=0}^{L-1} v^2[j] = \frac{1}{L} \sum_{j \in T} v^2[j] + \frac{1}{L} \sum_{j \in N} v^2[j].$$

Suponiendo  $v_t$  y  $v_n$  se definen como los dos vectores, entonces su potencia puede definirse como,

$$\|v_t\|^2 = \frac{1}{|T|} \sum_{j \in T} v^2[j] \quad \|v_n\|^2 = \frac{1}{|N|} \sum_{j \in N} v^2[j],$$

donde  $|T|$  y  $|N|$  son la cardinalidad de los dos conjuntos (el número de elementos). Dados los valores para  $\|v\|$  (la potencia total en el vector), y  $\|v_n\|$  (la potencia en el segundo componente del vector), un decodificador puede calcular,

$$\|v_t\|^2 = \frac{L\|v\|^2 - |N|\|v_n\|^2}{|T|}$$

5 Por lo tanto, si se envía una versión cuantificada de la potencia en el conjunto N ( $Q(\|v_n\|)$ ), y se envía la potencia total  $Q(\|v\|)$ , es suficiente información para el decodificador.

Es importante observar que, usando el mismo vector de código  $x$  para realizar la segmentación, el codificador evita tener que enviar ninguna información relacionada con la segmentación puesto que el coeficiente seleccionado desde cada vector  $x$  y  $w$  está implícito en las reglas (por ejemplo,  $x[j] \geq aM$ ). Incluso en casos cuando el índice del vector de código o vector de movimiento que corresponde a  $x$  no se envía (es un vector de código aleatorio), la segmentación de los conjuntos  $T$  y  $N$  puede coincidir entre codificador y decodificador usando un vector aleatorio siendo el estado del generador de vector aleatorio determinístico basándose en información que tienen tanto el codificador como el decodificador. Por ejemplo, el vector aleatorio puede determinarse usando alguna combinación de los bits menos significativos (LSB) de datos que se han codificado y enviarse al decodificador (tal como en la banda base codificada) y a continuación usar esos para sembrar un generador de números pseudoaleatorios. De esta manera la segmentación puede controlarse implícitamente incluso si no se envía el vector de código real.

Esta transformación combinando dos vectores permite mejor representación del vector que se ha de codificar. El vector  $w$  puede provenir de un libro de códigos y puede enviarse un índice para representarlo, o puede ser aleatorio, caso en el que no es necesario que se envíe información adicional. Obsérvese que en el ejemplo dado anteriormente, la segmentación es implícita puesto que se hace usando una regla de comparación en los coeficientes (por ejemplo,  $x[j] \geq aM$ ) usando el vector  $x$ , por lo que no necesita enviarse información con respecto a la segmentación. Esta transformación es útil cuando el vector a codificarse tiene dos distribuciones diferentes.

La Figura 11 es un gráfico de una palabra de código según se compara a la banda que está modelando. En este ejemplo (1100), el vector de código se ha elegido para coincidir mejor los picos en el vector. Sin embargo, aunque los picos están bien coincidos, el resto del vector no tiene potencia similar. La porción restante del vector de código tiene mucho menos potencia con relación a los picos que lo que tiene el vector real. Esto da como resultado artefactos de compresión perceptibles. Sin embargo, cuando se selecciona la porción de  $v$  que se codificó bien mediante el vector de código del primer vector y a continuación se aplica un segundo vector de código a la porción restante, se obtiene un resultado mucho mejor.

30 La Figura 12 es un gráfico de una palabra de código transformada según se compara a la subbanda que está modelando. La subbanda modelada se modela mediante una palabra de código creada desde dos palabras de código.

La Figura 13 es un gráfico de una palabra de código, una subbanda a codificarse mediante la palabra de código, una versión cambiada de escala de la palabra de código, y una versión modificada de la palabra de código.

35 **Modificación de palabra de código ejemplar mediante operaciones selectivas**

Una versión alternativa de los vectores de códigos múltiple (por ejemplo, múltiples palabras de código) añade el primer vector de código en lugar de sustituirlo para ciertos coeficientes seleccionados. Esto puede hacerse aplicando la siguiente ecuación:

$$y[j] = \begin{cases} S_x x[j], & \text{if } j \in T \\ S_w w[j] + S_x x[j], & \text{if } j \in N \end{cases}$$

40 **Mejora ejemplar de la banda base**

En este ejemplo, un vector de código se combina con una codificación de base. Esto es similar al enfoque de dos vectores (o múltiples vectores), excepto que el primer vector  $x$  es tanto el vector que se está codificando y se usa el mismo como uno de los dos vectores para codificarse a sí mismo. Por ejemplo, una codificación de base se modifica para incluir aquellos coeficientes donde la codificación de base está funcionando bien y se toman mejores coeficientes desde el segundo vector, como anteriormente. Para cada vector (subbanda) que se codifique, si ya

existe una codificación de base, esta codificación de base entonces es el primer vector de código en el esquema de vector múltiple, donde se segmenta en regiones T y N (o más regiones). La segmentación (por ejemplo, selección de coeficiente) puede proporcionarse usando las mismas técnicas que en el enfoque de vector de código múltiple.

5 Por ejemplo, para cada codificación de base, si hay algún coeficiente con un valor de 0, todos estos irán entonces al conjunto N que se codifican a continuación mediante una capa de mejora (por ejemplo, segundo vector). Un método de este tipo puede usarse para rellenar grandes huecos espectrales que resultan a menudo de codificar a tasas de bits muy bajas. Las modificaciones pueden incluir no rellenar huecos o coeficientes 'cero' a menos que no sean mayores que algún umbral, cuando pueda definirse el umbral para que sea un cierto número de Hercios (Hz) o coeficientes (coeficientes múltiples de cero). Puede haber también limitaciones sobre no rellenar huecos que están por debajo de una cierta frecuencia. Estas limitaciones modifican las reglas de segmentación implícitas proporcionadas anteriormente (por ejemplo,  $x[j] \geq aM$ , etc.). Por ejemplo, si se proporciona un umbral 'T' en un tamaño mínimo de un hueco espectral, entonces este esencialmente cambia la definición del conjunto N a lo siguiente:

$$N = \{j : x[j - K] \leq aM \ \& \ x[j - K + 1] \leq aM \ \& \ \dots \ \& \ x[j - K + T - 1] \leq aM, \\ j = 0, 1, \dots, L - 1\}$$

15 para algún K entre 0,...,T-1. Por lo que para que  $x[j]$  esté en el conjunto N, tiene que ser parte de un grupo de T coeficientes consecutivos, todos los cuales tienen un valor menor que o igual a (aM). Este puede calcularse en dos etapas, calcular en primer lugar para cada coeficiente si su valor es menor que el umbral, y a continuación agruparlos juntos para observar si cumplen los requisitos 'consecutivos'. Para un hueco espectral verdadero de tamaño T, a=0. Otras condiciones tales como restricciones de frecuencia mínima se añaden a la restricción adicional para que pertenezcan al conjunto N,  $j > T_{\text{minfreq}}$ .

20 Las reglas anteriores proporcionan un filtro que requiere que múltiples coeficientes en una fila (por ejemplo, T coeficientes consecutivos) satisfagan la condición  $x[j] \leq aM$ , antes de que la regla señale sustituir los coeficientes por valores desde el segundo vector.

25 Otra modificación que puede necesitar realizarse es debido al hecho de que la codificación de base también codifica los canales después de aplicar una transformación de canal. Por lo tanto, después de una transformación de canal la codificación de base y codificación de mejora pueden tener diferentes agrupaciones de canal. Por lo que, en lugar de simplemente mirar en la codificación de base para el canal particular después de que se aplican las mejoras, la segmentación puede mirar en más del canal de codificación base. Esto de nuevo modifica la restricción de segmentación. Por ejemplo, supóngase que los canales 0 y 1 están codificados conjuntamente. Entonces la regla para aplicar la mejora se cambia a lo siguiente. Para aplicar la mejora, el hueco espectral tiene que estar presente en ambos canales de banda base codificados puesto que ambos canales codificados contribuyen a ambos canales reales.

#### Optimización de segmentación de subbandas ejemplar

35 La buena segmentación de frecuencia es importante para la calidad de codificar datos espectrales. La segmentación implica descomponer los datos espectrales en unidades denominadas subbandas o vectores. Una segmentación sencilla es dividir uniformemente el espectro en un número deseado de segmentos homogéneos o subbandas. La segmentación homogénea puede ser subóptima. Puede haber regiones del espectro que puedan representarse con tamaños más grandes de subbanda, y otras regiones se representan mejor con tamaños de subbanda más pequeños. Se describen diversas características para proporcionar intensidad de datos espectrales dependiendo de la segmentación. Se proporciona segmentación más precisa para regiones de varianza espectral mayor y se proporciona segmentación más basta para regiones más homogéneas. Por ejemplo, inicialmente se proporciona una segmentación por defecto o inicial, y una configuración de optimización o posterior varía la segmentación basándose en una varianza de intensidad de datos espectrales.

#### Segmentación por defecto ejemplar

45 Los datos espectrales se segmentan inicialmente en subbandas. Opcionalmente, una segmentación inicial puede variarse para producir una segmentación óptima o posterior. Dos segmentaciones iniciales o por defecto de este tipo se denominan una segmentación de división uniforme y una configuración de división no uniforme. Estas u otras configuraciones de subbanda pueden proporcionarse inicialmente o por defecto. Opcionalmente, la configuración inicial o por defecto puede reconfigurarse para proporcionar una configuración de subbanda posterior.

50 Dados los datos espectrales de L coeficientes espectrales, una segmentación de división uniforme de M subbandas de datos se identifica con la siguiente ecuación:

$$s[j] = \text{round}\left(\frac{jL}{M}\right), j = 0, 1, \dots, M-1, M$$

Por ejemplo, si los L coeficientes espectrales se etiquetan como puntos como 0, 1, ..., L-1, entonces las M subbandas se inician en los  $s[j]$  coeficientes en los datos espectrales. Por lo tanto, la subbanda de orden 'j' tiene coeficientes de  $s[j]$  a  $s[j+1]-1$ ,  $j=0, 1, \dots, M-1$ , con un tamaño de subbanda de  $s[j+1]-s[j]$  coeficientes.

- 5 La segmentación de división no uniforme se hace de una manera similar, excepto que se proporcionan multiplicadores de subbanda. Un multiplicador de subbanda se define para cada una de las M subbandas,  $a[j]$ ,  $j=0, 1, \dots, M-1$ . Además, se proporciona un multiplicador acumulativo como sigue:

$$b[j] = \sum_{k=0}^{j-1} a[k], j = 0, 1, \dots, M$$

El punto de inicio para las subbandas en la configuración del caso de división no uniforme se define como:

10 
$$s[j] = \text{round}\left(\frac{b[j]L}{b[M]}\right), j = 0, 1, \dots, M-1, M$$

De nuevo, la subbanda de orden 'j' incluye los coeficientes de  $s[j]$  a  $s[j+1]-1$ , donde  $j = 0, 1, \dots, M-1$ , con un tamaño de subbanda de  $s[j+1] - s[j]$  coeficientes. La configuración no uniforme tiene tamaños de subbanda que aumentan con la frecuencia, pero puede ser cualquier configuración. Además, si es deseable, puede predeterminarse, de modo que no necesita enviarse información adicional para describirla. Para el caso no uniforme por defecto, se proporciona un ejemplo de multiplicadores de subbanda como sigue:

15 
$$a = \{1, 1, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, \dots\}$$

Por lo tanto, el multiplicador de tamaño de banda no uniforme por defecto es una configuración de división donde los tamaños de banda son monótonicamente no decrecientes (las primeras pocas subbandas son más pequeñas, y las subbandas de frecuencia más alta son más grandes). Las subbandas de frecuencia más alta a menudo tienen menos variación para empezar, por lo que menos subbandas más grandes pueden capturar la escala y forma de la banda. Adicionalmente, las subbandas de frecuencia más alta tienen menos importancia en la distorsión perceptual global puesto que tienen menos energía y son perceptualmente menos importantes para el oído humano. Obsérvese que la división uniforme puede explicarse también usando multiplicadores de subbanda, excepto que  $a[j] = 1$  para toda j.

- 25 Aunque una segmentación por defecto o inicial a menudo es suficiente para codificar datos espectrales, y de hecho el esquema no uniforme puede manejar un gran porcentaje de casos, hay señales que se benefician de una segmentación optimizada. Para tales señales, se define una segmentación que es similar al caso no uniforme, excepto que los multiplicadores de banda son en su lugar arbitrarios o fijos. Los multiplicadores de banda arbitrarios reflejan las divisiones y uniones de las subbandas. En un ejemplo, un codificador señala al decodificador con un primer bit que indica si la segmentación está fijada (por ejemplo, por defecto) o variable (por ejemplo, optimizada o modificada). Se proporciona un segundo bit para señalar si la segmentación inicial es división uniforme o una división no uniforme.

Segmentación optimizada ejemplar

35 Empezando con una segmentación por defecto (tal como una segmentación uniforme o no uniforme), las subbandas se dividen o unen para obtener una segmentación optimizada o uniforme. Se realiza una decisión para dividir una subbanda en dos subbandas, o para unir dos subbandas en una subbanda. Una decisión para dividir o unir puede basarse en diversas características de los datos espectrales dentro de una subbanda inicial, tal como una medición de intensidad de cambio a través de una subbanda. En un ejemplo, se realiza una decisión para dividir o unir basándose en características de datos espectrales de subbanda tales como tonalidad u horizontalidad espectral en una subbanda.

40 En un ejemplo de este tipo, si la relación de energía es similar entre dos subbandas, y al menos una de las bandas no es tonal, entonces las dos subbandas adyacentes se unen. Esto es debido a que un vector de forma única (por ejemplo, palabra de código) y un factor de escala probablemente serán suficientes para representar las dos subbandas. Un ejemplo de una relación de energía de este tipo se proporciona como sigue:

$$\frac{\min(E_0, E_1)}{\max(E_0, E_1)} \geq (1 - a) \ \&\& \ (\text{Tonalidad}_0 < T \ \|\ \text{Tonalidad}_1 < T)$$

En este ejemplo,  $E_0$  es la energía en la subbanda 0,  $E_1$  es la energía en una subbanda adyacente 1, 'a' es un valor umbral constante (típicamente en el intervalo  $0 < a < 1$ ) y T es una métrica de comparación de tonalidad. La medida de tonalidad (por ejemplo, Tonalidad<sub>0</sub>) en una subbanda puede obtenerse usando diversos métodos analizando el espectro.

De manera similar, si dividir una única subbanda en dos subbandas crea dos subbandas con energía distinta, entonces debería realizarse la división. O, si dividir una subbanda crea dos subbandas que son intensamente tonales con diferentes características de forma, entonces la subbanda debería dividirse. Por ejemplo, una condición de este tipo se define como sigue:

$$\frac{\max(E_0, E_1)}{\min(E_0, E_1)} \geq (1 + b) \ \|\ (\text{Tonalidad}_0 > T \ \&\& \ \text{Tonalidad}_1 > T \ \&\& \ \text{forma diferente})$$

donde 'b' es una constante mayor que cero. Por ejemplo, pueden definirse dos subbandas para tener diferente forma si la coincidencia de forma mejora significativamente cuando se divide la subbanda. En un ejemplo, una coincidencia de forma se considera mejor si las dos subbandas de división tienen una coincidencia de diferencia euclídea de media mínima cuadrática (MSE) mucho inferior después de la división, en comparación con la coincidencia antes de la división. Por ejemplo, una subbanda se compara a una diversidad de palabras de código para determinar una palabra de código de mejor coincidencia para la subbanda única. Entonces la subbanda se divide en dos bandas, cada subbanda comparada a (la mitad) palabras de código para hallar una mejor coincidencia para cada subbanda de la división. La MSE de las dos coincidencias de subbandas se compara a la MSE de la coincidencia de subbanda única, y una coincidencia mejorada significativamente indica un valor de mejora de la tara extra de codificar una división. Por ejemplo, si una MSE mejora en el 20 % o más, la división se considera eficaz. En este ejemplo, aunque no se requiere, la coincidencia de forma se hace relevante si ambas subbandas de división son tonales.

En un ejemplo, se ejecuta un algoritmo repetitivamente hasta que no se dividen o unen subbandas adicionales en una iteración presente. Puede ser beneficioso etiquetar subbandas como división, unión o en orden para reducir la posibilidad de un bucle infinito. Por ejemplo, si una subbanda está marcada como una subbanda de división, entonces no se unirá de vuelta con una subbanda desde la que se dividió. Un bloque que se marca como unido, no se dividirá en la misma configuración.

Se utilizan diversas métricas para calcular tonalidad, energía o forma diferente. Un vector de movimiento y una métrica de escala pueden usarse para codificar una subbanda extendida. Si dividir una subbanda en dos subbandas crea una energía significativamente diferente en el factor de escala (por ejemplo,  $\geq (1 + b)$ , donde b es 0,2 - 0,5), entonces la subbanda puede dividirse. En un ejemplo, se calcula la tonalidad en el dominio de la transformada rápida de Fourier (FFT). Por ejemplo, una señal de entrada se divide en bloques fijos de 256 muestras, y se ejecuta la FFT en tres bloques de FFT adyacentes. Se realiza una media de tiempo en las tres salidas de FFT adyacentes para obtener una salida de FFT promediada en tiempo para el bloque actual. Se ejecuta un filtro de mediana sobre las tres salidas de FFT promediadas en tiempo para obtener una línea de base. Si un coeficiente está por encima de un cierto umbral por encima de la línea de base, entonces el coeficiente se clasifica como tonal, y el porcentaje que está por encima de la línea de base es una medida de la tonalidad. Si el coeficiente está por debajo del umbral, entonces no es tonal y la medida de la tonalidad es 0. La tonalidad para una pieza de frecuencia de tiempo particular se halla mapeando las dimensiones de la pieza a los bloques de FFT y acumulando la medida de tonalidad a través del bloque. El umbral que un coeficiente tiene que estar por encima de la línea de base puede definirse para que sea un umbral absoluto, una relación con relación a la línea de base, o una relación con relación a la varianza de la línea de base. Por ejemplo, si el coeficiente está por encima de una desviación típica local desde la línea de base (mediana filtrada, tiempo promediado), puede clasificarse como que es tonal. En un caso de este tipo, la subbanda traducida correspondiente en la MLT que representa los bloques de FFT tonales se etiqueta como tonal, y puede dividirse. El análisis está relacionado con la magnitud de la FFT en oposición a la fase. Con respecto a la métrica de MSE en diferentes formas, una métrica de MSE mucho inferior puede variar sustancialmente la tasa de bits. Por ejemplo, con tasas de bits más altas, si la MSE se reduce en aproximadamente el 20 %, entonces una determinación de división puede tener sentido. Sin embargo, a tasas de bits más bajas la decisión de división puede tener lugar a una MSE el 50 % más baja.

Multiplicador de banda variable y codificación ejemplar

Después de que las bandas se dividen y/o unen, se calcula la relación entre el tamaño de subbanda más pequeño original y el nuevo tamaño de subbanda más pequeño. Se define una relación como  $\text{minRatioBandSize} = \max(1, \text{tamaño de subbanda más pequeño original} / \text{nuevo tamaño de subbanda más pequeño})$ . A continuación, la

subbanda optimizada con el tamaño más pequeño (por ejemplo, número de coeficientes en la subbanda) se asigna un multiplicador de sub-banda de 1, y los otros tamaños de subbanda tienen un multiplicador de banda establecido como la función  $\text{round}(\text{este tamaño de subbanda} / \text{tamaño de subbanda más pequeño})$ . Por lo tanto, los multiplicadores de sub-banda son enteros mayores que o iguales a 1, y `minRatioBandSize` es también un entero mayor que o igual a 1. Los multiplicadores de sub-banda se codifican esencialmente codificando una diferencia entre el multiplicador de sub-banda esperado y el multiplicador de sub-banda optimizado usando un código de longitud variable sin tabla. Una diferencia de 0 se codifica con 1 bit, una diferencia que es una de las 15 diferencias más pequeñas posible excluyendo el 0 se codifica con 5 bits, y el resto de las diferencias se codifican usando un código sin tabla.

10 Como un ejemplo, considérese el siguiente caso donde los tamaños de subbanda para un caso no uniforme por defecto se proporcionan como se muestra en la Tabla 4.

Tabla 4							
Tamaños de banda:	4	4	8	8	16	16	16
Multiplicadores de banda:	1	1	2	2	4	4	4

Supóngase además, que después de dividir/unir, se crea la siguiente configuración de subbanda optimizada como se muestra en la Tabla 5.

Tabla 5							
Tamaños de banda:	2	4	10	24	8	8	16

15 La Figura 14 es un diagrama de unas series ejemplares de transformaciones de tamaño de subbanda. Por ejemplo, los tamaños de subbanda en la Tabla 5 pueden obtenerse desde la Tabla 4 mediante las transformaciones de la Figura 14.

20 Usar la fórmula anterior para  $\text{minRatioBandSize} = \max(1, 4/2) = 2$ , se proporciona el tamaño de subbanda de relación mínima de 2, y pueden obtenerse los valores para multiplicadores de tamaño de banda como se muestra en la Tabla 6.

Tabla 6							
Tamaños de banda:	2	4	10	24	8	8	16
Multiplicador de banda:	1	2	5	12	4	4	8
<code>minRatioBandSize</code> :	2						

25 Se usa un método para calcular el multiplicador de subbanda esperado. En primer lugar, supóngase que los bloques que no están divididos o unidos debieran tener el multiplicador de tamaño de banda por defecto (multiplicador de tamaño de banda esperado = multiplicador de tamaño de banda real). Esto ahorra bits puesto que únicamente necesitan codificarse cambios desde el multiplicador de tamaño de banda esperado. Además, cuanto menor es la modificación desde la configuración de banda por defecto, menos bits son necesarios para codificar la configuración. De otra manera, el multiplicador de banda esperado se calcula en el decodificador usando la siguiente lógica.

- 30 • Ver qué subbanda en la configuración por defecto estamos decodificando actualmente mirando en el punto de inicio de la banda real y comparar con los puntos de inicio y final de las bandas en la configuración de banda por defecto.
- El multiplicador de banda esperado se calcula tomando el número de coeficientes restantes en la banda en la configuración por defecto y dividiendo por el tamaño de bloque (subbanda) más pequeño en la configuración real.

35 Por ejemplo, siendo  $s_d[j]$  la posición de inicio de la banda de orden 'j' en la configuración de banda por defecto, siendo  $s_a[j]$  la posición de inicio de la banda de orden 'j' en la configuración de banda real, siendo  $m_d$  el tamaño de banda mínimo en el caso por defecto, y siendo  $m_a$  el tamaño de banda mínimo en el caso real. A continuación, calcular lo siguiente,

$$r = \max(1, m_d / m_a)$$

$$a[j] = (s_a[j+1] - s_a[j]) / m_a$$

5 donde 'r' es el minRatioBandSize, y a[j] es el multiplicador de banda para la banda de orden 'j'. Para calcular el multiplicador esperado para la banda de orden 'j', calcular en primer lugar 'i', el índice de la configuración de banda por defecto que contiene la posición de inicio de la banda real. A continuación, calcular  $a_{expected}[j]$  para que sea el multiplicador esperado de la banda de orden 'j'. Esto puede calcularse como sigue,

$$s_d[i] \leq s_a[j] < s_d[i+1]$$

$$a_{expected}[j] = (s_d[i+1] - s_a[j]) / m_a$$

10 Obsérvese que si una banda no se divide o une, entonces el multiplicador de banda esperado será el mismo que el real. También, siempre que  $s_d[i+1]$  sea el mismo que  $s_a[j+1]$ , entonces el multiplicador de banda esperado será el mismo que el real.

Continuando con el ejemplo, se muestra una configuración de subbanda por defecto en la Tabla 7.

Tabla 7							
Tamaños de banda :	4	4	8	8	16	16	16
Índice de banda :	0	1	2	3	4	5	6
Punto de inicio :	0	4	8	16	24	40	56
Punto final :	4	8	16	24	40	56	72

15 Se muestran las subbandas reales u optimizadas como se mapean a la configuración de banda por defecto en la Tabla 8.

Tabla 8							
Tamaños de banda :	2	4	10	24	8	8	16
Multiplicador de banda :	1	2	5	12	4	4	8
Punto de inicio :	0	2	6	16	40	48	56
Índice de Banda por defecto :	0	0	1	3	5	5	6
Coefficientes restantes :	4	2	2	16	16	8	16
Múltiples bandas esperadas :	2	1	1	8	8	4	8
Diferencia :	-1	1	4	4	-4	0	0

20 El Índice de Banda por Defecto es el valor de 'i' para una j dada. Coeficientes restantes es  $s_d[i+1] - s_a[j]$ . El Multiplicador de Banda Esperado es  $a_{expected}[j]$ , y el Multiplicador de Banda es a[j]. De nuevo, obsérvese que cualquier subbanda que no se divida o se una siempre tendrá una diferencia de 0. La codificación codificará el valor de "Diferencia" para cada subbanda y el minRatioBandSize ('r') para la configuración usando un código de longitud variable para cada una. El uso de minRatioBandSize permite codificar una configuración de banda en la que las bandas más pequeñas son más pequeñas que las bandas en la configuración por defecto.

Entorno informático

25 La Figura 15 ilustra un ejemplo generalizado de un entorno (1500) informático adecuado en el que pueden implementarse las realizaciones ilustrativas. El entorno (1500) informático no se pretende para sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la invención, ya que la presente invención puede



implementarse en diversos entornos informáticos de fin general o de fin especial.

Con referencia a la Figura 15, el entorno (1500) informático incluye al menos una unidad (1510) de procesamiento y memoria (1520). En la Figura 15, esta configuración (1530) más básica se incluye con una línea discontinua. La unidad (1510) de procesamiento ejecuta instrucciones ejecutables por ordenador y puede ser un procesador real o uno virtual. En un sistema multi-procesamiento, múltiples unidades de procesamiento ejecutan instrucciones ejecutables por ordenador para aumentar la potencia de procesamiento. La memoria (1520) puede ser memoria volátil (por ejemplo, registros, caché, RAM), memoria no volátil (por ejemplo, ROM, EEPROM, memoria flash, etc.), o alguna combinación de las dos. La memoria (1520) almacena software (1580) que implementa un codificador y/o decodificador de audio.

Un entorno informático puede tener características adicionales. Por ejemplo, el entorno (1500) informático incluye el almacenamiento (1540), uno o más dispositivos (1550) de entrada, uno o más dispositivos (1560) de salida, y una o más conexiones (1570) de configuración. Un mecanismo de interconexión (no mostrado) tal como un bus, controlador o red interconecta los componentes del entorno (1500) informático. Típicamente, el software de sistema operativo (no mostrado) proporciona un entorno operativo para que otro software se ejecute en el entorno (1500) informático, y coordina las actividades de los componentes del entorno (1500) informático.

El almacenamiento (1540) puede ser extraíble o no extraíble, e incluye discos magnéticos, cintas o casetes magnéticos, CD-ROM, CD-RW, DVD o cualquier otro medio que pueda usarse para almacenar información y que pueda accederse en el entorno (1500) informático. El almacenamiento (1540) almacena instrucciones para que el software (1580) implemente el codificador y/o decodificador de audio.

El dispositivo o dispositivos (1550) de entrada pueden ser un dispositivo de entrada táctil tal como un teclado, ratón, lapicero o bola de mando, un dispositivo de entrada de voz, un dispositivo de exploración u otro dispositivo que proporcione entrada al entorno (1500) informático. Para audio, el dispositivo o dispositivos (1550) de entrada pueden ser una tarjeta de sonido o dispositivo similar que acepte entrada de audio en forma analógica o digital. El dispositivo o dispositivos (1560) de salida pueden ser una pantalla, impresora, altavoz u otro dispositivo que proporcione salida desde el entorno (1500) informático.

La conexión o conexiones (1570) de comunicación posibilitan la comunicación a través de un medio de comunicación a otra entidad informática. El medio de comunicación transporta información tal como instrucciones ejecutables por ordenador, información de audio o vídeo comprimido u otros datos en una señal de datos modulada. Una señal de datos modulada es una señal que tiene una o más de sus características establecidas o cambiadas de tal manera para codificar información en la señal. A modo de ejemplo, y no como limitación, los medios de comunicación incluyen técnicas alámbricas o inalámbricas implementadas con un soporte eléctrico, óptico, de RF, de infrarrojos, acústico u otro.

La invención puede describirse en el contexto general de medio legible por ordenador. Medio legible por ordenador es cualquier medio disponible que pueda accederse dentro de un entorno informático. A modo de ejemplo, y no como limitación, con el entorno (1500) informático, medio legible por ordenador incluye la memoria (1520), almacenamiento (1540), medio de comunicación y combinaciones de cualquiera de lo anterior.

La invención puede describirse en el contexto general de instrucciones ejecutables por ordenador, tales como aquellas incluidas en módulos de programa, que se ejecutan en un entorno informático en un procesador real o virtual objetivo. En general, los módulos de programa incluyen rutinas, programas, bibliotecas, objetos, clases, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. La funcionalidad de los módulos de programa puede combinarse o dividirse entre módulos de programa según se desee en diversas realizaciones. Las instrucciones ejecutables por ordenador para módulos de programa pueden ejecutarse dentro de un entorno informático local o distribuido.

Para fines de presentación, la descripción detallada usa términos como “determinar”, “obtener”, “ajustar” y “aplicar”, para describir operaciones informáticas en un entorno informático. Estos términos son abstracciones de alto nivel para operaciones realizadas mediante un ordenador, y no deberían confundirse con actos realizados por un ser humano. Las operaciones informáticas reales que corresponden a estos términos pueden variar dependiendo de la implementación.

**REIVINDICACIONES**

1. Un método de codificación de audio, que comprende:
- transformar una señal de audio de entrada en un conjunto de coeficientes espectrales;
- codificar una porción de banda base del conjunto de coeficientes espectrales en el flujo de bits de salida;
- 5 dividir una banda extendida de los coeficientes espectrales en diversas subbandas;
- cambiar de escala las diversas subbandas en la banda extendida;
- transformar al menos una palabra de código desde una biblioteca de palabras de código;
- comparar el conjunto de coeficientes espectrales de una subbanda a al menos unas palabras de código transformadas desde la biblioteca;
- 10 codificar los coeficientes espectrales de la subbanda en un flujo de bits de salida que comprende codificar un identificador de una o más palabras de código desde la biblioteca y un identificador de transformador.
2. El método de codificación de audio de la reivindicación 1, que comprende adicionalmente:
- comparar el conjunto de coeficientes espectrales de la subbanda con al menos una palabra de código desde la biblioteca que no se ha transformado, en el que la biblioteca comprende diversas palabras de código desde la porción de banda base.
- 15 3. El método de codificación de audio de la reivindicación 1, en el que las transformaciones disponibles para transformar al menos una palabra de código desde la biblioteca comprenden una o más de las siguientes transformaciones:
- aplicar un exponente a cada coeficiente de una palabra de código;
- 20 negar cada coeficiente de una palabra de código; o
- invertir el orden de los coeficientes en una palabra de código.
4. El método de codificación de audio de la reivindicación 1, en el que transformar al menos una palabra de código desde la biblioteca comprende crear una palabra de código con coeficientes desde dos o más palabras de código que comprende:
- 25 desde todas menos la palabra de código final, seleccionar coeficientes que satisfacen una regla;
- desde una palabra de código final, proporcionar los otros coeficientes.
5. El método de codificación de audio de la reivindicación 1, en el que la biblioteca comprende adicionalmente palabras de código desde un libro de códigos de ruido o una palabra de código rellena usando un generador de números aleatorio sembrado de manera determinativa.
- 30 6. El método de codificación de audio de la reivindicación 1, en el que codificar los coeficientes espectrales de la subbanda incluye proporcionar un identificador de dos o más palabras de código y el identificador de transformación comprende al menos uno de una indicación de exponente, una indicación de signo, una indicación de dirección o una ordenación de identificadores de palabra de código en el flujo de bits de salida, indicando la ordenación una selección implícita de coeficientes.
- 35 7. El método de codificación de audio de la reivindicación 1, en el que codificar los coeficientes espectrales de la subbanda en el flujo de bits de salida incluye un identificador de dos o más palabras de código y el identificador de transformación es un identificador de una regla explícita para selección de coeficientes desde las dos o más palabras de código.
- 40 8. El método de codificación de audio de la reivindicación 1, en el que la al menos una palabra de código transformada comparada desde la biblioteca son dos o más palabras de código creadas usando una transformación exponencial de una palabra de código coincidente más cercana desde la biblioteca.

9. El método de codificación de audio de la reivindicación 8, en el que la palabra de código coincidente más cercana desde la biblioteca se identifica usando una comparación de media mínima cuadrática y las dos o más palabras de código creadas desde la transformación exponencial se comparan usando una función de masa de probabilidad.
- 5 10. El método de codificación de audio de la reivindicación 1, en el que las palabras de código comparadas comprenden diversas palabras de código desde la biblioteca y comparar el conjunto de coeficientes espectrales de la subbanda con la al menos una palabra de código transformada desde la biblioteca comprende una búsqueda exhaustiva en las palabras de código de la biblioteca y las transformaciones de la misma que comprenden negación, dirección inversa y transformaciones exponenciales usando dos o más exponentes.
- 10 11. El método de codificación de audio de la reivindicación 1, en el que transformar al menos una palabra de código desde la biblioteca comprende crear una palabra de código con coeficientes desde dos o más palabras de código que comprende:
- desde una primera palabra de código, seleccionar coeficientes que satisfacen una regla; y
- para coeficientes en la primera palabra de código que no satisfacen la regla, realizar una operación matemática para crear otros coeficientes, comprendiendo la operación matemática un operador y diversos operandos,
- 15 siendo un primer operando un coeficiente desde la primera palabra de código que no satisface la regla, y
- siendo un segundo operando un coeficiente obtenido desde una segunda palabra de código.
12. El método de codificación de audio de la reivindicación 1, que comprende adicionalmente preseleccionar palabras de código antes de comparar la subbanda con palabras de código, comprendiendo la preselección:
- crear una envolvente que comprende ejecutar una función de media ponderada en una señal de audio; y
- 20 determinar las palabras de código preseleccionadas comparando la envolvente con la subbanda.
13. El método de codificación de audio de la reivindicación 12, en el que comparar la envolvente con la subbanda comprende adicionalmente:
- transformar la envolvente usando una o más transformaciones que comprenden una transformación de negación, una transformación inversa o una transformación exponencial; y
- 25 en el que comparar la envolvente con la subbanda comprende determinar una distancia euclídea.
14. Un método de decodificación de audio que comprende:
- decodificar coeficientes espectrales codificados en un flujo de bits; y
- decodificar una o más subbandas codificadas en el flujo de bits que comprende,
- determinar uno o más identificadores de palabra de código para cada subbanda,
- 30 obtener la una o más palabras de código determinadas para cada subbanda, y
- para al menos una subbanda, determinar una regla de transformación,
- para la al menos una subbanda, transformar una palabra de código obtenida para la subbanda usando la regla de transformación.
- 35 15. El método de decodificación de audio de la reivindicación 14, en el que la regla de transformación determinada comprende una o más de las siguientes transformaciones:
- aplicar un exponente a cada coeficiente de una palabra de código;
- negar cada coeficiente de una palabra de código; o
- invertir el orden de coeficientes en una palabra de código.
16. El método de decodificación de audio de la reivindicación 14, en el que la regla de transformación determinada

crea una palabra de código desde dos o más palabras de código que comprende:

de todas menos la palabra de código final, seleccionar coeficientes que satisfacen una regla; y

desde una palabra de código final, proporcionar los otros coeficientes.

17. Un codificador de audio que comprende:

- 5 un transformador para transformar un bloque de señal de audio de entrada en coeficientes espectrales;  
un codificador de base para codificar valores de una porción de banda base de coeficientes espectrales en un flujo de bits;  
un divisor para dividir una porción de coeficientes espectrales en subbandas;  
un cambiador de escala para cambiar de escala subbandas;
- 10 un comparador para comparar coeficientes espectrales de las subbandas con palabras de código desde una biblioteca de palabras de código;  
un codificador (350) de banda extendida para codificar coeficientes espectrales de las subbandas en el flujo de bits, en el que una codificación de los coeficientes espectrales para una subbanda comprende un identificador de una palabra de código y un exponente para transformar la palabra de código identificada.

15

# Figura 1

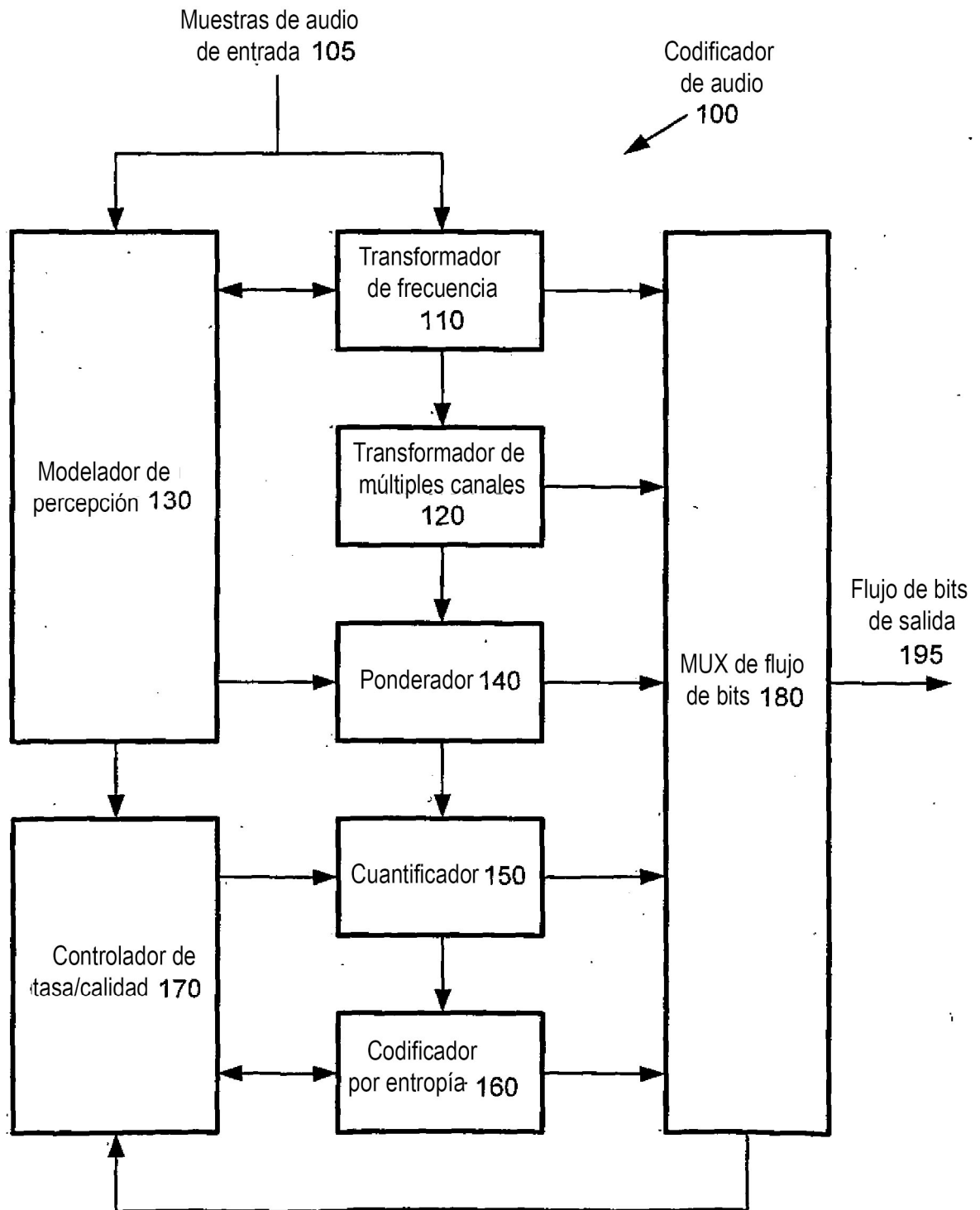


Figura 2

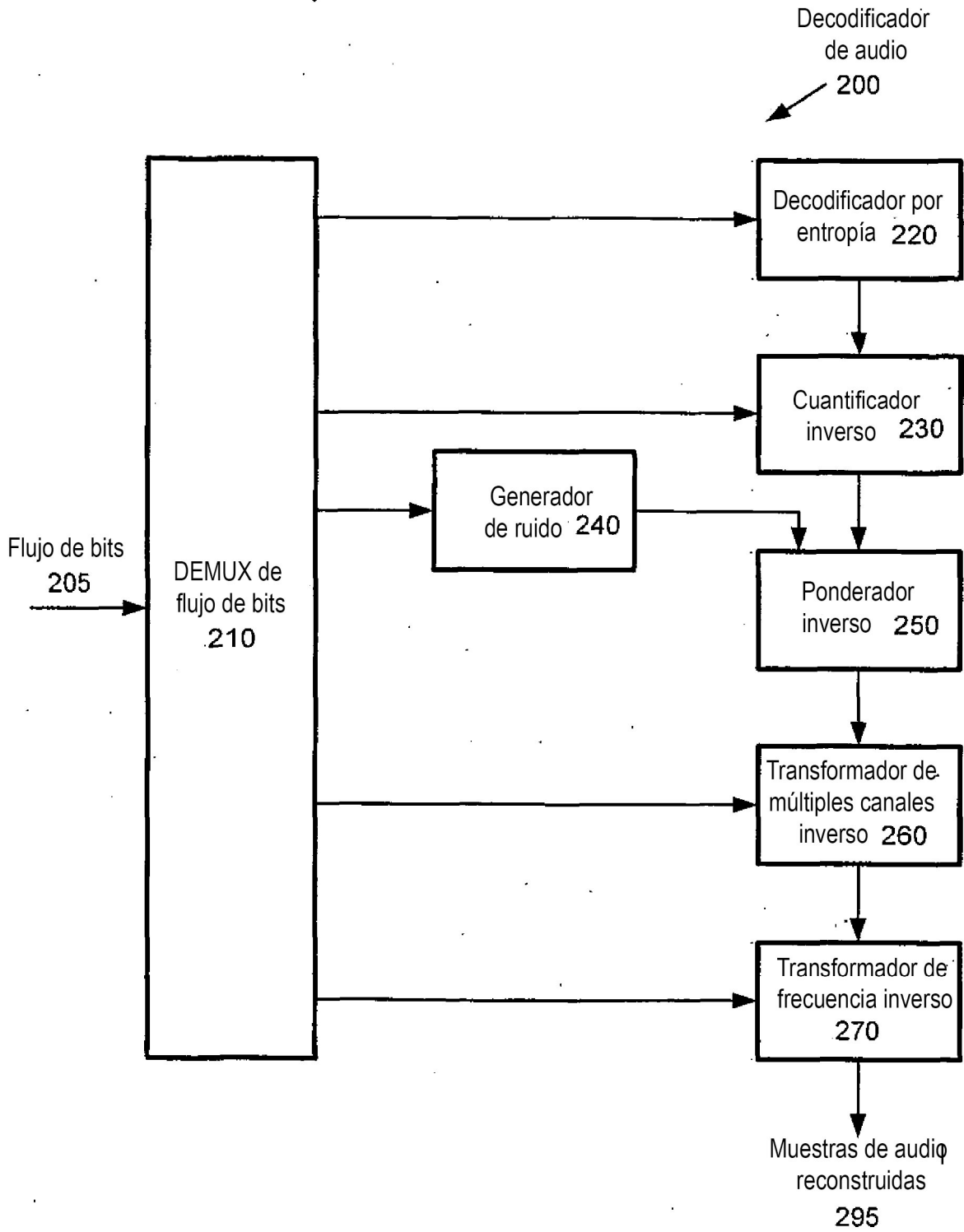


Figura 3

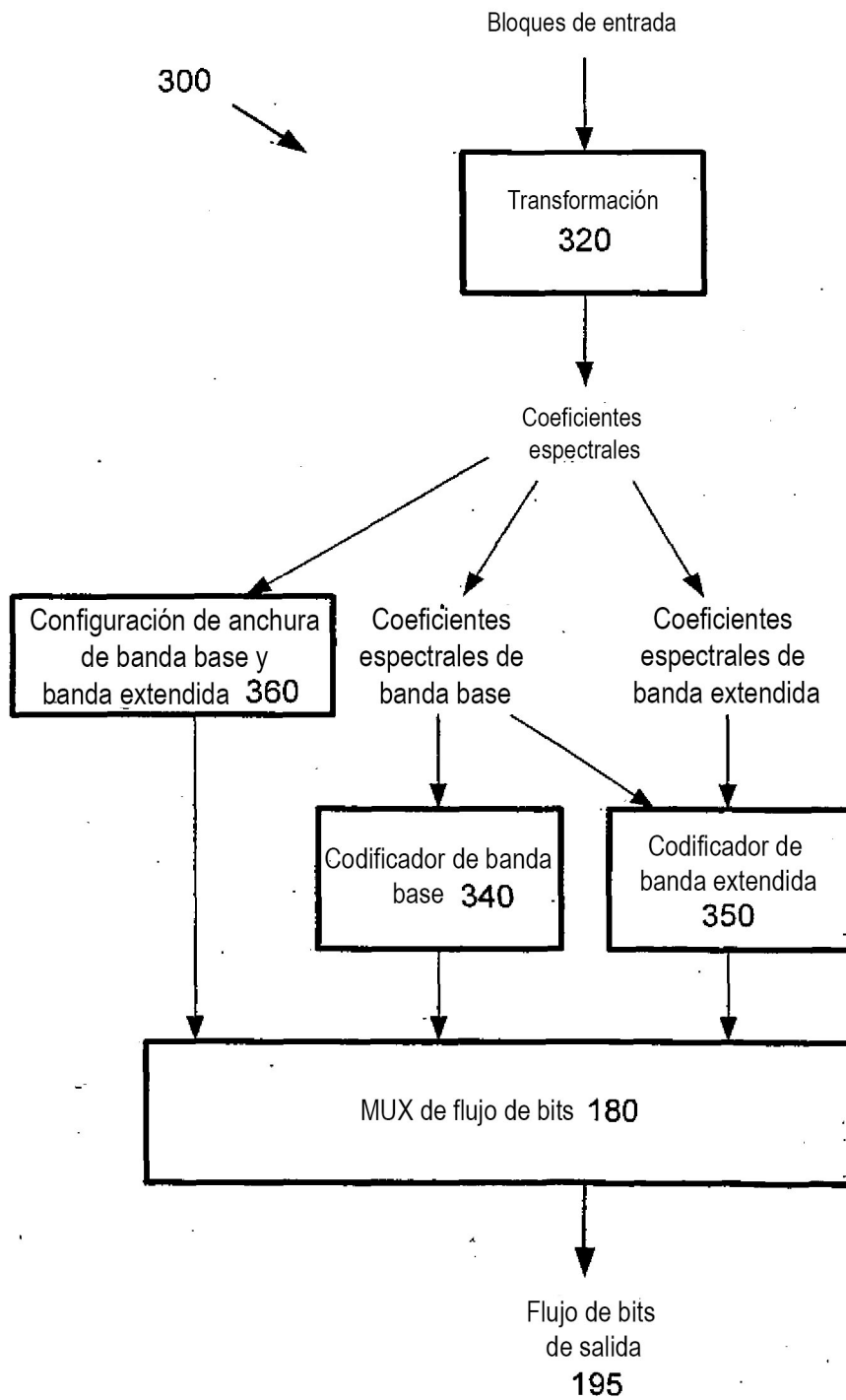


Figura 4

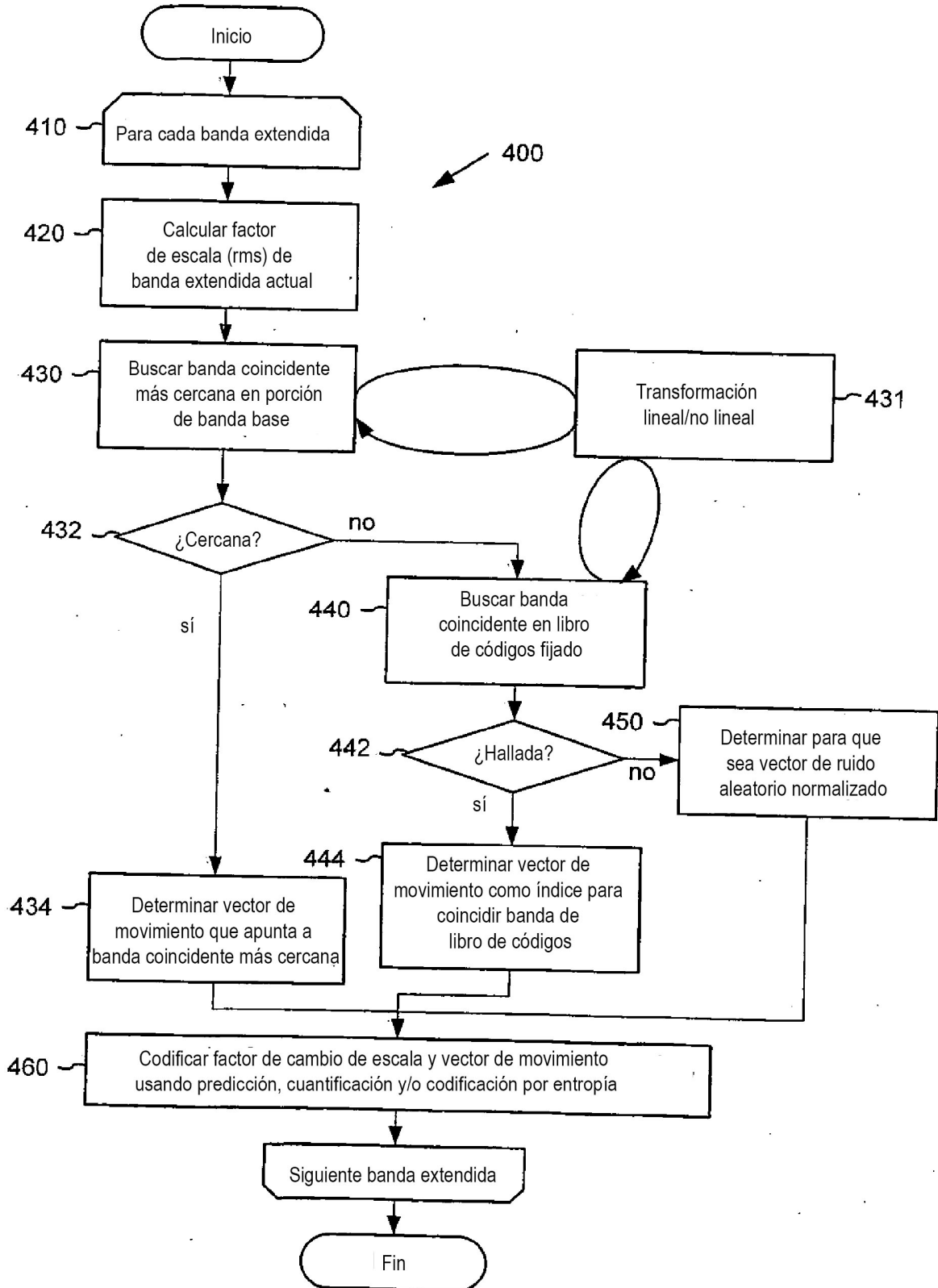




Figura 5

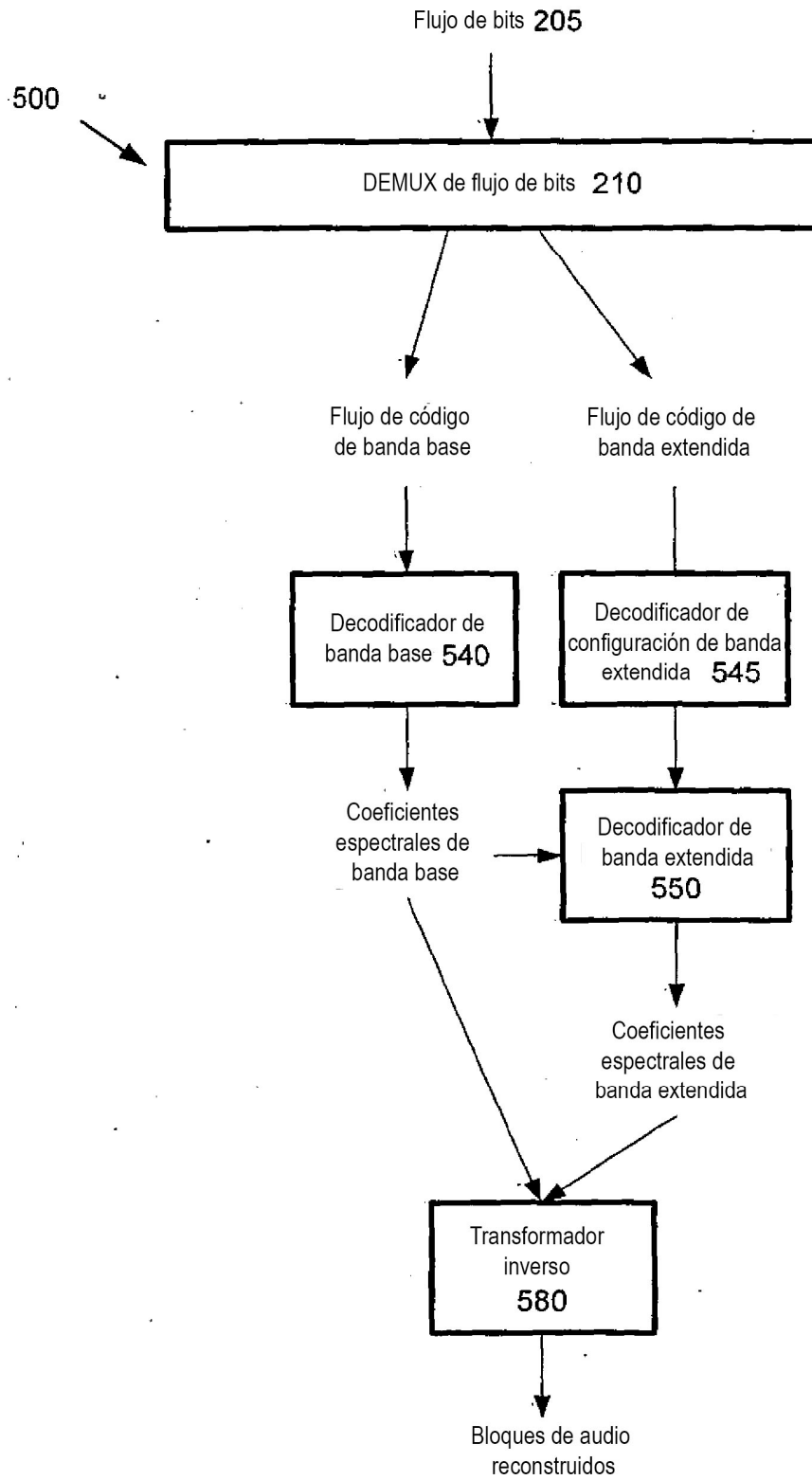
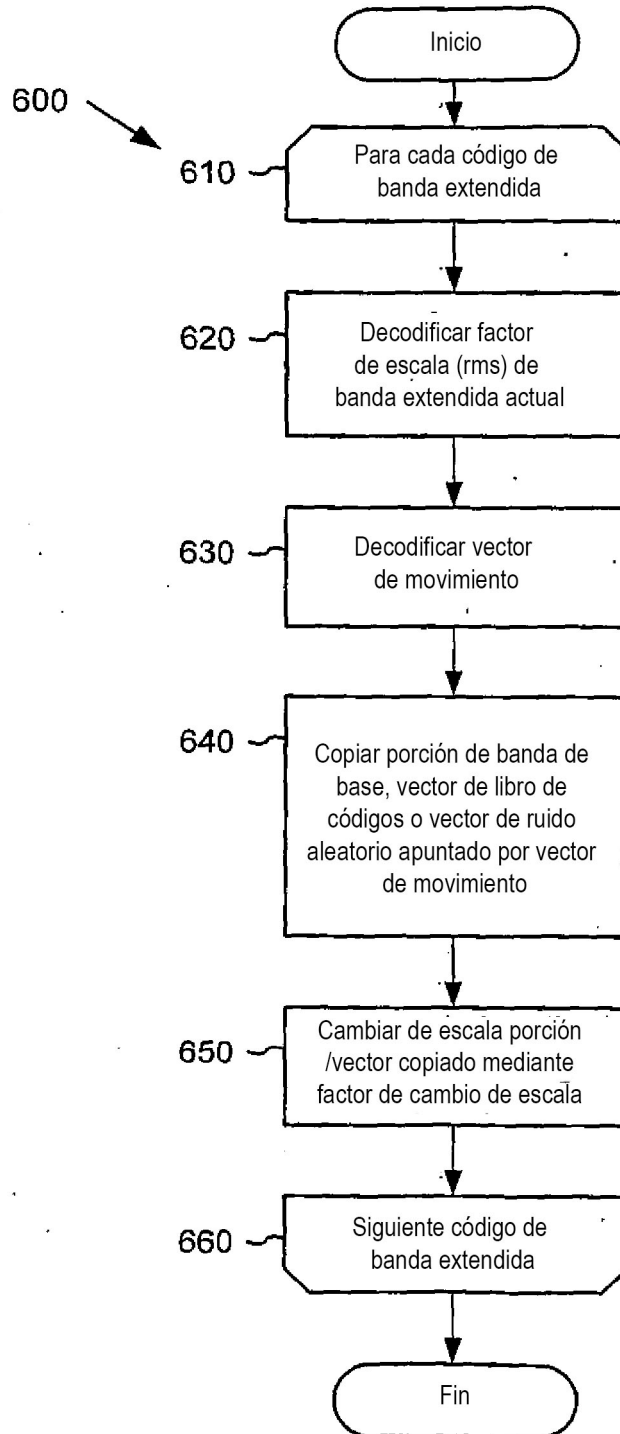
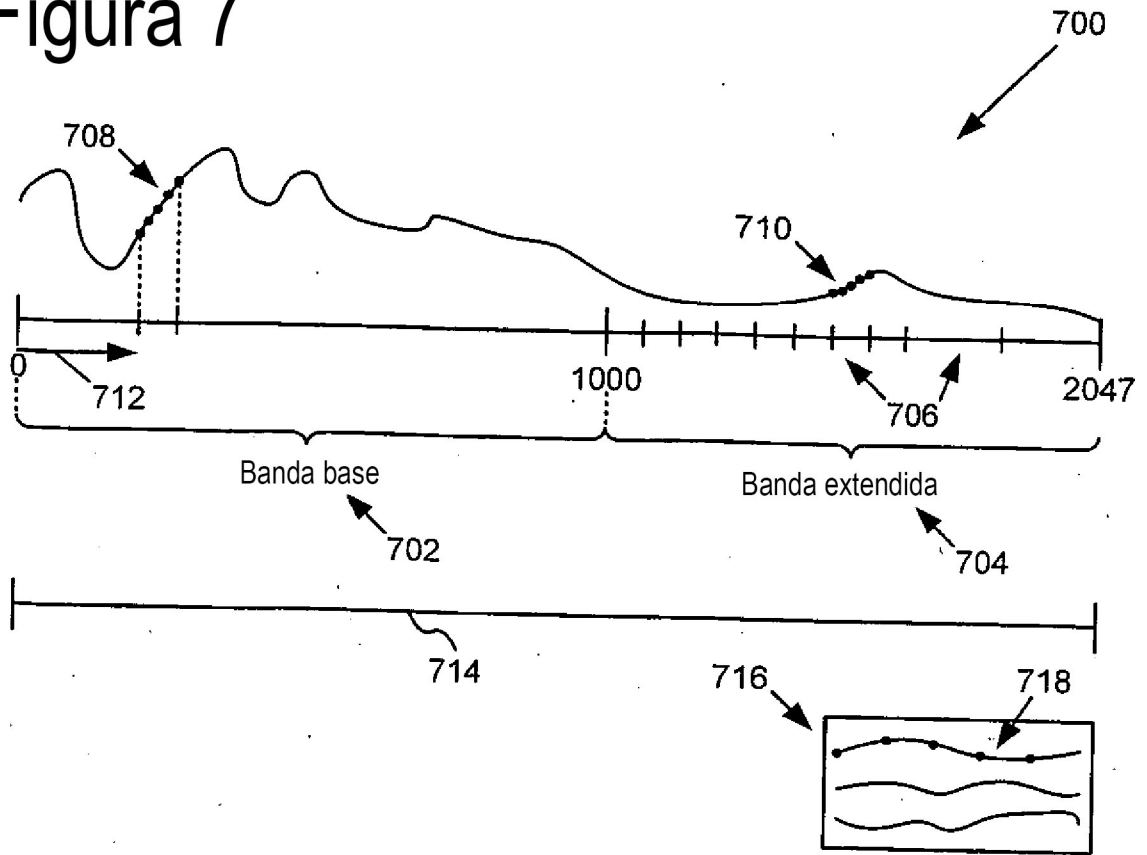


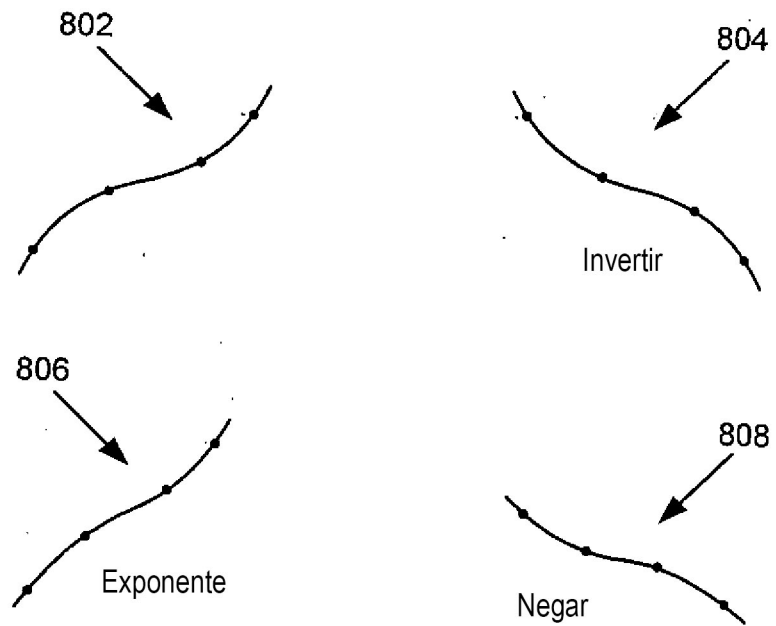
Figura 6



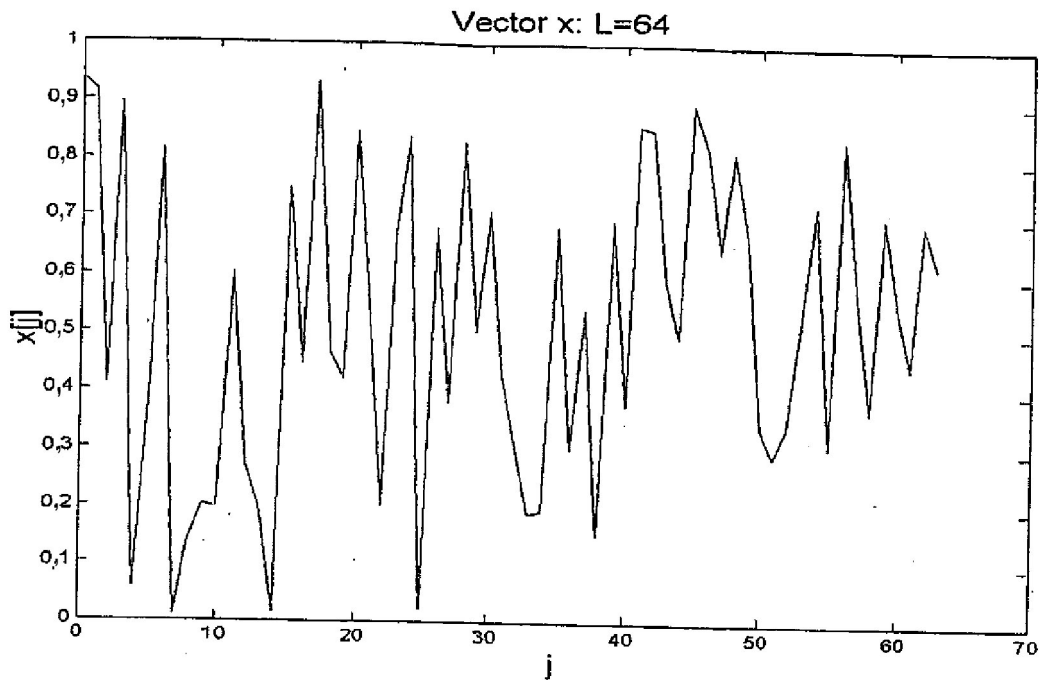
# Figura 7



# Figura 8



# Figura 9



# Figura 10

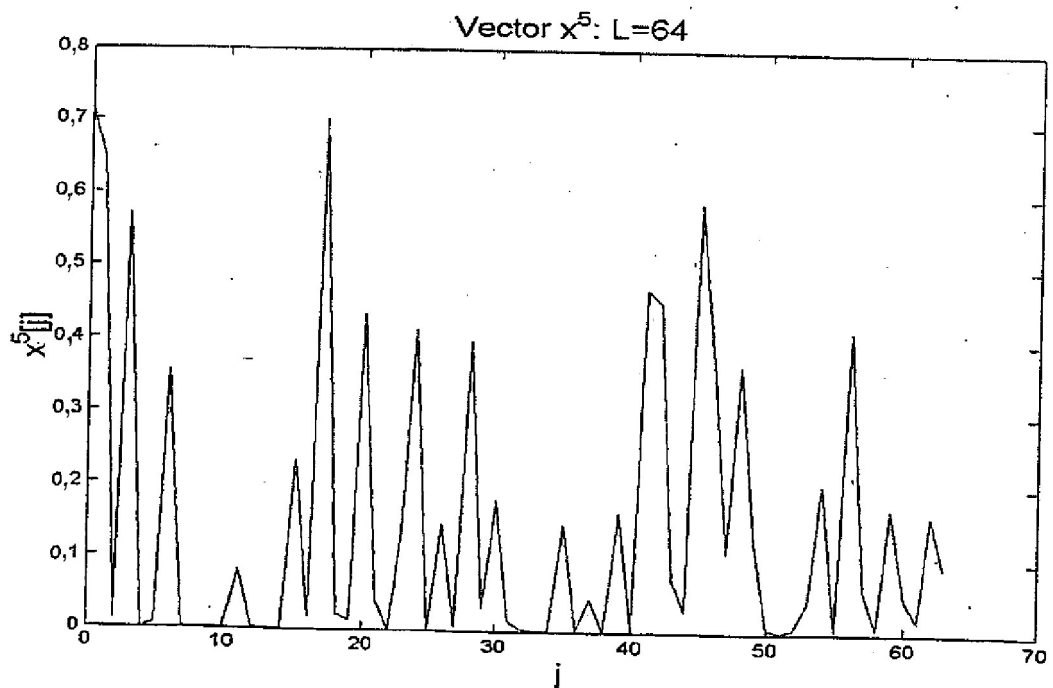


Figura 11

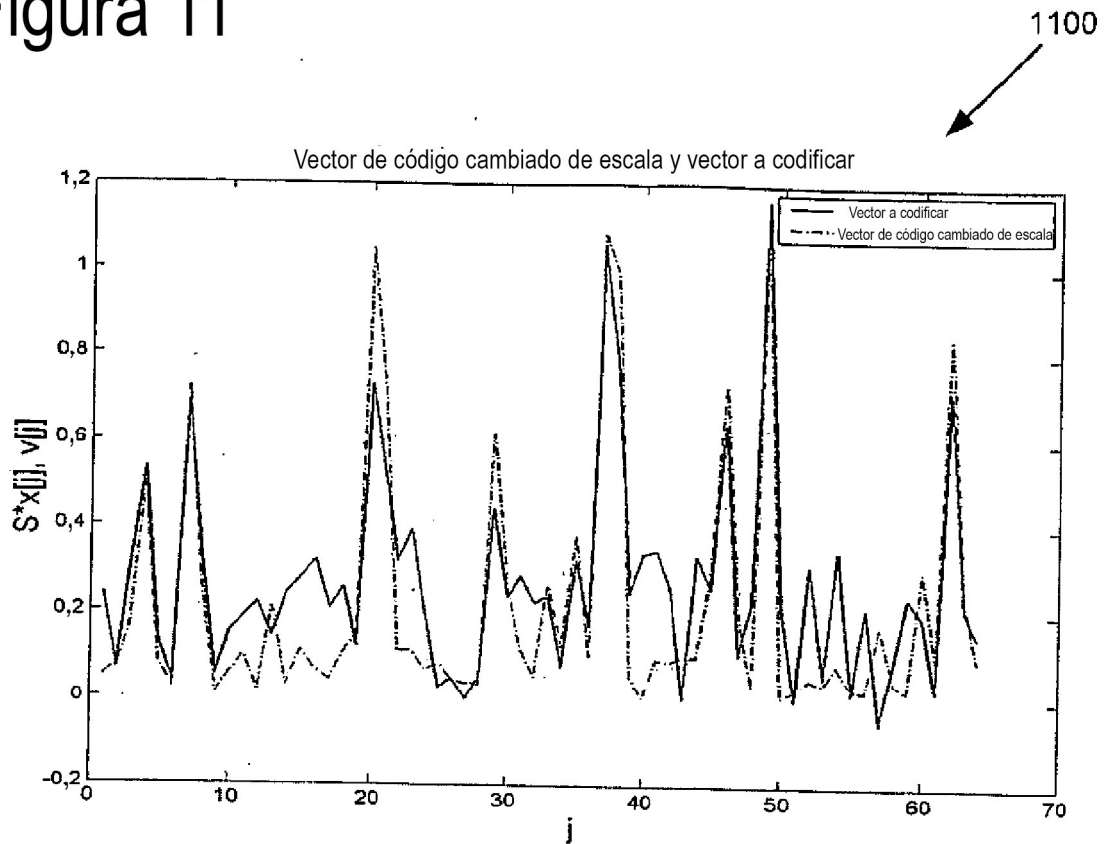


Figura 12

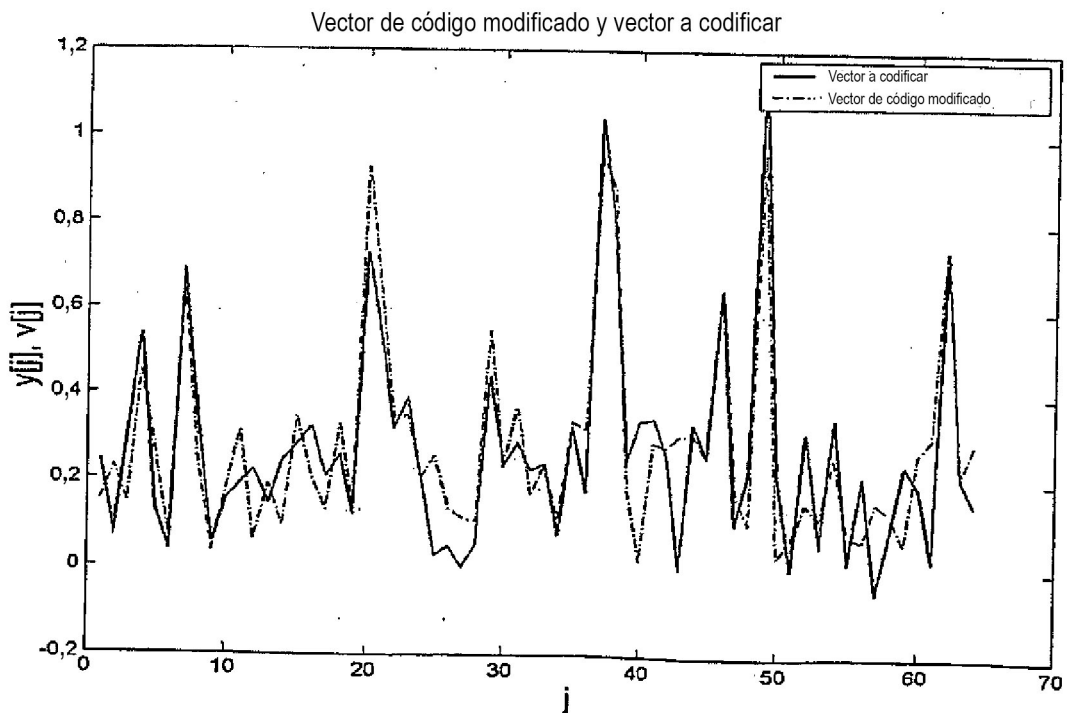


Figura 13

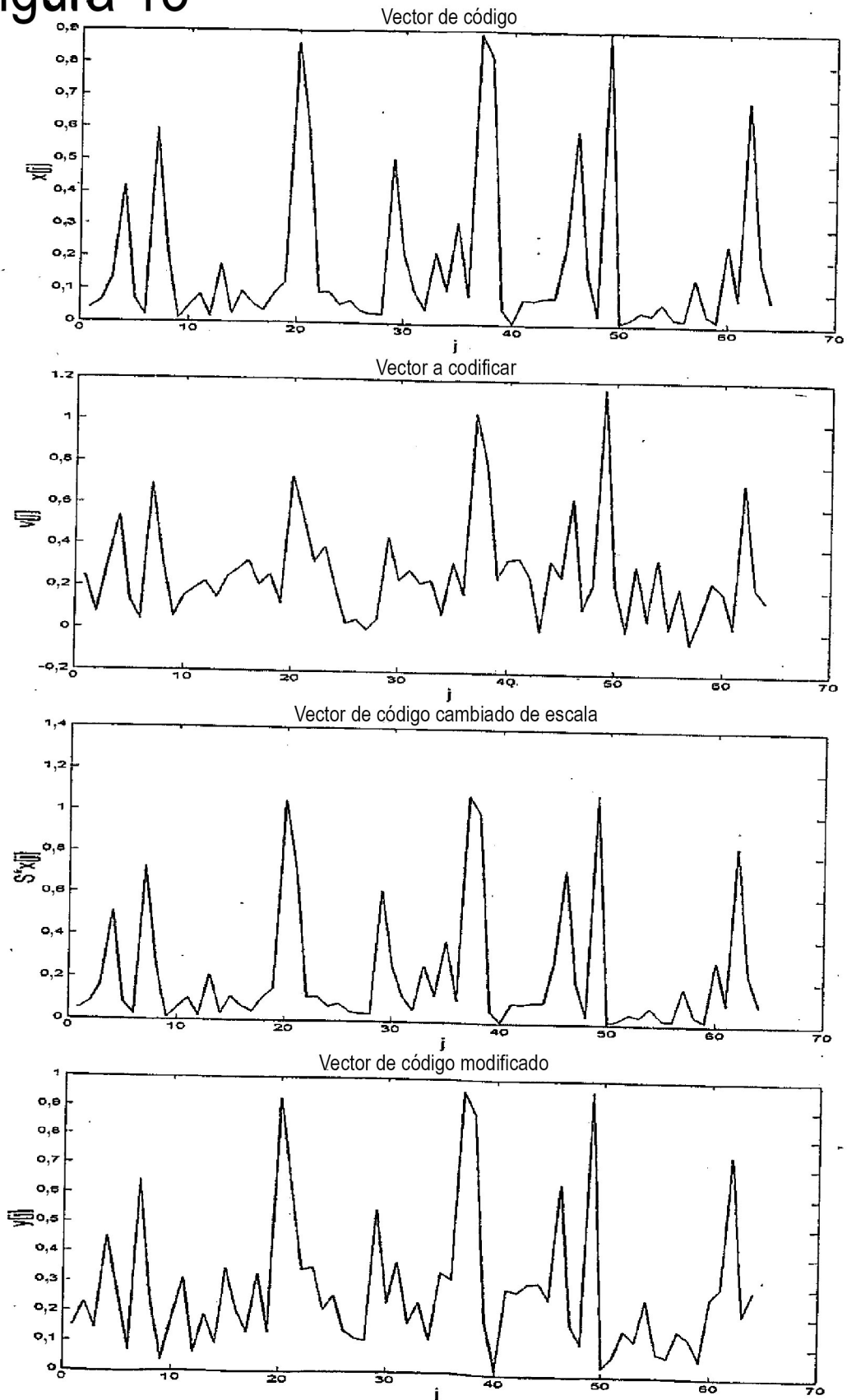


Figura 14

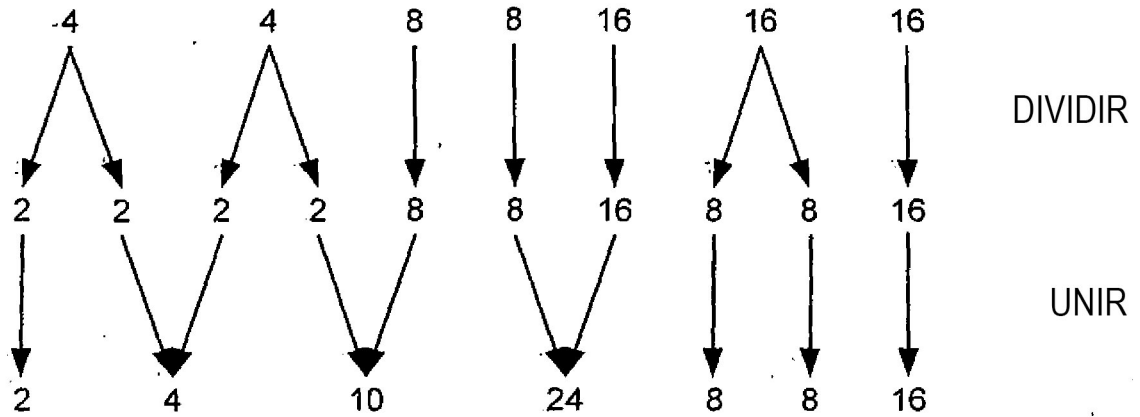
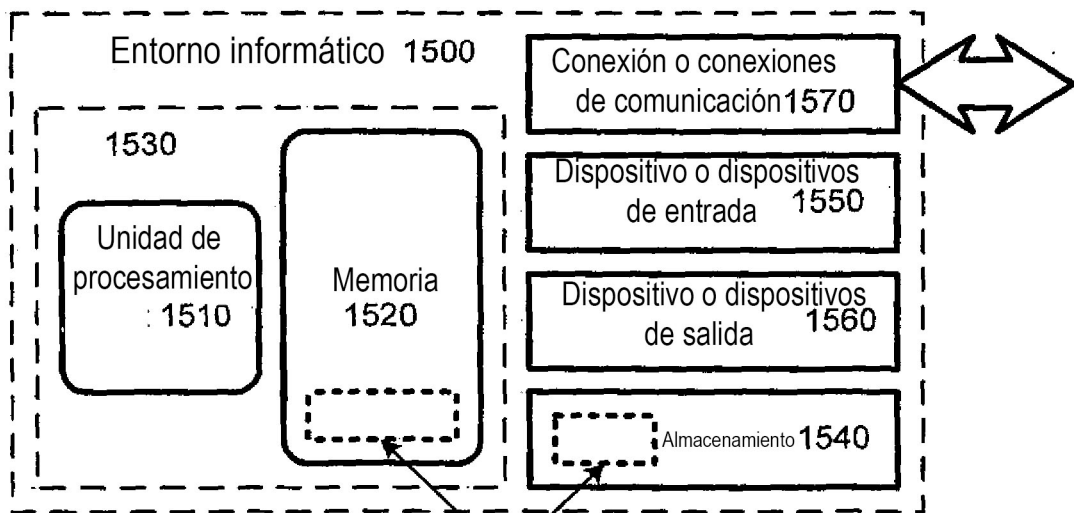


Figura 15



Software 1480 que implementa modificación de codificación/decodificación de palabras de código y/o segmentación de frecuencia variable