(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 987 078**

(51) Int. Cl.:

G06F 16/683 (2009.01)
G10L 25/51 (2013.01)
G10L 25/54 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **28.07.2017 E 22151836 (8)**

(97) Fecha y número de publicación de la concesión europea: **19.06.2024 EP 4006748**

(54) Título: **Coincidencia de audio**

(30) Prioridad:

15.08.2016 GB 201613960

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
13.11.2024

(73) Titular/es:

INTRASONICS S.A.R.L. (100.0%)
12-14 Rue Léon Thyes
2636 Luxembourg, LU

(72) Inventor/es:

SCHALKWIJK, JEROME y
KELLY, PETER JOHN

(74) Agente/Representante:

PONTI & PARTNERS, S.L.P.

DESCRIPCIÓN

Coincidencia de audio

[0001] La presente invención se refiere a la toma de huellas de audio y, en particular, a procedimientos y aparatos para identificar un conjunto optimizado de filtros para su uso en la generación de una huella acústica y a dispositivos de usuario que generan huellas acústicas usando los filtros optimizados.

[0002] Hay una serie de técnicas de reconocimiento de audio conocidas que incluyen:

- reconocimiento de audio activo donde las marcas de agua se codifican dentro de una señal de audio que se puede extraer más tarde para identificar la señal de audio,
- reconocimiento de audio pasivo donde se muestrea una señal de audio y el audio muestreado se usa para identificar el audio de una base de datos de muestras de audio, y
- reconocimiento de audio híbrido que combina las técnicas activa y pasiva.

[0003] El solicitante ha propuesto una serie de diferentes técnicas de reconocimiento de audio activo y estas se describen, por ejemplo, en los documentos WO2008/145994, W02009/144470 y WO02/45273. Todas estas técnicas se basan en ocultar esteganográficamente datos dentro del audio a reconocer que a continuación es recuperado por un dispositivo de usuario. Estos sistemas funcionan bien, especialmente cuando el audio se capta a través de un micrófono en lugar de obtenerse eléctricamente directamente a través de un canal de radiodifusión. Como se discutió en estas solicitudes de patente anteriores, hay una serie de usos para estos sistemas que incluyen la encuesta de audiencia y la interacción del usuario con los medios de difusión. Sin embargo, para que funcione, el acceso a los medios de difusión debe proporcionarse antes (o durante) la transmisión para permitir la ocultación de las marcas de agua en el audio. Esto no siempre es posible.

[0004] Las técnicas de reconocimiento de audio pasivo no requieren la ocultación de una marca de agua, por lo que, en teoría, se pueden usar para reconocer cualquier muestra de audio. Sin embargo, las técnicas de reconocimiento de audio pasivo tienen la desventaja de que requieren una comparación más exigente entre el audio muestreado y una gran base de datos de muestras de audio. Además, cuando el audio es capturado por el micrófono del dispositivo de un usuario, es probable que el audio capturado sea relativamente ruidoso en comparación con el audio original y esto puede conducir fácilmente a errores en los resultados del reconocimiento. Una técnica común usada para emparejar pasivamente el audio es crear una "huella" acústica del sonido capturado y a continuación comparar esta huella con las huellas correspondientes de las señales de audio en la base de datos. La huella intenta capturar las características acústicas importantes de la señal de audio, lo que ayuda a reducir la carga de hacer corresponderse la muestra de audio con el audio de la base de datos. Sin embargo, estos sistemas de reconocimiento de huellas aún requieren un procesamiento significativo para poder comparar la huella de la consulta con las huellas de la base de datos, y esta carga de procesamiento continúa creciendo a medida que se crea más y más contenido multimedia. Se requieren técnicas para reducir esta carga de procesamiento.

[0005] El artículo "Fingerprint Information Maximisation For Content Identification", 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, de Naini y col., describe una técnica para generar huellas basada en la maximización de la información mutua a través del canal de distorsión.

[0006] El artículo "A Review of Algorithms for Audio Fingerprinting", 2002, IEEE, de Cano y col., proporciona una revisión de las técnicas de huellas de audio disponibles.

[0007] El artículo "A Highly Robust Audio Fingerprinting System", 2002 IRCAM, de Haitsma y col., describe un sistema de huellas de audio que permite a un usuario identificar una canción.

[0008] Al desarrollar su sistema de reconocimiento de audio basado en huellas, los inventores también idearon una técnica para generar huellas de audio que son robustas al ruido y otras interferencias y que hacen que sea más fácil distinguir entre huellas de diferentes muestras de audio.

[0009] La presente invención proporciona un procedimiento de identificación de un conjunto optimizado de filtros para su uso en la generación de una huella acústica, comprendiendo el procedimiento: i) proporcionar una o más bases de datos que comprenden una pluralidad de muestras de audio que incluyen N_M pares coincidentes de muestras de audio y N_N pares no coincidentes de muestras de audio, comprendiendo cada par coincidente de muestras de audio una muestra de audio original y una versión distorsionada de una misma señal de audio original y comprendiendo cada par no coincidente de muestras de audio una muestra de audio original y una versión de una señal de audio original diferente; ii) determinar un espectrograma para cada una de la pluralidad de muestras de audio en la una o más bases de datos; iii) aplicar cada uno de N_f filtros candidatos a los espectrogramas y binarizar un resultado para generar una pluralidad de vectores de bits binarios, estando asociado cada vector de bits binarios con un filtro candidato y una muestra de audio; iv) comparar los bits binarios en los vectores asociados con un par coincidente seleccionado de muestras de audio para un filtro actual para determinar la información de tasa de error de

bits para el filtro actual y el par coincidente seleccionado de muestras de audio; v) repetir la etapa iv) para cada par coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa iv) para el filtro actual y los pares coincidentes de muestras de audio; vi) comparar los bits binarios en los vectores asociados con un par no coincidente seleccionado de muestras de audio para el filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par no coincidente seleccionado de muestras de audio; vii) repetir la etapa vi) para cada par no coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa vi) para el filtro actual y los pares no coincidentes de muestras de audio; viii) repetir las etapas iv) a vii) para cada filtro candidato para determinar la información de media y varianza para cada filtro candidato para los pares coincidentes de muestras de audio y para determinar la información de media y varianza para cada filtro candidato para los pares no coincidentes de muestras de audio; y ix) determinar, usando una técnica de optimización de programación dinámica, un subconjunto de dichos filtros candidatos como dicho conjunto optimizado de filtros para su uso en la generación de una huella acústica usando la información de media y varianza determinada para cada filtro candidato para los pares coincidentes de muestras de audio y la información de media y varianza determinada para cada filtro candidato para los pares no coincidentes de muestras de audio.

[0010] La presente invención también proporciona un aparato para identificar un conjunto optimizado de filtros para su uso en la generación de una huella acústica, comprendiendo el aparato: una o más bases de datos que comprenden una pluralidad de muestras de audio que incluyen N_M pares coincidentes de muestras de audio y N_N pares no coincidentes de muestras de audio, comprendiendo cada par coincidente de muestras de audio una muestra de audio original y una versión distorsionada de una misma señal de audio original y comprendiendo cada par no coincidente de muestras de audio una muestra de audio original y una versión de una señal de audio original diferente; y uno o más procesadores configurados para: i) determinar un espectrograma para cada una de dicha pluralidad de muestras de audio en la una o más bases de datos; ii) aplicar cada uno de N_f filtros candidatos a los espectrogramas y binarizar un resultado para generar una pluralidad de vectores de bits binarios, estando asociado cada vector de bits binarios con un filtro candidato y una muestra de audio; iii) comparar los bits binarios en los vectores asociados con un par coincidente seleccionado de muestras de audio para un filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par coincidente seleccionado de muestras de audio; iv) repetir iii) para cada par coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa iii) para el filtro actual y los pares coincidentes de muestras de audio; v) comparar los bits binarios en los vectores asociados con un par no coincidente seleccionado de muestras de audio para el filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par no coincidente seleccionado de muestras de audio; vi) repetir v) para cada par no coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en v) para el filtro actual y los pares no coincidentes de muestras de audio; vii) repetir iii) a vi) para cada filtro candidato para determinar la información de media y varianza para cada filtro candidato para los pares coincidentes de muestras de audio y para determinar la información de media y varianza para cada filtro candidato para los pares no coincidentes de muestras de audio; y viii) determinar, usando una técnica de optimización de programación dinámica, un subconjunto de dichos filtros candidatos como dicho conjunto optimizado de filtros para su uso en la generación de una huella acústica usando la información de media y varianza determinada para cada filtro candidato para los pares coincidentes de muestras de audio y la información de media y varianza determinada para cada filtro candidato para los pares no coincidentes de muestras de audio.

[0011] La presente invención también proporciona un dispositivo de usuario para su uso en un sistema de coincidencia de audio, comprendiendo el dispositivo de usuario: medios para capturar una señal de audio; medios para procesar la señal de audio capturada para generar una huella acústica de consulta representativa de la señal de audio capturada usando un conjunto de filtros optimizados determinados usando el procedimiento descrito anteriormente; medios para emitir la huella acústica de consulta a un servidor de coincidencia de audio; y un medio para recibir una respuesta de coincidencia que comprende información relacionada con el audio capturado.

[0012] La invención también proporciona un producto de programa informático que comprende instrucciones implementables por ordenador para hacer que un dispositivo informático programable realice todas las etapas del procedimiento descritas anteriormente.

[0013] Estos y otros aspectos de la invención serán evidentes a partir de la siguiente descripción detallada de realizaciones ejemplares que se describen con referencia a los dibujos adjuntos donde:

La Figura 1 es un diagrama de bloques que ilustra los componentes principales de un sistema de correspondencia de audio;
La Figura 2 es un diagrama de bloques que ilustra los componentes principales de un teléfono celular que forma parte del sistema de coincidencia de audio de la Figura 1;
La Figura 3 es un diagrama de bloques funcionales que ilustra los principales componentes funcionales del software de aplicación que se ejecuta en el teléfono celular que se muestra en la Figura 2;
La Figura 4 ilustra el funcionamiento de una unidad de análisis de frecuencia que forma parte del software de aplicación que se muestra en la Figura 3 y que ilustra la forma en que una señal de audio capturada se divide en tramas sucesivas

que se analizan en frecuencia para generar un espectrograma de la señal de audio;
La Figura 5a ilustra una cantidad de filtros básicos que son usados por una unidad de generación de huellas para generar una huella a partir del espectrograma;
La Figura 5b ilustra el espectrograma generado por la unidad de análisis de frecuencia y la forma en que se aplica un filtro al espectrograma;
Las Figuras 5c y 5d ilustran la forma en que la unidad de generación de huellas aplica un filtro al espectrograma para generar un vector de valores combinados obtenidos combinando valores seleccionados del espectrograma con coeficientes del filtro;
La Figura 5e ilustra la forma en que la unidad de generación de huellas binariza el vector de valores combinados al generar la huella;
La Figura 5f ilustra gráficamente la huella generada por la unidad de generación de huellas aplicando un primer conjunto de filtros al espectrograma y concatenando los vectores binarios resultantes para formar una huella acústica 2d;
La Figura 6 ilustra una huella convencional y una huella generada por el teléfono celular de la Figura 1 concatenando los vectores binarios en un orden específico para maximizar la probabilidad de que los vectores binarios adyacentes en la huella sean similares entre sí;
La Figura 7 es un diagrama de bloques que ilustra los componentes principales del servidor de coincidencia de audio 5 que forma parte del sistema que se muestra en la Figura 1;
Las Figuras 8a, 8b, 8c y 8d ilustran la forma en que una unidad de generación de huellas gruesas que forma parte del servidor que se muestra en la Figura 7 genera una huella de consulta gruesa a partir de la huella de consulta fina generada por el teléfono celular;
La Figura 9 ilustra la información contenida en las entradas dentro de una base de datos que forma parte del sistema de coincidencia de audio que se muestra en la Figura 1;
La Figura 10a ilustra la forma en que una unidad de coincidencia de huellas gruesas que forma parte del servidor que se muestra en la Figura 7 coincide con la huella de consulta gruesa con huellas gruesas almacenadas dentro de la base de datos que se ilustra en la Figura 9 para identificar un subconjunto de entradas de base de datos que pueden coincidir con la huella de consulta gruesa;
La Figura 10b ilustra un resultado obtenido por la unidad de coincidencia de huellas gruesas cuando no hay coincidencia entre la huella de consulta gruesa y una huella de base de datos gruesa;
La Figura 10c ilustra un resultado obtenido por la unidad de coincidencia de huellas gruesas cuando hay una coincidencia entre la huella de consulta gruesa y una huella de base de datos gruesa;
La Figura 11 ilustra la forma en que una unidad de coincidencia de huellas finas que forma parte del servidor que se muestra en la Figura 7 coincide con la huella de consulta fina con las huellas finas almacenadas dentro de la base de datos para el subconjunto de entradas de base de datos identificadas por la unidad de coincidencia de huellas gruesas para identificar la entrada de base de datos que mejor coincide con la huella de consulta fina;
Las Figuras 12a y 12b ilustran parte de un procedimiento de entrenamiento usado para determinar un primer conjunto de filtros optimizados que se usan para generar una huella fina a partir de un espectrograma;
La Figura 13a ilustra distribuciones separadas obtenidas para un filtro para hacer corresponder pares de muestras de audio de entrenamiento y para pares no emparejados de muestras de audio de entrenamiento;
La Figura 13b ilustra distribuciones superpuestas obtenidas para un filtro para hacer coincidir pares de muestras de audio de entrenamiento y para pares no coincidentes de muestras de audio de entrenamiento;
La Figura 14 ilustra un enrejado de nodos y un procedimiento de propagación de ruta para identificar una mejor ruta a través del enrejado; y
Las Figuras 15a y 15b ilustran parte de un procedimiento de entrenamiento usado para determinar un segundo conjunto de filtros optimizados que se usan para generar una huella gruesa a partir de una huella fina.

Visión general

[0014] La Figura 1 es un diagrama de bloques que ilustra los componentes principales de un sistema de coincidencia de audio que incorpora la presente invención. El sistema se basa en un usuario que tiene un dispositivo de usuario 1 (en este caso un teléfono celular) que puede capturar el sonido 2 generado por una fuente de sonido 3 (tal como un televisor 3-1, una radio 3-2 o una actuación en vivo, etc.). El dispositivo de usuario 1 procesa el sonido capturado 2 y genera una huella acústica que representa el sonido capturado. En la siguiente descripción, estas huellas acústicas se denominarán simplemente huellas para facilitar la explicación. La forma en que se genera la huella se describirá con más detalle a continuación. El dispositivo de usuario 1 a continuación transmite la huella generada como una consulta a un servidor de coincidencia de audio remoto 5 ya sea a través de la estación base 7 y la red de telecomunicaciones 9 o a través de un punto de acceso 11 y la red informática 13 (por ejemplo, Internet). En esta realización, en respuesta a la recepción de la huella de consulta, el servidor de coincidencia de audio 5 procesa la huella de consulta para generar una huella de consulta gruesa que el servidor a continuación usa para buscar entradas posiblemente coincidentes dentro de la base de datos 15. La huella gruesa tiene una resolución o velocidad de bits más baja en comparación con la huella de consulta recibida. Esta primera búsqueda usando la huella de consulta gruesa identificará un subconjunto de entradas posiblemente coincidentes dentro de la base de datos 15. El servidor de coincidencia de audio 5 a continuación compara la huella de consulta de mayor resolución (o "fina") recibida del dispositivo de usuario 1 con el subconjunto de entradas identificadas por la primera búsqueda para identificar la entrada de base de datos que es más similar a la huella de consulta fina. El servidor de coincidencia de audio 5 a continuación

emite información recuperada de la entrada de coincidencia en la base de datos 15, al dispositivo de usuario 1. Se puede devolver diversa información diferente, como la información de identificación del audio capturado por el dispositivo de usuario 1; información del artista; contenido relacionado (como otro contenido del mismo artista); e incluso enlaces informáticos a contenido almacenado en otros servidores conectados a la red informática 13. El dispositivo de usuario 1 a continuación emite la información devuelta al usuario, por ejemplo, a través de la pantalla 17 del dispositivo de usuario. Un ejemplo es cuando el procedimiento de coincidencia de audio identifica el anuncio o programa de televisión que está viendo un espectador y a continuación presenta contenido relevante al usuario. La información recuperada de la base de datos 15 se puede proporcionar a un tercero en lugar de o además del dispositivo de usuario 1. Esto puede ser útil en aplicaciones de encuestas de audiencia, donde el propósito del procedimiento de coincidencia de audio es identificar el programa de televisión o radio que el usuario está escuchando o viendo; cuya información se envía a continuación a un servidor de sondeo de audiencia de terceros 19 a través de la red informática 13.

[0015] A continuación se dará una descripción más detallada de las partes principales del sistema de comparación de audio descrito anteriormente.

Teléfono Móvil Del Usuario

[0016] La Figura 2 es un diagrama de bloques que ilustra los componentes principales del teléfono móvil 1 del usuario usado en esta realización. Como se muestra, el teléfono celular 1 incluye un micrófono 23 para recibir las señales acústicas 2 (como el sonido emitido por la televisión 3-1 o la radio 3-2) y para convertir estas señales acústicas en señales eléctricas equivalentes. Las señales eléctricas del micrófono 23 son filtradas a continuación por el filtro 51 para eliminar las frecuencias no deseadas, típicamente las que están fuera de la banda de frecuencia de 200 Hz a 20 kHz. El audio filtrado es a continuación digitalizado por un convertidor analógico a digital 53, que muestrea el audio filtrado típicamente a una frecuencia de muestreo de 24 o 48 kHz y representa cada muestra por un valor digital de 16 bits. El flujo de audio digitalizado (D(t)) se introduce a continuación en un procesador 63 (que puede comprender una o más unidades centrales de procesamiento).

[0017] Cuando se realiza una llamada de voz, el procesador 63 comprime el audio recibido y a continuación lo pasa a una unidad de procesamiento de RF 57 que modula los datos de audio comprimidos en una o más señales portadoras de RF para su transmisión a la estación base 7 a través de la antena 27. De manera similar, las señales de audio comprimidas recibidas a través de la antena 27 se alimentan a la unidad de procesamiento de RF 57, que demodula las señales de RF recibidas para recuperar los datos de audio comprimidos de la señal o señales portadoras de RF, que a continuación se pasan al procesador 63 para su descompresión. Las muestras de audio regeneradas a continuación se envían al altavoz 25 a través del convertidor digital a analógico 59 y el amplificador 61.

[0018] Como se muestra en la Figura 2, el procesador 63 está controlado por software almacenado en la memoria 65. El software incluye el software del sistema operativo 67 (para controlar el funcionamiento general del teléfono celular 1), un navegador 68 para acceder a Internet y el software de aplicación 69 para proporcionar funcionalidad adicional al teléfono celular 1. En esta realización, el software de aplicación 69 es parte del sistema de coincidencia de audio y hace que el teléfono celular capture el sonido 2 con fines de reconocimiento. El software de aplicación 69 también genera la huella fina descrita anteriormente que se envía al servidor de coincidencia de audio 5 como una consulta. El software de aplicación 69 también responde a los datos recibidos del servidor de coincidencia de audio 5, por ejemplo, emitiendo información al usuario en la pantalla 17; o recuperando información de otro servidor usando un enlace devuelto desde el servidor de coincidencia de audio 5.

Software de Aplicación - Análisis de Frecuencia

[0019] La Figura 3 es un diagrama de bloques que ilustra la funcionalidad de procesamiento principal del software de aplicación 69 usado en esta realización. Como se muestra, el software de aplicación 69 recibe como entrada la señal de audio muestreada (D(t)) del convertidor A/D 53. Este audio muestreado se almacena en un búfer de audio 32. Las muestras de audio en el búfer de audio 32 son procesadas por una unidad de análisis de frecuencia 34 que procesa las muestras de audio en el búfer de audio 32 para generar un espectrograma 35 de la señal de audio (D(t)) que se almacena en el búfer de espectrograma 37. El espectrograma 35 es una representación de tiempo y frecuencia de la señal de audio (D(t)) e ilustra la forma en que el contenido de frecuencia de la señal de audio (D(t)) cambia con el tiempo a lo largo de la duración de la señal de audio. La unidad de análisis de frecuencia 34 construye el espectrograma 35 extrayendo tramas de muestras de audio de la señal de audio entrante D(t) y determinando el contenido de frecuencia de la señal de audio en cada trama (es decir, qué frecuencias están presentes y a qué amplitudes). En particular, como se ilustra en la Figura 4, la señal de audio de entrada D(t) se divide en tramas superpuestas 39 para permitir un análisis espectral de "corto tiempo" de las muestras de audio en cada trama, como es estándar en el campo del procesamiento de audio. Típicamente, una trama 39 de muestras se extrae una vez cada 10 a 20 milisegundos y las tramas 39 pueden solaparse (como se ilustra) o no solaparse. Típicamente, la unidad de análisis de frecuencia 34 funciona en paralelo con la escritura de las muestras entrantes en el búfer de audio 32. En otras palabras, la unidad de análisis de frecuencia 34 puede comenzar su análisis tan pronto como la primera trama (f_1) de muestras de audio se escribe en el búfer de audio 32 y se detiene después de un tiempo predefinido o al final

del clip de audio que es capturado por el micrófono 23.

[0020] Como es bien conocido en la técnica, normalmente se usa una función de formación de ventana (tal como una ventana Hamming) para extraer las tramas 39 de muestras de la señal de audio entrante (D(t)) - para reducir las distorsiones introducidas por la extracción. Una vez que se ha extraído una trama 39 de muestras, la unidad de análisis de frecuencia 34 realiza un procedimiento de análisis de frecuencia en las muestras de audio para determinar el contenido de frecuencia dentro de una banda de frecuencia definida de interés, que normalmente será una parte de la banda de paso del filtro 51. En esta realización, esta banda de frecuencia de interés se limita a la banda de 475 Hz a 2,52 kHz. Por supuesto, se pueden usar otras bandas de frecuencia.

[0021] Como apreciarán los expertos en la materia, el procedimiento de análisis de frecuencia realizado por la unidad de análisis de frecuencia 34 se puede realizar de varias maneras diferentes, como mediante el uso de una Transformada Rápida de Fourier (FFT) o una Transformada de Coseno Discreta (DCT) o mediante el uso de transformadas de ondícula o incluso mediante el uso de una matriz de bancos de filtros. En la realización preferida se usan transformadas de ondículas. Este análisis de frecuencia generará, para cada trama 39 de muestras de audio, un vector de números, que representa el contenido de frecuencia (amplitud) en cada uno de un número (K) de sub-bandas de frecuencia dentro de la banda de frecuencia definida de interés (por ejemplo, 475 Hz a 2.52 kHz). Por lo tanto, como se muestra en la Figura 4, el análisis de frecuencia de la primera trama fi da como resultado la generación del vector de números $f_1^1, f_1^2, f_1^3 \ldots f_1^K$; donde el número f_1^1 representa el contenido de frecuencia en la primera subbanda de frecuencia de las muestras de audio en la primera trama, f_1^2 representa el contenido de frecuencia en la segunda subbanda de frecuencia de las muestras de audio en la primera trama, f_1^3 representa el contenido de frecuencia en la tercera subbanda de frecuencia de las muestras de audio en la primera trama, etc. El número de subbandas consideradas (es decir, el valor de K) depende de la potencia de procesamiento disponible del procesador 63 y la resolución de frecuencia requerida para extraer una huella significativa (distinguible). Los inventores han descubierto que un valor de K que está entre 25 y 50 subbandas produce buenos resultados para una banda de frecuencia de interés que tiene aproximadamente 2 kHz de ancho. De manera similar, el análisis de frecuencia del segundo cuadro f_2 da como resultado la generación del vector de números $f_2^1, f_2^2, f_2^3 \ldots f_2^K$; donde el número f_2^1 representa el contenido de frecuencia en la primera subbanda de las muestras de audio en la segunda trama, f_2^2 representa el contenido de frecuencia en la segunda subbanda de las muestras de audio en la segunda trama, f_2^3 representa el contenido de frecuencia en la tercera subbanda de las muestras de audio en la segunda trama, etc.

[0022] Como se ilustra en la Figura 4, el espectrograma 35 se forma concatenando los vectores generados a partir de la serie de tramas 39 extraídas de la señal de audio D(t). El número de tramas extraídas (y por lo tanto el tamaño (L) del espectrograma 35) depende de la duración del clip de audio entrante. Por lo general, se generará un espectrograma correspondiente a varios segundos de audio. Si el clip de audio es demasiado corto, entonces es más probable que la huella resultante coincida con múltiples entradas en la base de datos 15 y si es demasiado largo, esto aumentará los cálculos requeridos por el servidor de coincidencia de audio 5 para hacer coincidir la huella con las entradas en la base de datos 15. Para dar un ejemplo, con una velocidad de muestreo de audio de 8 kHz y si cada trama 39 tiene 1024 muestras de audio y con una trama 39 que se extrae cada 128 muestras de audio, entonces un clip de audio de ocho segundos dará como resultado que el tamaño del espectrograma 35 sea L = 500.

[0023] El espectrograma así generado es efectivamente una matriz K x L de valores que representan el clip de audio. Las filas de la matriz representan las diferentes sub-bandas de frecuencia y las diferentes columnas representan diferentes puntos de tiempo dentro del clip de audio. El valor individual en el espectrograma en la ubicación (i, j) corresponde a la amplitud del componente de frecuencia en la subbanda i en el tiempo j. Por supuesto, la matriz podría escribirse en la transpuesta, con las columnas representando las subbandas de frecuencia y las filas representando los puntos de tiempo. Por lo tanto, las referencias a filas y columnas en este documento son intercambiables.

Software de Aplicación: Generación de Huellas

[0024] Volviendo a la Figura 3, una vez que el espectrograma 35 se ha calculado y almacenado en el búfer de espectrograma 37, una unidad de generación de huellas 41 procesa el espectrograma 35 para generar una huella 43. La unidad de generación de huellas 41 genera la huella 43 aplicando un primer conjunto optimizado 45 de filtros al espectrograma 35 y binarizando el resultado. Hay muchas combinaciones de filtros posibles diferentes que se pueden usar para generar una huella y el primer conjunto optimizado 45 de filtros se ha encontrado a través de un procedimiento de optimización. Más adelante se describirá la forma en que se realiza este procedimiento de optimización. La forma en que se usa este primer conjunto optimizado 45 de filtros para generar la huella 43 se explicará ahora en detalle con referencia a las Figuras 5a a 5f.

[0025] La Figura 5a ilustra cinco tipos diferentes de filtro 47-1, 47-2, 47-3, 47-4 y 47-5 que se pueden aplicar a diferentes partes del espectrograma 35. Cada filtro 47 tiene una altura (H) y un ancho (W) que define el tamaño del filtro; y un desplazamiento (O) que define las subbandas de frecuencia del espectrograma 35 al que se aplicará el filtro 47. En esta realización, los coeficientes de cada filtro 47 suman cero. Por lo tanto, por ejemplo, el tipo de filtro 47-1 puede formarse a partir de la siguiente matriz de coeficientes:

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

Y el tipo de filtro 47-2 puede formarse a partir de la siguiente matriz de coeficientes:

$$[-1 \quad 2 \quad -1]$$

[0026] La Figura 5b ilustra la forma en que un filtro 47 (en este caso el filtro 47-1) se aplica al espectrograma 35 en un desplazamiento dado desde la base del espectrograma. Como se representa por la flecha 49, el filtro 47 está escalonado a través del eje de tiempo del espectrograma 35 y en cada etapa los coeficientes del filtro 47 se usan para realizar una combinación ponderada de los valores de frecuencia en la parte relevante del espectrograma 35. El número de valores de frecuencia que se combinan en cada etapa depende del tamaño (W, H) del filtro 47 y cómo se combinan depende de los coeficientes del filtro 47. El valor combinado de cada etapa se escribe a continuación en un vector de valores combinados que se cuantifica (o binariza) en "1"s y "0"s dependiendo de si los valores combinados son mayores o menores que cero.

[0027] Este procedimiento se ilustra con más detalle en las Figuras 5c a 5e. La Figura 5c ilustra la aplicación de un filtro de ejemplo (que es un filtro tipo 47-1) que tiene un tamaño de 2 por 2 y un desplazamiento de 10. Por lo tanto, el filtro 47-1 se aplica a los valores en el espectrograma 35 que están a 10 filas desde la parte inferior del espectrograma 35 (por supuesto, el desplazamiento se puede definir desde cualquier punto dentro del espectrograma 35). En la primera etapa, el primer bloque 32 de valores de amplitud del espectrograma 35 en el desplazamiento definido se combina multiplicando estos valores de amplitud por los coeficientes correspondientes en el filtro 47-1 y a continuación sumando los valores. Por lo tanto, como se muestra en la Figura 5c, en la primera etapa el valor de amplitud 6 se multiplica por el coeficiente de filtro -1; el valor de amplitud 4 se multiplica por el coeficiente de filtro 1; el valor de amplitud 3 se multiplica por el coeficiente de filtro 1; y el valor de amplitud 5 se multiplica por el coeficiente de filtro -1. Los cuatro números resultantes se suman para proporcionar un resultado combinado de -4. Este valor se escribe en el primer elemento de un vector 42 de valores combinados. Como se ilustra en la Figura 5d, el filtro 47-1 se pasa entonces a lo largo de una etapa de tiempo y se combina de una manera similar con el siguiente bloque 34 de valores de amplitud del espectrograma 35. Como se muestra, esta combinación da como resultado el valor 2, que se escribe en el siguiente elemento del vector 42. Este procedimiento se repite hasta que el filtro 47-1 se ha escalonado a través de la longitud (eje de tiempo) del espectrograma 35 y el vector resultante 42 tendrá, por lo tanto, L elementos correspondientes a la longitud temporal del espectrograma 35.

[0028] Los valores combinados en este vector 42 serán números positivos y negativos. Con el fin de simplificar la huella 43 que se genera (y, por lo tanto, reducir los datos necesarios para representar la huella), estos valores se cuantifican en valores binarios, por ejemplo, estableciendo todos los valores superiores a 0 en el valor binario "1" y estableciendo todos los valores inferiores a cero en el valor binario "0", como se muestra en la Figura 5e. El vector binarizado resultante 44 formará una fila de la huella 43 (que se muestra en la Figura 5f). Como apreciarán los expertos en la materia, este procedimiento de binarización se puede realizar en cada valor combinado a medida que se genera y se escribe directamente en el vector binarizado 44, en lugar de generar primero el vector intermedio 42.

[0029] Tal como se analizó anteriormente, la huella 43 se genera aplicando un primer conjunto optimizado 45 de estos filtros 47 al espectrograma 35. Cada filtro diferente 47 en este primer conjunto 45 producirá una fila diferente de la huella final 43. Por lo tanto, la concatenación de los diferentes vectores binarios 44 producidos mediante la aplicación del primer conjunto 45 de filtros al espectrograma 35 en una matriz, forma la huella 2D de salida final 43. El orden en que se concatenan los vectores binarios 44 se determina de antemano y se usa el mismo primer conjunto 45 de filtros y ordenación para generar las huellas correspondientes para las entradas en la base de datos 15, de modo que el servidor de coincidencia de audio 5 pueda comparar las huellas que se han generado de la misma manera. En esta realización, el primer conjunto 45 de filtros comprende treinta y dos filtros diferentes y, por lo tanto, la huella que se genera será una matriz de 32 por L de valores binarios. Se eligieron treinta y dos filtros, ya que esto permite el procesamiento conveniente de las huellas por un procesador de 32 bits o 64 bits (que se puede usar, por ejemplo, para realizar el procedimiento de coincidencia de huellas en el servidor de coincidencia de audio 5). Sin embargo, como apreciarán los expertos en la materia, se puede usar cualquier cantidad de filtros en el primer conjunto 45.

[0030] Además, como antes, las filas y columnas de la huella 43 son intercambiables. Por lo tanto, en lugar de que los vectores binarios 44 formen las filas de la huella 43, se pueden usar para formar las columnas. En este caso, la huella será una matriz L por 32 de valores binarios. Siempre que se realice el mismo procedimiento para generar las huellas para las entradas en la base de datos de audio 15, la orientación de la huella 43 no importa.

Orden de Filas/Columnas de Huellas

[0031] Tal como se analizó anteriormente, los vectores binarios 44 generados mediante la aplicación de los filtros 47 en el conjunto optimizado 45 al espectrograma 35 se concatenan entre sí en un orden que se define de antemano. Normalmente, el orden no importa, siempre que se aplique el mismo orden al generar las huellas para las entradas en la base de datos 15. Esto significa que en una huella convencional, el Is y el Os aparecerán distribuidos aleatoriamente a lo largo de la huella, como para el ejemplo de la huella 43-1 que se muestra en la Figura 6. Sin embargo, en esta realización, el ordenamiento se elige de una manera específica, en particular de una manera que maximice (o al menos aumente) la probabilidad de que los vectores binarios adyacentes 44 (es decir, las filas o columnas adyacentes) en la huella 43 sean similares entre sí. Como se explicará con más detalles más adelante, este ordenamiento específico se determina durante una etapa de entrenamiento donde se generan huellas para una gran colección de muestras de audio y se encuentra que el ordenamiento maximiza la probabilidad de que las filas/columnas adyacentes en la huella 43 sean similares. Esta ordenación específica se define dentro del software de aplicación 69 y controla la forma en que la unidad de generación de huellas 41 concatena los vectores binarios 44 para formar la huella 43. La Figura 6 también muestra un segundo ejemplo de huella 43-2 que se genera usando el pedido específico discutido anteriormente. Como se puede observar, la huella 43-2 es mucho menos aleatoria en apariencia que la huella 43-1, ya que muchos más bits adyacentes en la huella 43-2 tienen el mismo valor y, por lo tanto, se agrupan para definir islas más grandes del mismo valor binario. Como se explicará más adelante, esto es importante para que el servidor de audio remoto 5 pueda generar una huella gruesa a partir de la huella 43 que reducirá la carga de procesamiento para encontrar una entrada coincidente en la base de datos 15.

[0032] Como apreciarán los expertos en la materia, como el orden de los vectores binarios 44 se conoce de antemano, los valores binarizados individuales podrían escribirse directamente en la parte relevante de la huella 43 sin escribirse primero en un vector binario 44. La explicación anterior se ha dado para facilitar la comprensión de la forma en que se genera la huella 43.

Software de Aplicación - Respuesta de Coincidencia

[0033] Volviendo a la Figura 3, una vez que se ha generado la huella 43, el software de aplicación 69 pasa la huella 43 al procesador 63 para su transmisión al servidor de coincidencia de audio 5. Como el software de aplicación 69 está diseñado para trabajar con el servidor de coincidencia de audio 5, el software de aplicación 69 tendrá almacenada en él información de dirección para el servidor de coincidencia de audio 5 de modo que pueda enviar la huella al servidor de coincidencia de audio 5; ya sea a través de la red de telecomunicaciones 9 o a través de la red informática 13. El software de aplicación 69 pasará esta información de dirección y la huella generada 43 al procesador 63 solicitando que la huella 43 se envíe al servidor de coincidencia de audio remoto 5. A continuación, el procesador 63 enviará la huella 43 al servidor de coincidencia de audio 5 y espera un mensaje de respuesta. Cuando el mensaje de respuesta de coincidencia 46 se devuelve desde el servidor de coincidencia de audio 5 (ya sea a través de la red de telecomunicaciones 9 o a través de la red informática 13), el procesador 63 recibirá y pasará el mensaje de respuesta de coincidencia 46 al software de aplicación 69. El software de aplicación 69 toma entonces una acción apropiada según el contenido del mensaje de respuesta coincidente 46. Por ejemplo, si el mensaje de respuesta de coincidencia 46 simplemente proporciona detalles del audio capturado, como el nombre de la canción, el artista, etc., entonces el software de aplicación 69 puede emitir esta información al usuario, como a través de la pantalla 17 o el altavoz 25. Si el mensaje de respuesta coincidente 46 incluye un enlace para obtener más información o contenido relacionado con el audio capturado, entonces el software de aplicación 69 puede indicar al usuario si el usuario desea recuperar la información o el contenido del enlace proporcionado. En respuesta a que el usuario acepta la recuperación (por ejemplo, presionando una tecla 33 en el teclado 31), el software de aplicación 69 puede recuperar la información o el contenido del propio enlace o puede solicitar al software del navegador 68 que recupere la información o el contenido (qué información o contenido recuperado se envía al usuario, por ejemplo, en la pantalla 17). Si el software de aplicación 69 forma parte de una aplicación de sondeo de audiencia, entonces el software de aplicación 69 puede simplemente recopilar la información sobre el audio que se ha capturado (como el canal de televisión y el programa que se está viendo) y a continuación enviarla al servidor remoto de sondeo de audiencia 19 junto con un identificador del usuario que posee el teléfono (que puede ser solo un identificador del teléfono 1). Si el mensaje de respuesta de coincidencia 46 es un informe "nulo", que indica que no se han encontrado coincidencias, entonces el software de aplicación 69 puede emitir esta información al usuario.

Servidor de Coincidencia De Audio

[0034] La Figura 7 es un diagrama de bloques de los componentes principales del servidor de coincidencia de audio 5 usado en esta realización. Como se muestra, el servidor de coincidencia de audio 5 incluye un procesador (que puede ser una o más unidades de procesamiento central) 201 que se comunica con el dispositivo de usuario 1 a través de una interfaz de red 205 y la red de telecomunicaciones 9 o la red informática 13. El procesador 201 también se comunica con la base de datos 1543 a través de una interfaz de base de datos 207. En la práctica, las interfaces 205 y 207 pueden estar formadas por una única interfaz física, tal como una interfaz LAN o similar.

[0035] En esta realización, el procesador 201 está controlado por instrucciones de software almacenadas en la memoria 209 (aunque en otras realizaciones, el procesador 201 puede formarse a partir de uno o más procesadores de hardware dedicados, tales como Circuitos Integrados de Aplicación Específica). Las instrucciones de software

incluyen un sistema operativo 211 que controla el funcionamiento general del servidor de coincidencia de audio 5; un módulo de control de comunicaciones 213 que controla las comunicaciones entre el servidor de coincidencia de audio 5 y el dispositivo de usuario 1 y la base de datos 15; una unidad de generación de huella gruesa 215 que genera una huella gruesa a partir de la huella de consulta 43 recibida de un dispositivo de usuario 1; una unidad de coincidencia de huellas gruesas 217 que hace coincidir la huella gruesa generada por la unidad de generación de huellas gruesas 215 con huellas gruesas almacenadas en la base de datos 15; una unidad de coincidencia de huellas finas 219 que coincide con la huella 43 recibida del dispositivo de usuario 1 con una huella fina de un subconjunto de las entradas en la base de datos 15 para identificar una entrada coincidente; y una unidad de informe de respuesta de coincidencia 220 que informa los resultados de coincidencia al dispositivo de usuario 1 en un mensaje de respuesta de coincidencia 46. Como se explicará con más detalle a continuación, la unidad de generación de huellas gruesas 215 genera la huella gruesa usando un segundo conjunto 221 de filtros optimizados que se almacena en la memoria 209.

[0036] Tal como se analizó anteriormente, el servidor de coincidencia de audio 5 realiza operaciones de coincidencia entre huellas gruesas/finas correspondientes a una consulta recibida desde el dispositivo de usuario 1 y huellas gruesas/finas almacenadas dentro de la base de datos 15. Para distinguir entre estas diferentes huellas, la huella 43 recibida del dispositivo de usuario 1 se denominará "huella de consulta fina" 43 y la huella gruesa que se genera a partir de ella se denominará "huella de consulta gruesa". Las huellas almacenadas en la base de datos 15 se denominarán "huellas gruesas de la base de datos" y "huellas finas de la base de datos".

Conjunto de Generación De Huellas Gruesas

[0037] Como se mencionó anteriormente, en esta realización, la unidad de generación de huellas gruesas 215 genera la huella de consulta gruesa a partir de la huella de consulta fina 43 recibida del dispositivo de usuario 1. Esto es ventajoso ya que significa que el dispositivo de usuario 1 no necesita transmitir, por ejemplo, el espectrograma 35 del clip de audio al servidor de coincidencia de audio 5 para que se genere la huella de consulta gruesa.

[0038] La Figura 8 ilustra el procedimiento usado por la unidad de generación de huellas gruesas 215 para generar la huella de consulta gruesa. El procedimiento es muy similar al procedimiento usado para generar la huella de consulta fina (descrito anteriormente con referencia a la Figura 5), excepto que se usa un segundo conjunto (diferente) 221 de filtros optimizados y en este caso los filtros 47 de este segundo conjunto 221 se aplican a la huella de consulta fina 43 en lugar de al espectrograma 35. Además, en lugar de pasar cada filtro 47 en este segundo conjunto 221 de filtros optimizados sobre la huella de consulta fina 43 un punto de tiempo a la vez, se omiten varios puntos de tiempo en cada etapa, con el fin de reducir el tamaño de la huella de consulta gruesa. En esta realización, los filtros omiten 10 puntos de tiempo entre cada etapa y el número de filtros en el segundo conjunto 221 de filtros se mantiene igual que en el primer conjunto 45 (=32). El resultado es una huella compacta (gruesa) que permite una búsqueda inicial más rápida de las entradas en la base de datos 15. Por supuesto, las huellas gruesas podrían hacerse aún más compactas reduciendo el número de filtros 47 usados en el segundo conjunto 221 (en comparación con el número de filtros usados en el primer conjunto 45). Por lo tanto, en el ejemplo anterior, si la huella de consulta fina tiene una velocidad de bits (o resolución) de 2000 bits por segundo de la señal de audio capturada, entonces la huella de consulta gruesa tendrá una velocidad de bits (o resolución) de 200 bits por segundo de la señal de audio capturada.

[0039] La Figura 8a muestra la huella de consulta fina 43 escrita en formato de matriz y con “0” binarios escritos como el valor -1. Esto garantiza que el procedimiento de filtrado combine con precisión las diferentes partes de la huella de consulta fina 43 que se combinarán con el filtro. En este ejemplo, el filtro 47 que se aplica a la huella de consulta fina 43 es un filtro de tipo 47-2 que tiene una altura 2 y un ancho 3. El ancho (dimensión temporal) del filtro suele ser mayor que esto y se ha elegido el valor de 3 para simplificar la ilustración. El ancho del filtro es típicamente aproximadamente el doble de la tasa de diezmado temporal entre la huella fina y la huella gruesa. Entonces, si la velocidad de datos de la huella gruesa es 1/10 de la de la huella fina, la velocidad de diezmado es 10 y el ancho típico del filtro sería 20, aunque cada filtro puede tener un ancho diferente. Si el ancho es menor que la tasa de diezmado, por ejemplo, 4 en este ejemplo, entonces 6 de cada 10 de los valores de huella fina no tendrían impacto en los valores de huella gruesa. El filtro tiene un desplazamiento de 8, lo que significa que el filtro 47 se aplica a los elementos de la huella de consulta fina 43 que están a 8 filas de la parte inferior. El filtro 47 se aplica al primer bloque 301 de valores en la huella de consulta fina 43 (correspondiente al punto de tiempo t=1). El tamaño de este bloque 301 coincide con el del filtro 47, de modo que cada valor en el bloque 301 tiene un coeficiente de filtro correspondiente con el que se multiplica, como se muestra. Los valores multiplicados se suman para obtener el valor 2, que se escribe en el primer elemento del vector 303. En la siguiente etapa, el filtro 47 se mueve a lo largo del eje de tiempo de la huella de consulta fina 43, pero esta vez omitiendo algunos de los elementos de la huella de consulta fina 43. En este ejemplo, el filtro se omite a lo largo de 10 elementos (puntos de tiempo); lo que significa que la huella de consulta gruesa que se genera tendrá 1/10º de la longitud temporal de la huella de consulta fina 43.

[0040] La Figura 8b ilustra que el siguiente bloque 305 de valores (en el tiempo t=11) en la huella de consulta fina 43 se combina de la misma manera con los coeficientes de filtro para generar el valor -2 que se escribe en el siguiente lugar en el vector 303. Una vez que el filtro 47 se ha escalonado a lo largo de la huella de consulta fina 43 y el vector 303 se ha llenado, el vector 303 se binariza (como se muestra en la Figura 8c) para producir un vector binario 307. Como antes, esta binarización se puede realizar en el momento en que se genera cada valor combinado y a

continuación se escribe directamente en el vector binario 307 sin usar el vector 303 (o se escribe directamente en la huella de consulta gruesa). El vector binario 307 que se produce formará una fila (o una columna) de la huella de consulta gruesa. El mismo procedimiento se repite para todos los diferentes filtros 47 en el segundo conjunto 221 de filtros optimizados y los vectores binarios resultantes 307 se concatenan para formar la huella de consulta gruesa 309 (que se muestra en la Figura 8d). De nuevo, siempre que los vectores binarios 307 se concatenen en el mismo orden que se usó para concatenar los vectores binarios que se generaron para formar las huellas gruesas de la base de datos, entonces el servidor de coincidencia de audio 5 puede realizar una coincidencia adecuada entre la huella gruesa de consulta 309 y las huellas gruesas de la base de datos.

[0041] Los inventores han descubierto que la huella gruesa generada de esta manera sigue siendo lo suficientemente distintiva como para permitir su uso en una búsqueda inicial de la base de datos 15 con el fin de reducir significativamente el número de entradas de la base de datos que deben compararse con la huella de consulta fina. Esto se debe a la orden específica que se realizó para generar la huella de consulta fina. Este ordenamiento significa que hay patrones de bits más grandes (que contienen información) en la huella fina y la información contenida en estos patrones de bits más grandes sobrevive en cierta medida a través del procedimiento de generación de las huellas gruesas correspondientes. Si se usa una huella fina más tradicional (de aspecto aleatorio) (como la huella 43-1 que se muestra en la Figura 6), es probable que el procedimiento anterior de generación de una huella gruesa a partir de la huella fina resulte en la pérdida de la mayor parte de la información contenida en la huella fina. Esto significa que la huella gruesa no será distintiva y, por lo tanto, cuando se compara con otras huellas gruesas similares, es probable que muchas se consideren una posible coincidencia. Esto puede frustrar el propósito de generar la huella gruesa, ya que la huella de consulta fina aún tendrá que coincidir con una gran cantidad de entradas de base de datos potencialmente coincidentes.

Unidad de Coincidencia de Huellas Gruesas

[0042] Una vez que se ha generado la huella de consulta gruesa 309, la unidad de coincidencia de huella gruesa 217 compara la huella de consulta gruesa 309 con las huellas de base de datos gruesa almacenadas en la base de datos 15 para identificar un subconjunto de las entradas de base de datos que puede ser una posible coincidencia.

[0043] La Figura 9 ilustra la estructura general de las entradas de la base de datos 320. Cada entrada tiene un identificador: DB#1 para la primera entrada, DB#2 para la segunda entrada, etc. Como se muestra en la Figura 9, hay D entradas en la base de datos 15. El valor de D puede ser muy grande dependiendo de la aplicación. Si el sistema de coincidencia de audio es para su uso en un tipo de servicio "Shazam®", entonces D puede ser del orden de 10 a 20 millones. Sin embargo, si el sistema de coincidencia de audio es parte de un sistema de encuesta de audiencia que está diseñado para identificar el programa y el canal que el usuario está viendo, entonces el número de entradas (D) será mucho menor, aunque los clips de audio (o al menos las huellas que los representan) almacenados en la base de datos 15 serán mucho más largos. Por ejemplo, un sistema diseñado para monitorear las transmisiones de televisión realizadas durante los 30 días anteriores en 1000 canales de televisión contendrá aproximadamente 720.000 horas de contenido, lo que es equivalente en tamaño a un sistema tipo Shazam® con 10 millones de canciones.

[0044] Como se muestra en la Figura 9, cada entrada generalmente incluirá:

- el contenido de audio y/o vídeo 321 (aunque esto no es estrictamente necesario);
- metadatos 322 para ese contenido (como el nombre de la canción, artista, programa de televisión, canal de televisión, hora de emisión, director, etc.);
- una huella de base de datos fina 323 que se genera a partir del audio en el contenido de la misma manera que se genera la huella de consulta fina 43;
- una huella de base de datos gruesa 325 para el audio en el contenido que se genera a partir de la huella fina 323 de la misma manera que la huella de consulta gruesa 309 se generó a partir de la huella de consulta fina 43 (como se explicó anteriormente);
- Enlaces 327 y otra información relacionada con el contenido de audio y/o vídeo.

[0045] Por lo tanto, una vez que se ha generado la huella de consulta gruesa 309, la unidad de coincidencia de huellas gruesas 217 coincide (es decir, compara) la huella de consulta gruesa 309 con la huella de base de datos gruesa 325 almacenada en cada entrada 320 de la base de datos 15; para identificar una serie de posibles entradas coincidentes. Este procedimiento de coincidencia está ilustrado en la Figura 10. En particular, la Figura 10a ilustra todas las huellas gruesas de la base de datos 325. La huella gruesa de la base de datos para la primera entrada de la base de datos está etiquetada como 325-DB #1, la huella gruesa de la base de datos para la segunda entrada de la base de datos está etiquetada como 325-DB #2, la huella gruesa de la base de datos para la tercera entrada de la base de datos está etiquetada como 325-DB #3, etc. La Figura 10a ilustra que estas huellas gruesas de la base de datos 325 tienen diferentes longitudes temporales. Esto se debe a que generalmente se generan a partir de contenido de audio que tiene diferentes duraciones.

[0046] La Figura 10a también ilustra la huella de consulta gruesa 309 que se va a comparar con cada una de

estas huellas de base de datos gruesa 325. Típicamente (y como se muestra en la Figura 10a), la huella de consulta gruesa 309 tiene una duración mucho más corta que las huellas de base de datos gruesa 325. Para hacer coincidir la huella de consulta gruesa 309 con una huella de base de datos gruesa 325, la huella de consulta gruesa 309 se "avanza" a lo largo de la huella de base de datos gruesa más larga 325 de principio a fin. En cada etapa, se realiza una comparación por bits entre los bits en la huella de consulta gruesa 309 y los bits en una porción de tamaño correspondiente de la huella de base de datos gruesa 325. Como es bien sabido, esta comparación por bits se puede realizar usando un tipo de combinación XOR de los bits de las dos huellas, lo que resulta en un recuento del número de diferencias de bits entre las dos. Por lo tanto, este recuento representa la similitud entre la huella de consulta gruesa 309 y la porción actual de la huella de base de datos gruesa 325. La huella de consulta gruesa 309 se escalona a lo largo del eje temporal y se compara de una manera similar con la siguiente porción de la huella de base de datos gruesa 325. Por lo general, la huella de consulta gruesa 309 se desplaza a lo largo de un punto de tiempo a la vez en la huella de base de datos gruesa 325.

[0047] En esta realización, la comparación por bits considera el porcentaje de bits no coincidentes. Por lo tanto, si no hay una coincidencia, entonces el porcentaje esperado de bits no coincidentes debe ser de alrededor del 50 % (o 0,5). Si hay una coincidencia, entonces el porcentaje esperado de bits no coincidentes debe ser cercano a cero. La Figura 10b ilustra el resultado de este procedimiento de coincidencia cuando la huella de consulta gruesa 309 no coincide con ninguna porción de una huella de base de datos gruesa 325; y la Figura 10c ilustra el resultado de este procedimiento de coincidencia cuando la huella de consulta gruesa 309 coincide con una porción de la huella de base de datos gruesa 325 (identificada por el pico 326 en el porcentaje de bits no coincidentes). Por lo tanto, si la huella de consulta gruesa 309 coincide o no con una porción de la huella de base de datos gruesa 325 se determina comparando los porcentajes calculados con un nivel de umbral (por ejemplo, 10 %). Por lo tanto, si el porcentaje de bits no coincidentes cae por debajo de este umbral, entonces hay una coincidencia, si no lo hace, entonces no hay coincidencia. Como apreciarán los expertos en la materia, se podrían usar otras métricas de puntuación en su lugar.

[0048] El resultado de comparación para la comparación entre la huella de consulta gruesa 309 y las huellas de base de datos gruesas 325 incluye una lista de entradas de base de datos 320 que podrían coincidir con la huella de consulta gruesa 309. En la Figura 10a, esta lista de posibles entradas coincidentes incluye las entradas DB#10, DB#15, DB#260 y DB#500. Los resultados de comparación también pueden incluir opcionalmente información de tiempo que identifica qué porción(es) dentro de la huella de base de datos gruesa 325 coincide(n) con la huella de consulta gruesa 309. Por lo tanto, por ejemplo, en la coincidencia mostrada en la Figura 10c, la información de tiempo puede indicar que la coincidencia se encontró alrededor de 135 segundos desde el inicio de la canción representada por la huella de la base de datos gruesa 325. Si se proporciona, esta información de tiempo se puede usar para reducir aún más la comparación entre la huella de consulta fina 43 y las huellas de base de datos fina 323 correspondientes.

Unidad de Coincidencia de Huellas Finas

[0049] Los resultados de comparación obtenidos de la unidad de coincidencia de huellas gruesas 217 se pasan a continuación a la unidad de coincidencia de huellas finas 219 que usa esta información para restringir la operación de coincidencia que realiza entre la huella de consulta fina 43 y las huellas de base de datos fina 323. En particular, la unidad de coincidencia de huellas finas 219 usa la lista de posibles entradas de coincidencia de modo que las comparaciones de huellas finas se restringen solo a las huellas finas en las entradas de base de datos identificadas en esta lista de posibles entradas de coincidencia. Además, si los resultados de la comparación incluyen información de tiempo que indica el tiempo dentro del contenido de audio donde se encontró la coincidencia en la huella gruesa de la base de datos 325, entonces la unidad de coincidencia fina de huellas 219 usa esta información de tiempo para restringir la comparación entre la huella fina de consulta 43 y la huella fina de la base de datos 323 correspondiente a alrededor de este tiempo. Entonces, por ejemplo, si la coincidencia se encontró a 135 segundos desde el inicio de la huella gruesa 325, entonces la unidad de coincidencia fina 219 puede restringir el procedimiento de coincidencia de modo que la huella de consulta fina 43 solo coincida con las porciones de la huella de base de datos fina entre los tiempos 130 y 145 segundos desde el inicio.

[0050] La Figura 11 ilustra el procedimiento de coincidencia que se realiza entre la huella de consulta fina 43 y cada una de las huellas de base de datos fina 323-DB #10, 323-DB #15, 323-DB#260 y323-DB#500 (cuando dicha información de tiempo no está disponible). Como se ilustra mediante la flecha 313, el procedimiento de coincidencia pasa la huella de consulta fina 43 a lo largo de cada una de estas huellas de base de datos fina, de la misma manera que la huella de consulta gruesa 309 se pasó a lo largo de las huellas de base de datos gruesa 325. En cada etapa, se realiza una comparación similar por bits entre la huella de consulta fina 43 y la porción correspondiente de la huella de base de datos fina 323 para determinar el porcentaje de bits no coincidentes. La unidad de coincidencia de huella fina 219 usa el porcentaje determinado de bits no coincidentes para determinar si hay una coincidencia, de nuevo comparando el porcentaje determinado de bits no coincidentes con un umbral. Si la unidad de coincidencia de huellas finas 219 identifica una única entrada de base de datos como una coincidencia, entonces informa el identificador para la entrada de base de datos coincidente (por ejemplo, DB#260) a la unidad de informe de respuesta coincidente 220. Sin embargo, si la unidad de coincidencia de huellas finas 219 identifica más de una coincidencia posible, entonces compara el porcentaje de bits no coincidentes para cada coincidencia sospechosa para identificar qué entrada de base de datos tiene el porcentaje más pequeño de bits no coincidentes; y a continuación informa este como el resultado de

la coincidencia a la unidad de informes de respuesta de coincidencia 220. Si, por otro lado, ninguna de las posibles huellas finas de la base de datos coincidentes coincide realmente con la huella fina de consulta 43, entonces la unidad de coincidencia fina de huellas 219 puede devolver un resultado "nulo" a la unidad de informe de respuesta de coincidencia 220 o puede realizar una coincidencia completa entre la huella fina de consulta 43 y todas las demás huellas finas de la base de datos 323 que se excluyeron del procedimiento de coincidencia fina original debido a los resultados del procedimiento de coincidencia gruesa.

Unidad de Informes de Respuestas Coincidentes

[0051] La unidad de informe de respuesta de coincidencia 220 recibe un informe "nulo" o el identificador para la entrada de base de datos 320 que coincide con la huella de consulta fina. Si se recibe un informe "nulo", entonces la unidad de informe de respuesta coincidente 220 devuelve una respuesta "nula" al dispositivo de usuario 1. Si se recibe un identificador de base de datos, entonces la unidad de informes de respuesta coincidente 220 recupera información relevante de la entrada de base de datos correspondiente 320. La información recuperada puede incluir los metadatos almacenados 322 y/o enlaces almacenados 327 de la entrada de base de datos identificada 320. Esta información se devuelve al dispositivo de usuario 1 en un mensaje de respuesta coincidente 46.

Entrenamiento

Identificación del Primer Conjunto Optimizado de Filtros

[0052] La descripción anterior describe el funcionamiento de un sistema de coincidencia de audio que usa huellas de audio para identificar el audio capturado. Con el fin de generar la huella fina 43, se aplicó un primer conjunto 45 de filtros optimizados al espectrograma 35 del audio capturado. La forma en que se determina este primer conjunto 45 de filtros optimizados se explicará ahora. Este procedimiento ocurre de antemano durante una rutina de entrenamiento.

[0053] Como se discutió anteriormente con referencia a la Figura 5a, en esta realización, hay cinco tipos diferentes de filtro 47 que se pueden usar. Cada filtro 47 puede variar en altura y ancho. En el sistema de ejemplo explicado anteriormente, el espectrograma 35 tenía treinta y dos subbandas de frecuencia, por lo que la altura puede tener un valor de 1 a 32. Si bien el ancho podría ser, en teoría, cualquier valor hasta la longitud total del espectrograma 35, por simplicidad, también se permite que el ancho tenga un valor entre 1 y 32. Es posible aplicar cada filtro 47 en cualquier parte del espectrograma 35, es decir, puede tener cualquier valor de compensación entre 1 y 31. Tenga en cuenta, sin embargo, que algunos filtros siempre deben tener un ancho que sea un múltiplo de dos, y algunos deben tener un ancho que sea un múltiplo de tres para garantizar la simetría. Además, un filtro que tenga un valor de desplazamiento de 10 solo puede tener, como máximo, una altura de 22. Teniendo en cuenta todas estas limitaciones, el número total de filtros posibles (Nf) es 3x16x8x32 + 2x10x6x32=16.128 filtros. El procedimiento de entrenamiento descrito a continuación permite identificar un conjunto (combinación) óptimo de filtros 47 sin tener que considerar todas las combinaciones posibles ($16.128^{32} = 4x10^{134}$ combinaciones).

[0054] La Figura 12 ilustra parte del procedimiento de entrenamiento. Como se muestra, el procedimiento usa una base de datos 351 de clips de audio originales y una base de datos 353 de clips de audio distorsionados. Los clips de audio distorsionados en la base de datos 353 son versiones distorsionadas de los clips de audio originales en la base de datos 351. Las distorsiones incluyen las que normalmente se encuentran a través de la transmisión de los clips de audio originales a través de un canal de comunicaciones (que incluye un canal acústico). Por lo tanto, la versión distorsionada podría representar el audio después de que se haya emitido como una señal de sonido y haya sido captada por el micrófono de un dispositivo de usuario. Como se describirá a continuación, el procedimiento de entrenamiento aplica cada uno de los aproximadamente 16.000 filtros posibles 47 a un conjunto de pares coincidentes de las bases de datos 351 y 353 y también a un conjunto de pares no coincidentes de las bases de datos 351 y 353 y usa los resultados para identificar un conjunto óptimo de filtros que generarán huellas distintivas.

[0055] La Figura 12a ilustra la parte inicial del procedimiento de entrenamiento donde un filtro actual en consideración (filtro F(i)) se aplica a un par de clips de audio coincidentes y la Figura 12b ilustra el mismo procedimiento pero cuando el filtro (F(i)) en consideración se aplica a un par de clips de audio no coincidentes. En este contexto, un par coincidente de clips de audio incluye el clip de audio original de la base de datos 351 y la versión distorsionada correspondiente de ese clip de audio original de la base de datos 353; y un par de clips de audio no coincidentes incluye un clip de audio original de la base de datos 353 y una versión distorsionada de un clip de audio original diferente de la base de datos 353.

[0056] Con referencia a la Figura 12a, en la etapa s1, el clip de audio original se lee de la base de datos 351 y en la etapa s3 el clip de audio distorsionado (coincidente) correspondiente se lee de la base de datos 353. En la etapa s5 se determina un espectrograma 357 para el clip de audio original y en la etapa s7 se determina un espectrograma 359 para el clip de audio distorsionado. Estos espectrogramas se determinan de la manera descrita anteriormente con referencia a la Figura 4. En la etapa s9, el filtro actual en consideración (F(i)) se aplica al espectrograma 357 y el resultado se binariza para generar el vector binario 361. De manera similar, en la etapa s11 se aplica el mismo filtro

(F(i)) al espectrograma 359 y el resultado se binariza para generar el vector binario 363. En la etapa s13, se realiza una comparación por bits entre los vectores 361 y 363 para determinar el número de bits no coincidentes. Esto se puede lograr usando una simple comparación XOR entre los dos vectores. En la etapa s15, el número determinado de bits no coincidentes se normaliza mediante la longitud del espectrograma (L), para tener en cuenta las diferentes longitudes de los pares coincidentes de clips de audio almacenados en las bases de datos 351 y 353, para generar un valor $B_M(i)$ que define efectivamente el porcentaje de bits no coincidentes (es decir, la tasa de error de bits) entre el par coincidente de clips de audio.

[0057] Como se puede observar a partir de la Figura 12b, se lleva a cabo un procedimiento muy similar para un par de clips de audio no coincidentes tomados de las bases de datos 351 y 353. En la etapa s21, se lee un clip de audio original de la base de datos 351 y en la etapa s23 se lee un clip de audio distorsionado no coincidente de la base de datos 353. Si los dos clips de audio no coincidentes tienen duraciones diferentes, entonces la duración del clip más largo se puede truncar para que coincida con la del clip más corto. En la etapa s25 se determina un espectrograma 365 para el clip de audio original y en la etapa s27 se determina un espectrograma 367 para el clip de audio distorsionado no coincidente. Estos espectrogramas se determinan de la manera descrita anteriormente con referencia a la Figura 4. En la etapa s29, el filtro en consideración (F(i)) se aplica al espectrograma 365 y el resultado se binariza para generar el vector binario 369. De manera similar, en la etapa s31 se aplica el mismo filtro (F(i)) al espectrograma 367 y se binariza el resultado para generar el vector binario 371. En la etapa s33, se realiza una comparación por bits entre los vectores 369 y 371 para determinar el número de bits no coincidentes. Como antes, esto se puede lograr usando una simple comparación XOR entre los dos vectores. En la etapa s35, el número determinado de bits no coincidentes se normaliza por la longitud del espectrograma (L), para tener en cuenta las diferentes longitudes de los pares coincidentes de clips de audio almacenados en las bases de datos 351 y 353, para generar un valor $B_N(i)$ que define efectivamente el porcentaje de bits no coincidentes (es decir, la tasa de error de bits) para el par no coincidente de clips de audio.

[0058] El procedimiento ilustrado en la Figura 12a se lleva a cabo usando el mismo filtro (F(i)) en cada uno de un número (N_M - por ejemplo 100) de pares coincidentes de clips de audio; y el procedimiento ilustrado en la Figura 12b se lleva a cabo usando el mismo filtro (F(i)) en cada uno de un número (N_N - que también puede ser 100) de pares no coincidentes de clips de audio. Si los valores de N_M así obtenidos para $B_M(i)$ y los valores de N_N así obtenidos para $B_N(i)$ se representan en un histograma, cada uno de ellos exhibirá una distribución normal que se caracteriza por un valor medio y una varianza. Si el filtro actual en consideración es un buen candidato de filtro, entonces la distribución para los pares coincidentes y la distribución para los pares no coincidentes deben estar bastante bien separadas entre sí, como las distribuciones de ejemplo 401 y 403 que se muestran en la Figura 13a. La distribución 401 es la distribución obtenida para los pares coincidentes y la distribución 403 es la distribución para los pares no coincidentes. Mientras que, si el filtro en consideración es un candidato deficiente, entonces la distribución para los pares coincidentes y la distribución para los pares no coincidentes estarán más cerca entre sí y posiblemente se superpondrán, como las distribuciones de ejemplo 405 y 407 que se muestran en la Figura 13b.

[0059] Desafortunadamente, no es posible determinar solo las distribuciones para todos los 16.000 filtros posibles y a continuación elegir los que tienen la mejor discriminación (separación entre las distribuciones coincidentes y no coincidentes y las variaciones más pequeñas, etc.), ya que muchos de los filtros aislarán efectivamente el mismo rasgo característico en la señal de audio. Es decir, muchas de las distribuciones de los diferentes filtros estarán altamente correlacionadas entre sí. Es posible identificar estas correlaciones observando la covarianza entre las distribuciones de filtros y usar esta información en un procedimiento de optimización para encontrar la combinación óptima de filtros. El objetivo de esa optimización puede ser minimizar la posibilidad de "falsos positivos" (declarar falsamente un par como "coincidente") y minimizar la posibilidad de falsos negativos (declarar falsamente un par como "no coincidente"), cuando la huella generada se compara con las huellas de la base de datos. Estas son demandas contradictorias ya que, en general, reducir la posibilidad de falsos positivos aumenta la posibilidad de falsos negativos. Para abordar esto, podemos definir una cierta tasa aceptada de falsos positivos ($P_{FP,aceptar}$) y a continuación, sujeto a esta restricción, podemos encontrar el conjunto óptimo de filtros que minimice la tasa de falsos negativos.

[0060] Para calcular $P_{FP,aceptar,}$ tenga en cuenta que la distribución resultante de un conjunto de filtros es la suma de distribuciones normales y, por lo tanto, una distribución normal en sí misma. Por lo tanto, si las distribuciones para los pares coincidentes y no coincidentes están bien separadas (como la que se muestra en la Figura 13a), entonces se puede definir un umbral (γ) entre las dos distribuciones que se puede usar para definir si un par de clips de audio coinciden o no coinciden. En particular, para un par dado de clips de audio, si la tasa de error de bits entre ellos es menor que el umbral (es decir, $B<\gamma$), entonces se puede suponer que el par es un par coincidente; mientras que si la tasa de error de bits determinada está por encima del umbral (es decir, $B>\gamma$), entonces se puede suponer que el par es un par no coincidente.

[0061] La posibilidad de un falso positivo se basa en la posibilidad de que la tasa de error de bits de un par no coincidente caiga por debajo del umbral (γ), que, para una distribución normal, viene dada por:

$$P(B_N < \gamma) = \frac{1}{2}\text{erfc}\left(\frac{\mu_N - \gamma}{\sqrt{2}\,\sigma_N}\right)$$

Donde μ_N es la tasa media de error de bits para un par de huellas no coincidentes, σ_N es la desviación estándar de la tasa de error de bits para un par de huellas no coincidentes y erfc es la función de error complementaria estándar.

[0062] Cuando una huella se compara con una gran base de datos de huellas, la probabilidad de un falso positivo depende del tamaño de la base de datos (D) y puede aproximarse como:

$$P(\text{Falso positivo}) \approx D \times P(B_N < \gamma) = \frac{D}{2}\text{erfc}\left(\frac{\mu_N - \gamma}{\sqrt{2}\,\sigma_N}\right) = P_{\text{FP,aceptar}}$$

que se establece en la tasa de aceptación. Esta ecuación se puede invertir para encontrar el valor umbral correspondiente que logrará esta tasa aceptada de falsos positivos:

$$\gamma = \mu_N - \sqrt{2}\sigma_N \text{erfc}^{-1}\left(2P_{\text{FP, aceptar}}/D\right)$$

[0063] Por lo tanto, la tasa de falsos negativos ahora se puede minimizar (para maximizar así la tasa de reconocimiento), minimizando la posibilidad de un falso negativo, dado que el umbral se establece como se indicó anteriormente. El resultado es:

$$P(B_M > \gamma) = \frac{1}{2}\text{erfc}\left(\frac{\gamma - \mu_M}{\sqrt{2}\,\sigma_M}\right) = \frac{1}{2}\text{erfc}\left(\frac{\mu_N - \mu_M}{\sqrt{2}\,\sigma_M} - \frac{\sigma_N}{\sigma_M}\text{erfc}^{-1}\left(2P_{\text{FP, aceptar}}/D\right)\right)$$

Donde μ_M es la tasa media de errores de bits para un par de huellas coincidentes, σ_M es la desviación estándar de la tasa de errores de bits para un par de huellas coincidentes, μ_N es la tasa media de errores de bits para un par de huellas no coincidentes, σ_N es la desviación estándar de la tasa de errores de bits para un par de huellas no coincidentes y erfc es la función estándar de errores complementarios

[0064] Dado que la función de error complementario es una función monótonamente decreciente, minimizar la posibilidad de un falso negativo, es decir, minimizar la función anterior, es equivalente a maximizar el argumento de la función de error complementario, aquí llamada la primera 'Puntuación'

$$S^{(1)} = \frac{\mu_N - \mu_M}{\sqrt{2}\,\sigma_M} - \frac{\sigma_N}{\sigma_M}\text{erfc}^{-1}\left(2P_{\text{FP, aceptar}}/D\right)$$

[0065] Por lo tanto, el objetivo del procedimiento de optimización es encontrar el conjunto 45 de filtros con parámetros agregados $(\mu_M, \mu_N, \sigma_M, \sigma_N)$ que dan como resultado la puntuación más alta $S^{(1)}$.

[0066] Estos parámetros agregados sobre el conjunto 45 de filtros están relacionados con los parámetros individuales de los filtros individuales en el conjunto 45 de la siguiente manera:

$$\mu_M = \frac{1}{n}\sum_{l=0}^{n-1}\mu_M(l) \quad y \quad \mu_N = \frac{1}{n}\sum_{l=0}^{n-1}\mu_N(l)$$

[0067] Donde n es el número de filtros en el conjunto 45. La varianza agregada (cuadrado de la desviación estándar) se convierte en una combinación de las varianzas de los filtros individuales que pertenecen al conjunto 45, así como la covarianza entre pares de filtros en el conjunto 45, de la siguiente manera:

$$\sigma_M^2 = \frac{1}{n^2}\sum_{l=0}^{n-1}(\sigma_M(l))^2 + \frac{1}{n^2}\sum_{l=0}^{n-1}\sum_{k=0,k\neq l}^{n-1} COV_M^{(l,k)}$$

$$\sigma_N^2 = \frac{1}{n^2}\sum_{l=0}^{n-1}(\sigma_N(l))^2 + \frac{1}{n^2}\sum_{l=0}^{n-1}\sum_{k=0,k\neq l}^{n-1} COV_N^{(l,k)}$$

Donde $COV^{(l,k)}$ es la covarianza entre el filtro l y el filtro k.

[0068] Las medias y las varianzas para los filtros individuales para los pares de clips de audio coincidentes y no coincidentes se pueden determinar a partir del procedimiento de entrenamiento analizado anteriormente con referencia a las Figuras 12a y 12b. En particular, la tasa media de errores de bits para los N_M pares coincidentes de clips de audio y la tasa media de errores de bits para los N_N pares no coincidentes de clips de audio para cada filtro (i) se pueden determinar de la siguiente manera:

$$\mu_M(i) = \frac{1}{N_M}\sum_{k=1}^{N_M} B_{M,k}(i) \quad \text{y} \quad \mu_N(i) = \frac{1}{N_N}\sum_{k=1}^{N_N} B_{N,k}(i)$$

[0069] Y las variaciones correspondientes de:

$$(\sigma_M(i))^2 = \frac{1}{N_M - 1}\sum_{k=1}^{N_M}\left(B_{M,k}(i) - \mu_M(i)\right)^2 \quad \text{y} \quad \left(\sigma_N^{(i)}\right)^2 = \frac{1}{N_N - 1}\sum_{k=1}^{N_N}\left(B_{N,k}(i) - \mu_N(i)\right)^2$$

[0070] Además, el valor de covarianza $(COV_M^{(i,j)})$ entre dos filtros (i y j) para emparejar pares de clips de audio se puede determinar a partir de:

$$COV_M^{(i,j)} = \frac{1}{N_M - 1}\sum_{k=1}^{N_M}\left(B_{M,k}(i) - \mu_M(i)\right)\left(B_{M,k}(j) - \mu_M(j)\right)$$

[0071] Y el valor de covarianza $(COV_N^{(i,j)})$ entre dos filtros (i y j) para pares de clips de audio no coincidentes se puede determinar a partir de:

$$COV_N^{(i,j)} = \frac{1}{N_N - 1}\sum_{k=1}^{N_N}\left(B_{N,k}(i) - \mu_N(i)\right)\left(B_{N,k}(j) - \mu_N(j)\right)$$

[0072] Esto implica el cálculo y almacenamiento de N_f (número de filtros considerados, que como se ha analizado anteriormente es aproximadamente 16.000) valores de $(\mu_M(i), \sigma_M(i))$; N_f valores de $(\mu_N(i), \sigma_N(i))$; y (la parte dominante) $2(N_f)^2$ valores de covarianza. A partir de estos valores, es posible calcular la puntuación anterior

$S^{(1)}$ para cualquier combinación de filtros.

[0073] No es práctico calcular esta puntuación para cada combinación de n filtros de este conjunto de 16000 filtros posibles; el número de combinaciones lo prohíbe. Sin embargo, es posible usar una técnica de programación dinámica para dividir este problema en un problema de búsqueda de ruta iterativa a través de un enrejado de nodos que se propaga y puntúa las rutas a través del enrejado de nodos. Esto significa que la ruta óptima se puede encontrar a través del enrejado sin tener que considerar y puntuar todas las rutas.

[0074] Dicho enrejado 409 se ilustra en la Figura 14. En particular, los filtros N_f están ordenados verticalmente y representados por un nodo 411 respectivo en la columna izquierda del enrejado 409. Esta columna de nodos 411 se repite n veces para que haya n columnas de nodos 411, donde n es el tamaño del conjunto de filtros a crear. Tal como se analizó anteriormente, en esta realización, n se establece en el valor 32 ya que esto facilita el cálculo usando una unidad de procesamiento central (CPU) de 32 bits o 64 bits. Las conexiones (bordes) de cada nodo en la columna de la izquierda a cada nodo en la siguiente columna se realizan y califican usando la puntuación anterior ($S^{(1)}$) que se debe maximizar. A continuación, se repite el mismo procedimiento con conexiones que se realizan desde la segunda columna a la tercera columna y se calculan nuevas puntuaciones. Dado que la única dirección permitida a través del enrejado es de izquierda a derecha, las mejores rutas de puntuación en cualquier columna se pueden usar para determinar las mejores rutas de puntuación en la siguiente columna. Esto significa que no se deben considerar todas las combinaciones posibles de filtros, ya que la mejor solución se puede construir de forma iterativa. Una vez que este procedimiento ha alcanzado la columna de la derecha, la ruta que tiene la puntuación máxima $S^{(1)}$ a través del enrejado 409 identifica el conjunto óptimo 45 de filtros. Por ejemplo, la trayectoria que tiene la puntuación máxima $S^{(1)}$ se ilustra como la trayectoria 415 que se muestra en negrita en la Figura 14. Esta ruta comienza en el nodo correspondiente al filtro F(2), a continuación atraviesa el nodo correspondiente al filtro F(3) y a continuación a F(1), F(4) y F(6); y finalmente terminando en el nodo F(7). Estos son los filtros 47 que forman el primer conjunto optimizado 45 de filtros usados por la unidad de generación de huellas 41.

[0075] Una de las ventajas de usar una técnica de programación dinámica para encontrar la mejor ruta a través del enrejado es que las puntuaciones de cada ruta se pueden acumular durante el procedimiento de recorrido de la ruta. Específicamente, considerando la instancia donde una ruta candidata termina actualmente en el nodo *q* en el número de columna *K* en el enrejado (es decir, se han seleccionado filtros *K* hasta ahora), que representa un conjunto de filtros *l* = 1,2,... *K*. En este caso, tenga en cuenta que las medias agregadas μ_M y μ_N se pueden actualizar hacia el nodo *r* en la columna *K+1,* añadiendo las medias del nodo *r,* $\mu_M(r)$ y $\mu_N(r)$, es decir:

$$\mu_M^r = \frac{K\mu_M^q + \mu_M(r)}{K+1} \quad \text{y} \quad \mu_M^r = \frac{K\mu_N^q + \mu_N(r)}{K+1}$$

donde μ_M^q y μ_N^q son las medias agregadas en el nodo *q,* combinando filtros *l* = 1,2,... *K* (es decir, en la columna *K)* y μ_M^r y μ_M^r son las medias agregadas en el nodo *r.* Del mismo modo, las varianzas $(\sigma_M)^2$ y $(\sigma_N)^2$ se pueden actualizar de la columna K a la columna K+1 de la siguiente manera:

$$(\sigma_M^r)^2 = \frac{1}{(K+1)^2}\left(K^2\left(\sigma_M^q\right)^2 + (\sigma_M(r))^2 + \sum_{l=1}^{K} COV_M^{(l,r)}\right)$$

$$(\sigma_N^r)^2 = \frac{1}{(K+1)^2}\left(K^2\left(\sigma_N^q\right)^2 + (\sigma_N(r))^2 + \sum_{l=1}^{K} COV_N^{(l,r)}\right)$$

donde se debe tener en cuenta la covarianza del filtro añadido en el nodo *r* con todos los filtros anteriores en la ruta. A continuación, las métricas actualizadas se pueden usar para recalcular la puntuación S en el nodo *r.*

[0076] Como apreciarán los expertos en la materia, el enrejado 409 ilustrado en la Figura 14 es una representación gráfica que facilita la comprensión de los cálculos de programación dinámica que realizará el ordenador de entrenamiento (que puede ser un ordenador de entrenamiento dedicado o, en algunos casos, puede ser el servidor de coincidencia de audio 5) durante el procedimiento de entrenamiento anterior. Los cálculos reales se realizarán con estructuras de datos adecuadas dentro de la memoria del ordenador de entrenamiento. El conjunto optimizado

resultante 45 de filtros se proporcionará a los dispositivos de usuario 1 para que puedan generar las huellas finas 43. También serán usados por el ordenador de entrenamiento (o por el servidor de coincidencia de audio 5) para generar las huellas finas de la base de datos 323.

Identificación del Segundo Conjunto Optimizado de Filtros

[0077] Como se discutió anteriormente, para que se pueda generar una huella gruesa significativa 309 a partir de la huella fina 43, las filas (o columnas) de la huella fina 43 deben ordenarse de modo que haya cierto nivel de coherencia en la huella fina, es decir, los filtros que tienden a producir resultados similares se ordenan uno al lado del otro. De esta manera, los filtros que generalmente están correlacionados entre sí están uno al lado del otro. Esto da como resultado una huella fina 43 que es menos aleatoria en apariencia, es decir, que tiene áreas más grandes del mismo valor binario (como se ilustra en la huella 43-2 que se muestra en la Figura 6).

[0078] Tal como se analizó anteriormente, los valores de covarianza que se determinan para dos filtros proporcionan información sobre la correlación entre los dos filtros. Por lo tanto, podemos determinar el orden de los filtros según los valores de covarianza calculados para los n filtros en el conjunto optimizado 45 de filtros. Esto se puede lograr, por ejemplo, usando un orden inverso de Cuthill-Mckee en los valores de covarianza más grandes para los n filtros. La información de pedido determinada también se proporciona a los dispositivos de usuario 1 con el primer conjunto optimizado 45 de filtros.

[0079] A continuación, se puede aplicar un procedimiento de entrenamiento similar para determinar el segundo conjunto 221 de filtros optimizados. La principal diferencia entre este procedimiento de entrenamiento y el procedimiento de entrenamiento discutido anteriormente es que los filtros se aplican a las huellas finas que se obtienen para los pares de clips de audio coincidentes y no coincidentes. Además, el procedimiento de optimización tiene un objetivo diferente.

[0080] Las Figuras 15a y 15B ilustran el procesamiento realizado para determinar los valores $B^{(l)}_M$ para pares coincidentes de clips de audio y los valores $B^{(l)}_N$ para pares no coincidentes de clips de audio. Como se puede observar comparando la Figura 15 con la Figura 12, después de que los espectrogramas se han calculado en las etapas s5, s7, s25 y s27, las huellas finas 441, 443, 445 y 447 se determinan en las etapas s41, s43, s45 y s47 respectivamente, usando el primer conjunto anterior 45 de filtros optimizados y usando la información de pedido para definir cómo se forman las huellas finas. El filtro actual bajo prueba (F(i)) se aplica a las huellas finas y la comparación bit a bit se realiza como antes para determinar $B_M(i)$ y $B_N(i)$.

[0081] El objetivo de optimización para determinar el segundo conjunto 221 de filtros es encontrar los filtros que darán como resultado un subconjunto mínimo de posibles entradas de base de datos coincidentes, sin excluir la entrada correcta en la base de datos 15, lo que, por lo tanto, reducirá el número de comparaciones requeridas de la huella fina 43. Las entradas de la base de datos que deben buscarse con más detalle (es decir, aquellas para las que se realizará una comparación entre huellas finas) son aquellas que caen por debajo de algún segundo umbral $\gamma^{(2)}$ (que será diferente del umbral y usado anteriormente). El número esperado (N_r) de entradas de base de datos por debajo del umbral viene dado por:

$$N_r = DP\left(B_N^{(2)} < \gamma^{(2)}\right) = \frac{D}{2}\operatorname{erfc}\left(\frac{\mu_N^{(2)} - \gamma^{(2)}}{\sqrt{2}\sigma_N^{(2)}}\right)$$

donde todos los parámetros son representativos de las huellas gruesas y no de las huellas finas, como lo indica el superíndice (2). Para cuantificar N_r, se debe establecer el umbral $\gamma^{(2)}$. Este umbral se establece definiendo una probabilidad aceptable ($P_{aceptar}$) de un falso negativo (clasificado falsamente una huella coincidente como una huella no coincidente) de:

$$P\left(B_M^{(2)} > \gamma^{(2)}\right) = \frac{1}{2}\operatorname{erfc}\left(\frac{\gamma^{(2)} - \mu_M^{(2)}}{\sigma_M^{(2)}\sqrt{2}}\right) = P_{\text{aceptar}}$$

[0082] Esta ecuación se puede invertir para obtener:

$$\gamma^{(2)} = \mu_M^{(2)} + \sigma_M^{(2)}\sqrt{2}\operatorname{erfc}^{-1}\left(2P_{\text{aceptar}}\right)$$

que proporciona el umbral $\gamma^{(2)}$ para una tasa de falsos negativos aceptable dada. Insertar este umbral en la ecuación para el número esperado (N_r) de entradas de base de datos por debajo del umbral produce:

$$N_r = \frac{D}{2}\operatorname{erfc}\left(\frac{\mu_N^{(2)} - \mu_M^{(2)}}{\sqrt{2}\,\sigma_N^{(2)}} - \frac{\sigma_M^{(2)}}{\sigma_N^{(2)}}\operatorname{erfc}^{-1}\left(2P_{\text{aceptar}}\right)\right)$$

[0083] Para minimizar este número, debemos encontrar la combinación de filtros que maximice el argumento de esta función de error complementaria. Así, la puntuación a maximizar en este segundo procedimiento de optimización viene dada por:

$$S^{(2)} = \frac{\mu_N^{(2)} - \mu_M^{(2)}}{\sqrt{2}\,\sigma_N^{(2)}} - \frac{\sigma_M^{(2)}}{\sigma_N^{(2)}}\operatorname{erfc}^{-1}\left(2P_{\text{aceptar}}\right)$$

[0084] Una vez más, la tarea es encontrar la combinación de filtros que maximice esta puntuación; donde la media agregada y las varianzas $(\mu^{(2)}{}_N,\ \mu^{(2)}{}_M,\ \sigma^{(2)}{}_N \ \text{y} \ \sigma^{(2)}{}_M)$ para cualquier combinación de filtros se pueden calcular usando las medias, varianzas y covarianzas determinadas para cada filtro de la combinación durante el procedimiento de entrenamiento ilustrado en la Figura 15. En otras palabras, estos parámetros agregados están relacionados con los parámetros individuales de los filtros individuales en el conjunto de la siguiente manera:

$$\mu_M^{(2)} = \frac{1}{n}\sum_{l=0}^{n-1}\mu_M^{(2)}(l) \quad \text{y} \quad \mu_N^{(2)} = \frac{1}{n}\sum_{l=0}^{n-1}\mu_N^{(2)}(l)$$

[0085] Donde n es el número de filtros en el segundo conjunto 221 de filtros optimizados. La varianza agregada (cuadrado de la desviación estándar) se convierte en una combinación de las varianzas de los filtros individuales que pertenecen al conjunto 221, así como la covarianza entre pares de filtros, de la siguiente manera:

$$(\sigma_M^{(2)})^2 = \frac{1}{n^2}\sum_{l=0}^{n-1}\left(\sigma_M^{(2)}(l)\right)^2 + \frac{1}{n^2}\sum_{l=0}^{n-1}\sum_{k=0,k\neq l}^{n-1} COV_M^{(2)(l,k)}$$

$$(\sigma_N^{(2)})^2 = \frac{1}{n^2}\sum_{l=0}^{n-1}\left(\sigma_N^{(2)}(l)\right)^2 + \frac{1}{n^2}\sum_{l=0}^{n-1}\sum_{k=0,k\neq l}^{n-1} COV_N^{(2)(l,k)}$$

Donde $COV^{(2)(i,k)}$ es la covarianza entre el filtro l el filtro k.

[0086] Las medias, varianzas y covarianzas para filtros individuales para pares de clips de audio coincidentes y no coincidentes se determinan a partir del procedimiento de entrenamiento analizado anteriormente con referencia a las Figuras 15a y 15b. En particular, la tasa de error de bits media para los N_M pares coincidentes de clips de audio y la tasa de error de bits media para los N_N pares no coincidentes de clips de audio para cada filtro (i) se determinan de la siguiente manera:

$$\mu_M^{(2)}(i) = \frac{1}{N_M}\sum_{k=1}^{N_M} B_{M,k}^{(2)}(i) \quad \text{y} \quad \mu_N^{(2)}(i) = \frac{1}{N_N}\sum_{k=1}^{N_N} B_{N,k}^{(2)}(i)$$

[0087] Y las variaciones correspondientes de:

$$\left(\sigma_M^{(2)}(i)\right)^2 = \frac{1}{N_M - 1}\sum_{k=1}^{N_M}\left(B_{M,k}^{(2)}(i) - \mu_M^{(2)}(i)\right)^2 \quad \text{y}$$

$$\left(\sigma_N^{(2)}(i)\right)^2 = \frac{1}{N_N - 1}\sum_{k=1}^{N_N}\left(B_{N,k}^{(2)}(i) - \mu_N^{(2)}(i)\right)^2$$

[0088] Además, el valor de covarianza *($COV_M^{(2)(i,j)}$)* entre dos filtros (i y j) para emparejar pares de clips de audio se puede determinar a partir de:

$$COV_M^{(2)(i,j)} = \frac{1}{N_M - 1}\sum_{k=1}^{N_M}\left(B_{M,k}^{(2)}(i) - \mu_M^{(2)}(i)\right)\left(B_{M,k}^{(2)}(j) - \mu_M^{(2)}(j)\right)$$

[0089] Y el valor de covarianza *($COV_N^{(2)(i,j)}$)* entre dos filtros (i y j) para pares de clips de audio no coincidentes se puede determinar a partir de:

$$COV_N^{(2)(i,j)} = \frac{1}{N_N - 1}\sum_{k=1}^{N_N}\left(B_{N,k}^{(2)}(i) - \mu_N^{(2)}(i)\right)\left(B_{N,k}^{(2)}(j) - \mu_N^{(2)}(j)\right)$$

[0090] Como antes, no es práctico calcular la puntuación anterior ($S^{(2)}$) para cada combinación posible de n filtros del conjunto de 16.000 filtros posibles: el número de combinaciones posibles es demasiado grande. Sin embargo, como antes, podemos usar la Programación Dinámica para encontrar la ruta que tenga la puntuación máxima ($S^{(2)}$) usando el enrejado 409 y las técnicas de propagación de ruta analizadas anteriormente. Este procedimiento de programación dinámica identificará la mejor ruta a través del enrejado 409, que a su vez identifica la mejor combinación de filtros para formar el segundo conjunto 221 de filtros optimizados que usa el servidor de coincidencia de audio 5 para generar las huellas gruesas a partir de huellas finas.

[0091] Como antes, la puntuación de la ruta se puede acumular durante la propagación de la ruta de programación dinámica para encontrar la mejor ruta a través del enrejado, por lo que no es necesario volver a calcular la puntuación $S^{(2)}$ cada vez que se añade un nuevo nodo (filtro) a una ruta candidata. En cambio, la puntuación se actualiza usando las estadísticas individuales para el filtro asociado con el nuevo nodo *r,* columna *K+1,* cuando proviene del nodo *q* en el número de columna *K* como antes:

$$\mu_M^{r\,(2)} = \frac{K\mu_M^{q\,(2)} + \mu_M^{(2)}(r)}{K+1} \quad \text{y} \quad \mu_M^{r\,(2)} = \frac{K\mu_N^{q\,(2)} + \mu_N^{(2)}(r)}{K+1}$$

donde $\mu_M^{q\,(2)}$ y $\mu_N^{q\,(2)}$ son las medias agregadas en el nodo *q* y $\mu_M^{r\,(2)}$ y $\mu_M^{r\,(2)}$ son las medias agregadas en el nodo *r.* Del mismo modo, las varianzas $\left(\sigma_M^{(2)}\right)^2$ y $\left(\sigma_N^{(2)}\right)^2$ se pueden actualizar de la columna *K* a la columna *K+1* de la siguiente manera:

$$\left(\sigma_M^{r\,(2)}\right)^2 = \frac{1}{(K+1)^2}\left(K^2\left(\sigma_M^{q\,(2)}\right)^2 + \left(\sigma_M^{(2)}(r)\right)^2 + \sum_{l=1}^{K} COV_M^{(2)(l,r)}\right)$$

$$\left(\sigma_N^{r(2)}\right)^2 = \frac{1}{(K+1)^2}\left(K^2\left(\sigma_N^{q(2)}\right)^2 + \left(\sigma_N^{(2)}(r)\right)^2 + \sum_{l=1}^{K} COV_N^{(2)(l,r)}\right)$$

[0092] A continuación, las métricas actualizadas se pueden usar para recalcular la puntuación $S^{(2)}$ en el nodo *r*.

Modificaciones y Realizaciones Adicionales

[0093] Se ha descrito anteriormente una realización que ilustra la forma en que se pueden crear huellas para la identificación de una señal de audio en una base de datos de audio. Como apreciarán los expertos en la materia, se pueden realizar diversas modificaciones y mejoras a la realización anterior y ahora se describirán algunas de estas modificaciones.

[0094] En la realización anterior, el dispositivo de usuario generó una huella fina que transmitió al servidor de coincidencia de audio que generó una huella gruesa a partir de la huella fina. En otra realización, el propio dispositivo de usuario puede calcular la huella gruesa y enviarla junto con la huella fina al servidor de coincidencia de audio.

[0095] En las realizaciones anteriores, se generó una huella gruesa a partir de la huella fina. Esto es particularmente beneficioso en el escenario donde un dispositivo de usuario determina la huella fina y la envía a un servidor remoto para su comparación con las entradas de la base de datos. En otras realizaciones donde el dispositivo de usuario calcula tanto la huella gruesa como la huella fina, la huella gruesa se puede determinar a partir del espectrograma del audio capturado en lugar de a partir de la huella fina. En este caso, el segundo conjunto 221 de filtros optimizados se entrenaría usando la segunda puntuación descrita anteriormente, pero según vectores binarizados obtenidos aplicando los filtros al espectrograma en lugar de a la huella fina. Este también sería el caso si el dispositivo de usuario transmitiera la huella fina y el espectrograma al servidor remoto, que a continuación calculó la huella gruesa a partir del espectrograma recibido. Sin embargo, esta última posibilidad no se prefiere, ya que requiere que el espectrograma (que es una gran estructura de datos) se transmita desde el dispositivo del usuario al servidor.

[0096] En las realizaciones anteriores, el dispositivo de usuario o el servidor de coincidencia de audio generaron una huella gruesa a partir de una huella fina usando un conjunto de filtros optimizados que se aplican a la huella fina. En una realización más simple, la huella gruesa podría generarse simplemente submuestreando la huella fina o promediando la huella fina. Sin embargo, se prefiere aplicar el segundo conjunto de filtros optimizados descrito anteriormente a la huella fina, ya que se ha descubierto que la huella gruesa resultante es mejor para minimizar el número de entradas de la base de datos que se encuentra que posiblemente coinciden mientras se minimizan los falsos positivos y falsos negativos.

[0097] En realizaciones donde el tamaño de la base de datos es relativamente pequeño, el servidor de coincidencia de audio y la base de datos pueden formar parte del propio dispositivo de usuario. En este caso, no es necesario que el dispositivo del usuario transmita ningún dato de huella a través de la red de telecomunicaciones o de la red informática. Estos datos simplemente se enviarían entre los diferentes componentes de software que se ejecutan en el dispositivo del usuario (aunque cualquier resultado de la coincidencia puede transmitirse a través de la red a un servidor).

[0098] Las Figuras 12 y 15 ilustran dos bases de datos, una para muestras de audio originales y la otra para versiones distorsionadas de las muestras de audio. Como apreciarán los expertos en la materia, todas estas muestras de audio pueden almacenarse en una sola base de datos en lugar de en dos bases de datos separadas. De manera similar, estas figuras ilustran que los espectrogramas se determinan para cada muestra de audio en un par coincidente y para cada muestra de audio en un par no coincidente. Como apreciarán los expertos en la materia, las mismas muestras de audio se pueden incluir en un par de muestras de audio coincidentes y en un par de muestras de audio no coincidentes. En este caso claramente no es necesario determinar el espectrograma para la misma muestra de audio dos veces. Es decir, el procedimiento de optimización solo necesita determinar el espectrograma para cada muestra de audio en la base de datos y a continuación aplicar cada filtro a cada espectrograma.

[0099] En las realizaciones anteriores, el procedimiento de optimización estableció una tasa de falsos positivos aceptable y a continuación encontró el conjunto de filtros que minimizaban la tasa de falsos negativos. En otra realización, el procedimiento de optimización puede establecer una tasa de falsos negativos aceptable y a continuación encontrar el conjunto de filtros que minimiza la tasa de falsos positivos. En una realización adicional, el procedimiento de optimización puede establecer una tasa de falsos positivos aceptable y una tasa de falsos negativos aceptable y a continuación encontrar el conjunto de filtros que minimiza alguna otra función de costo.

[0100] En la realización anterior, los procedimientos de programación dinámica seleccionaron la ruta a través del enrejado 409 que tiene la puntuación más alta. Como apreciarán los expertos en la materia, la mejor ruta u óptima que se elija no tiene que ser la que tenga la puntuación más alta; por ejemplo, podría usarse en su lugar la ruta que tenga la segunda puntuación más alta o la tercera puntuación más alta.

[0101] En la realización anterior, un dispositivo de usuario capturó sonidos usando un micrófono y las muestras de audio se procesaron usando una aplicación de software almacenada en el dispositivo de usuario. Como apreciarán los expertos en la materia, parte o la totalidad de este procesamiento puede estar formado por circuitos de hardware dedicados, aunque se prefiere el software debido a su capacidad para añadirse al dispositivo de usuario portátil después de la fabricación y su capacidad para actualizarse una vez cargado. El software para hacer que el dispositivo de usuario portátil funcione de la manera anterior puede proporcionarse como una señal o en un soporte tal como un disco compacto u otro medio de soporte. Adicionalmente, se puede usar una gama de otros dispositivos portátiles, tales como ordenadores portátiles, PDA, tabletas y similares. Del mismo modo, el software que forma parte del servidor de coincidencia de audio puede reemplazarse por circuitos de hardware adecuados, como los circuitos integrados específicos de la aplicación.

[0102] Las realizaciones anteriores han descrito un sistema de coincidencia de audio basado en huellas. Este sistema también se puede usar junto con un sistema de coincidencia de audio de tipo marca de agua que detecta marcas de agua ocultas que se han ocultado en el audio. En particular, si no se puede encontrar una marca de agua en algún audio capturado, entonces el reconocimiento de audio de huellas anterior se puede usar para identificar el audio capturado.

[0103] En las realizaciones anteriores, los conjuntos de filtros optimizados usaron cinco tipos diferentes de filtro. En otras realizaciones, se pueden usar más o menos tipos de filtros. Además, no es esencial usar filtros de forma rectangular; se podrían usar otras formas irregulares de filtros (como filtros en forma de "L"). La forma del filtro solo define los valores vecinos en el espectrograma (o en la huella fina) que se ponderan por el coeficiente correspondiente en el filtro y a continuación se combinan.

[0104] En la realización anterior, al generar la huella fina, cada filtro del primer conjunto de filtros optimizados se escalonó a lo largo del espectrograma etapa por etapa. Esto significaba que la huella fina tenía la misma dimensión temporal que el espectrograma original. Como apreciarán los expertos en la materia, la huella fina podría omitir algunos de estos puntos de datos, al principio o al final del espectrograma. Además, también se podría usar un tamaño de etapa más grande, por ejemplo, se podría omitir un punto de tiempo en cada etapa. En este caso, la huella fina tendría una duración temporal de la mitad de la del espectrograma. Entonces, si el espectrograma tuviera 500 puntos de tiempo, la huella fina generada tendría 250 puntos de tiempo.

[0105] En la realización descrita anteriormente, la huella gruesa se generó con una resolución de tiempo l/10º a la del espectrograma. Es decir, se omitieron 10 puntos de tiempo entre las etapas cuando el segundo conjunto de filtros optimizados se pasó por el espectrograma. Como apreciarán los expertos en la materia, por supuesto, se podrían usar otros tamaños de etapas para lograr una compresión diferente de los datos en la dimensión de tiempo.

[0106] En las realizaciones anteriores, la señal de audio capturada por el dispositivo de usuario era una señal acústica. En otras realizaciones, el dispositivo de usuario puede capturar la señal de audio como una señal electromagnética recibida a través de la antena del dispositivo de usuario; o en el caso de que el dispositivo de usuario no sea un dispositivo portátil y sea, por ejemplo, un ordenador personal o un decodificador o un televisor inteligente, la señal de audio puede capturarse a través de una señal recibida a través de una red de televisión de difusión (por ejemplo, una red satelital, una red de cable, una red ADSL o similares), Internet o alguna otra red informática.

[0107] En las realizaciones anteriores, cada entrada en la base de datos contenía una huella gruesa y una huella fina. En otra realización, cada entrada de la base de datos puede no contener la huella gruesa, que en cambio se puede generar cuando sea necesario a partir de la huella fina de la base de datos. La huella gruesa de la base de datos puede generarse a partir de la huella fina de la base de datos de varias maneras diferentes, al igual que la huella gruesa de la consulta puede determinarse de varias maneras diferentes a partir de la huella fina de la consulta. Estas diferentes formas de determinar la huella gruesa a partir de la huella fina se analizaron anteriormente y no se repetirán de nuevo. Huelga decir que la técnica usada para generar la huella de consulta gruesa debe ser la misma que la técnica que se usa para generar la huella de base de datos gruesa.

REIVINDICACIONES

1. Un procedimiento de identificación de un conjunto optimizado de filtros para su uso en la generación de una huella acústica, comprendiendo el procedimiento:

i) proporcionar una o más bases de datos (351, 353) que comprenden una pluralidad de muestras de audio que incluyen N_M pares coincidentes de muestras de audio y N_N pares no coincidentes de muestras de audio, cada par coincidente de muestras de audio comprende una muestra de audio original y una versión distorsionada de una misma señal de audio original y cada par no coincidente de muestras de audio comprende una muestra de audio original y una versión de una señal de audio original diferente;
ii) determinar (s5, s7, s25, s27) un espectrograma para cada una de la pluralidad de muestras de audio en la una o más bases de datos;
iii) aplicar (s9, s11, s29, s31) cada uno de los N_f filtros candidatos a los espectrogramas y binarizar (361, 363, 369,371) un resultado para generar una pluralidad de vectores de bits binarios, estando asociado cada vector de bits binarios con un filtro candidato y una muestra de audio;
iv) comparar (s13) los bits binarios en los vectores asociados con un par coincidente seleccionado de muestras de audio para un filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par coincidente seleccionado de muestras de audio;
v) repetir la etapa iv) para cada par coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa iv) para el filtro actual y los pares coincidentes de muestras de audio;
vi) comparar (s33) los bits binarios en los vectores asociados con un par no coincidente seleccionado de muestras de audio para el filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par no coincidente seleccionado de muestras de audio;
vii) repetir la etapa vi) para cada par no coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa vi) para el filtro actual y los pares no coincidentes de muestras de audio;
viii) repetir las etapas iv) a vii) para cada filtro candidato para determinar la información de media y varianza para cada filtro candidato para los pares coincidentes de muestras de audio y para determinar la información de media y varianza para cada filtro candidato para los pares no coincidentes de muestras de audio; y
ix) determinar, usando una técnica de optimización de programación dinámica, un subconjunto de dichos filtros candidatos como dicho conjunto optimizado de filtros para su uso en la generación de una huella acústica usando la información de media y varianza determinada para cada filtro candidato para los pares coincidentes de muestras de audio y la información de media y varianza determinada para cada filtro candidato para los pares no coincidentes de muestras de audio.

2. Un procedimiento según la reivindicación 1, donde la determinación del conjunto optimizado de filtros usa la información de media y varianza determinada para cada filtro candidato para los pares coincidentes de muestras de audio y la información de media y varianza determinada para cada filtro candidato para los pares no coincidentes de muestras de audio para minimizar la posibilidad de falsos positivos o para minimizar la posibilidad de falsos negativos o una combinación de ambos.

3. Un procedimiento según la reivindicación 1 o 2, que comprende además determinar la información de covarianza para cada par de filtros de una pluralidad de pares de filtros usando la información de media y varianza determinada; y donde dicha determinación de dicho conjunto optimizado de filtros para su uso en la generación de una huella acústica usa la información de media y varianza determinada y la información de covarianza.

4. Un aparato para identificar un conjunto optimizado de filtros para su uso en la generación de una huella acústica, comprendiendo el aparato:

una o más bases de datos (351, 353) que comprenden una pluralidad de muestras de audio que incluyen N_M pares coincidentes de muestras de audio y N_N pares no coincidentes de muestras de audio, cada par coincidente de muestras de audio comprende una muestra de audio original y una versión distorsionada de una misma señal de audio original y cada par no coincidente de muestras de audio comprende una muestra de audio original y una versión de una señal de audio original diferente; y
uno o más procesadores (201) configurados para:

i) determinar (s5, s7, s25, s27) un espectrograma para cada una de dicha pluralidad de muestras de audio en la una o más bases de datos;
ii) aplicar (s9, s11, s29, s31) cada uno de los N_f filtros candidatos a los espectrogramas y binarizar (361, 363, 369,371) un resultado para generar una pluralidad de vectores de bits binarios, estando asociado cada vector de bits binarios con un filtro candidato y una muestra de audio;
iii) comparar (s13) los bits binarios en los vectores asociados con un par coincidente seleccionado de muestras de audio para un filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par coincidente seleccionado de muestras de audio;

iv) repetir iii) para cada par coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en la etapa iii) para el filtro actual y los pares coincidentes de muestras de audio;
v) comparar (s33) los bits binarios en los vectores asociados con un par no coincidente seleccionado de muestras de audio para el filtro actual para determinar la información de tasa de error de bits para el filtro actual y el par no coincidente seleccionado de muestras de audio;
vi) repetir v) para cada par no coincidente de muestras de audio para determinar la información de media y varianza para la información de tasa de error de bits determinada en v) para el filtro actual y los pares no coincidentes de muestras de audio;
vii) repetir iii) a vi) para cada filtro candidato para determinar la información de media y varianza para cada filtro candidato para los pares coincidentes de muestras de audio y para determinar la información de media y varianza para cada filtro candidato para los pares no coincidentes de muestras de audio; y
viii) determinar, usando una técnica de optimización de programación dinámica, un subconjunto de dichos filtros candidatos como dicho conjunto optimizado de filtros para su uso en la generación de una huella acústica usando la información de media y varianza determinada para cada filtro candidato para los pares coincidentes de muestras de audio y la información de media y varianza determinada para cada filtro candidato para los pares no coincidentes de muestras de audio.

5. Un dispositivo de usuario (1) para su uso en un sistema de coincidencia de audio, comprendiendo el dispositivo de usuario:

medios (23) para capturar una señal de audio;
medios (63) para procesar la señal de audio capturada para generar una huella acústica de consulta representativa de la señal de audio capturada usando un conjunto de filtros optimizados determinados usando el procedimiento según cualquiera de las reivindicaciones 1 a 3;
medios (27) para emitir la huella acústica de consulta a un servidor de coincidencia de audio; y
medios (27) para recibir una respuesta coincidente que comprende información relacionada con el audio capturado.

6. Un producto de programa informático que comprende instrucciones implementables por ordenador para hacer que un dispositivo informático programable realice el procedimiento según cualquiera de las reivindicaciones 1 a 3.

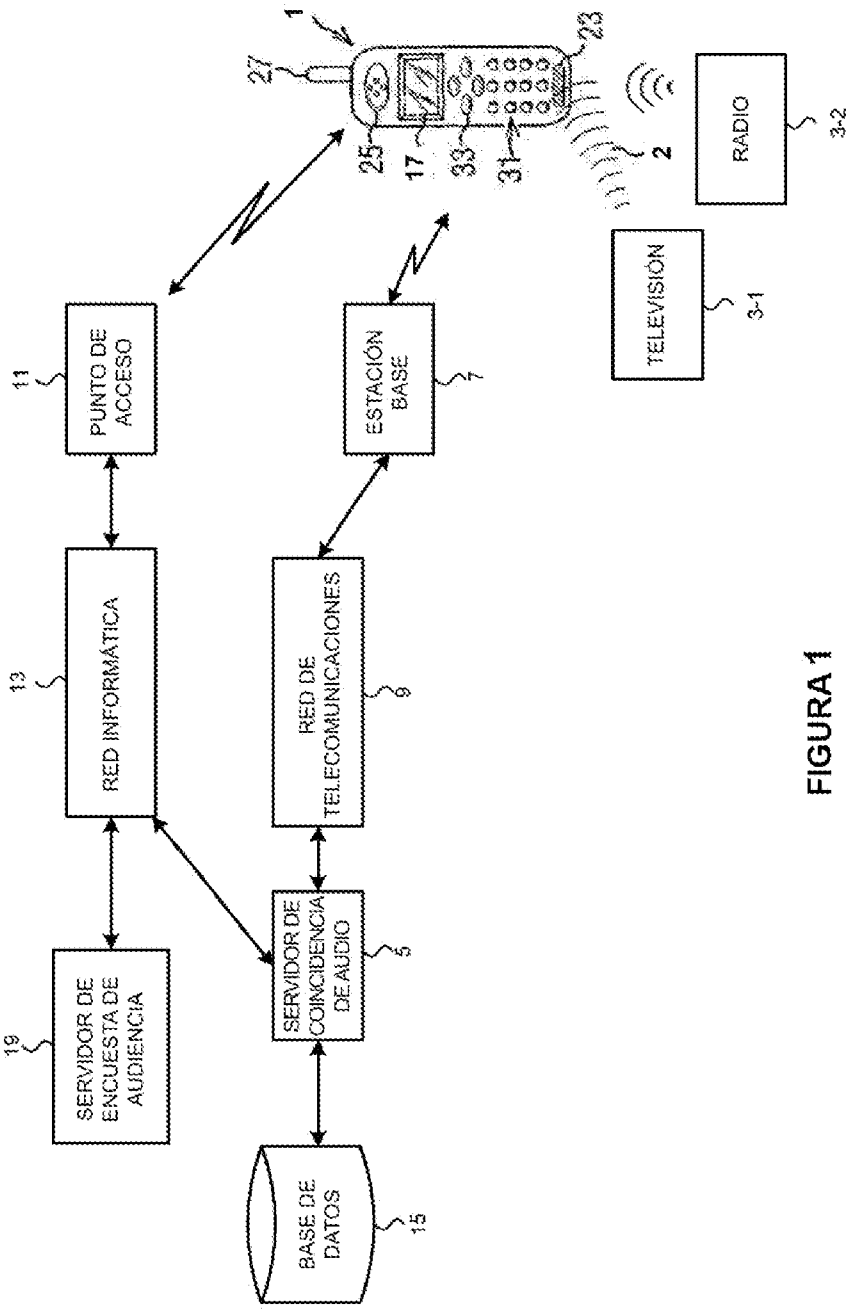
1
27
23
25
17
33
31
2
RADIO
3-2
TELEVISIÓN
3-1
11
PUNTO DE ACCESO
ESTACIÓN BASE
7
13
RED INFORMÁTICA
RED DE TELECOMUNICACIONES
9
SERVIDOR DE COINCIDENCIA DE AUDIO
5
19
SERVIDOR DE ENCUESTA DE AUDIENCIA
BASE DE DATOS
15

FIGURA 1

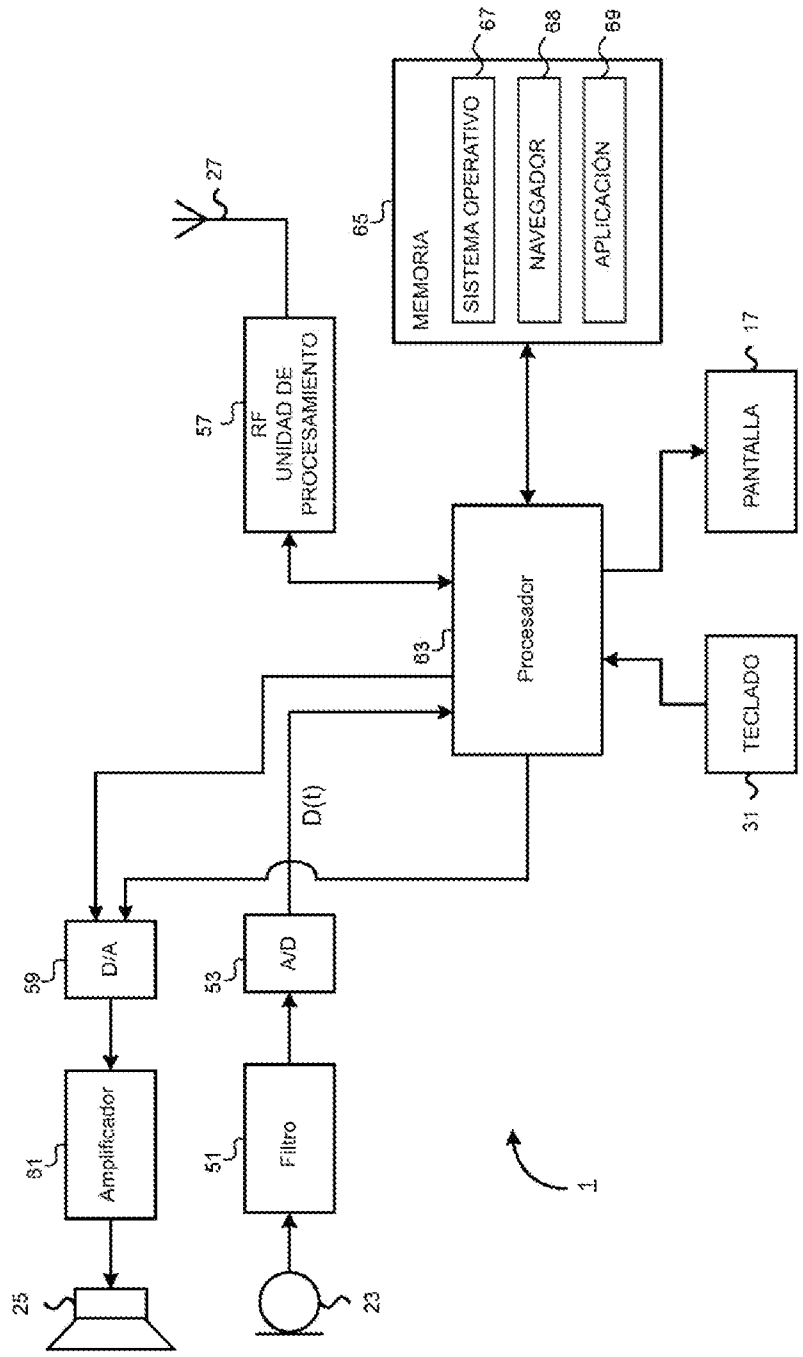
MEMORIA
65
SISTEMA OPERATIVO
67
NAVEGADOR
68
APLICACIÓN
69
27
RF
UNIDAD DE
PROCESAMIENTO
57
Procesador
63
PANTALLA
17
TECLADO
31
D(t)
D/A
59
A/D
53
Amplificador
61
Filtro
51
25
23
1

FIGURA 2

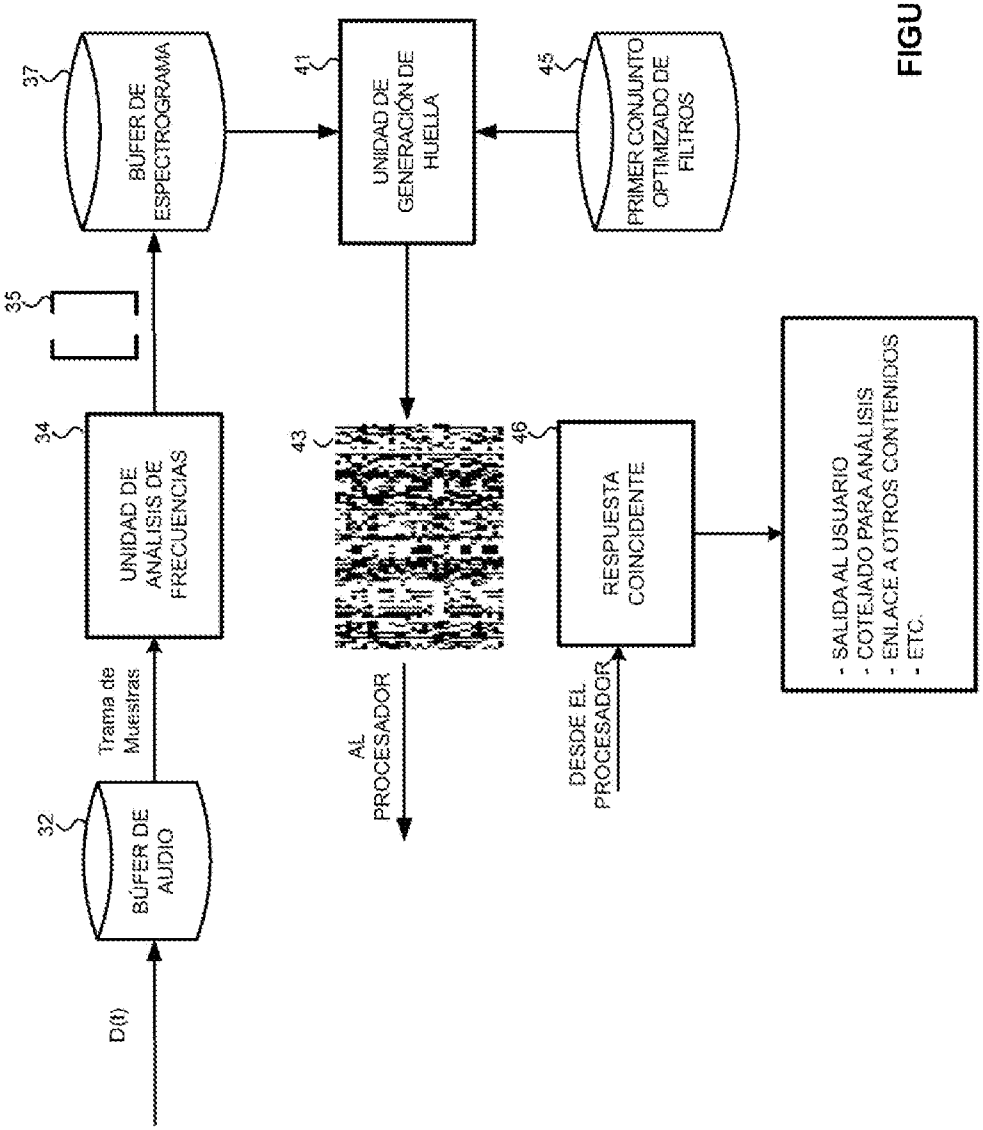
D(t)
32
BÚFER DE AUDIO
Trama de Muestras
34
UNIDAD DE ANÁLISIS DE FRECUENCIAS
35
37
BÚFER DE ESPECTROGRAMA
41
UNIDAD DE GENERACIÓN DE HUELLA
45
PRIMER CONJUNTO OPTIMIZADO DE FILTROS
43
AL PROCESADOR
DESDE EL PROCESADOR
46
RESPUESTA COINCIDENTE
- SALIDA AL USUARIO
- COTEJADO PARA ANÁLISIS
- ENLACE A OTROS CONTENIDOS
- ETC.

FIGURA 3

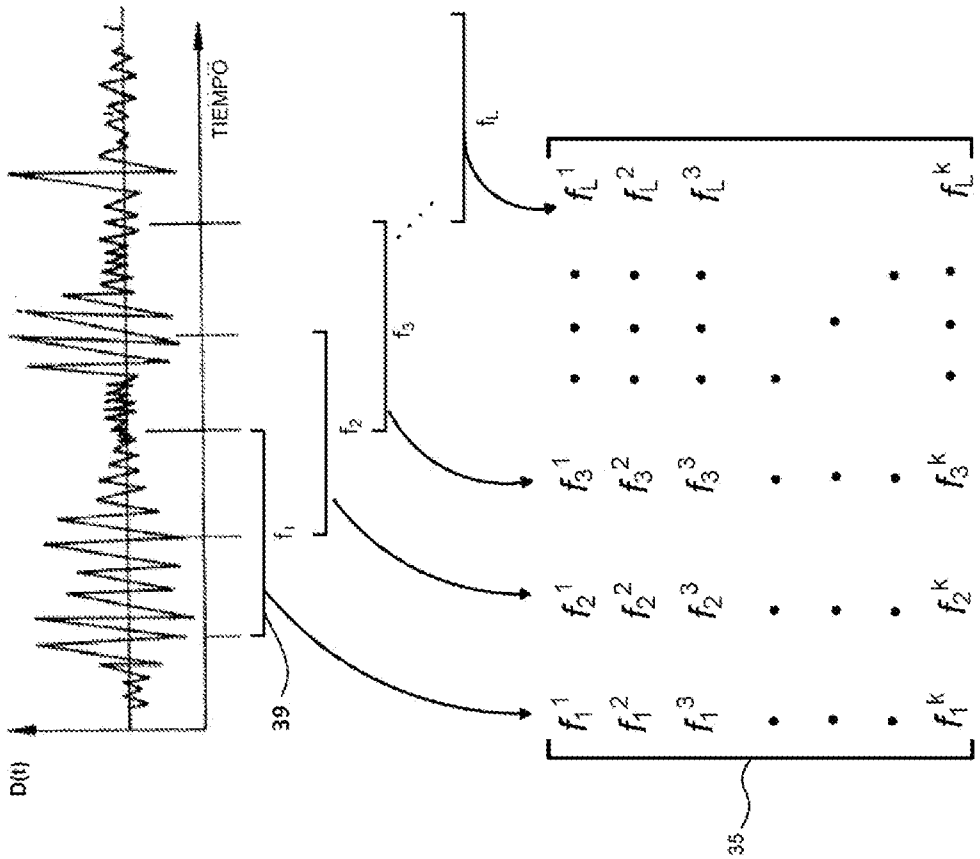
D(t)
TIEMPO
39
35
f_1
f_2
f_3
f_L
f_1^1 f_1^2 f_1^3 f_1^k
f_2^1 f_2^2 f_2^3 f_2^k
f_3^1 f_3^2 f_3^3 f_3^k
f_L^1 f_L^2 f_L^3 f_L^k

FIGURA 4

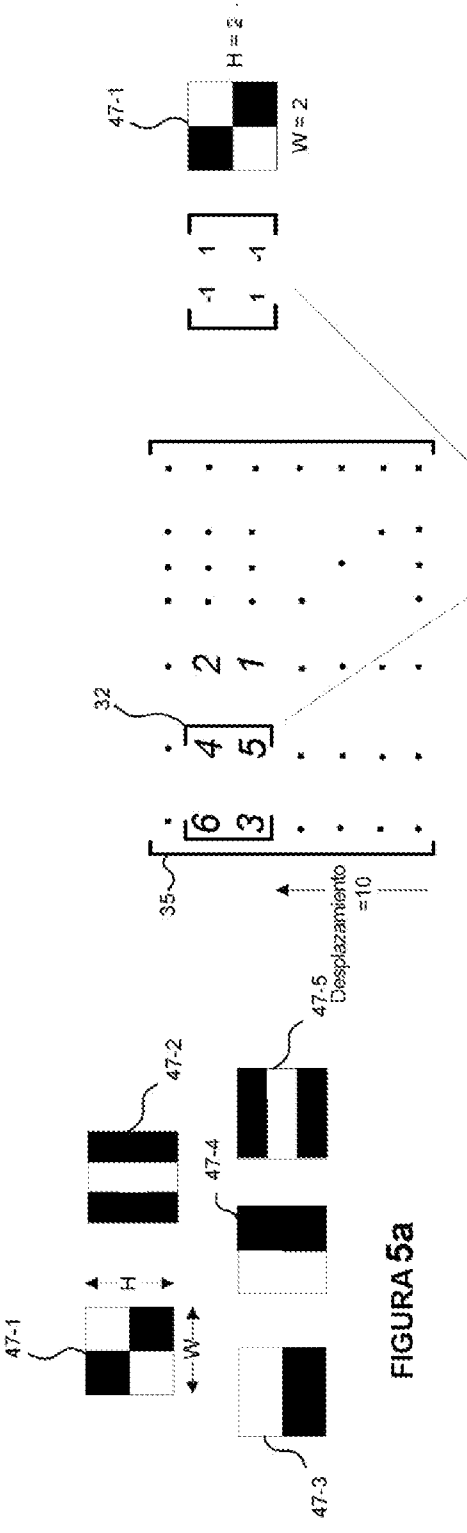

FIGURA 5a

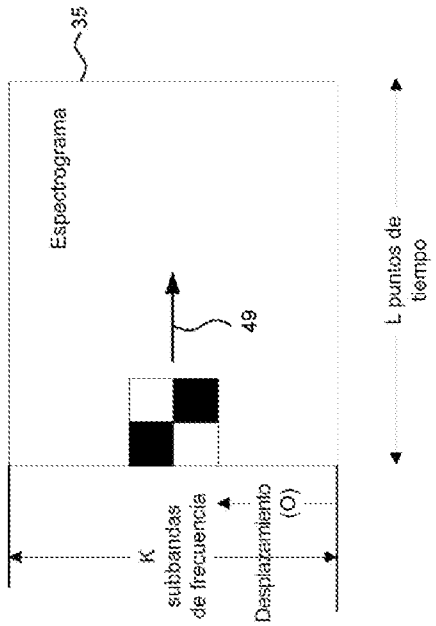

FIGURA 5b

FIGURA 5c

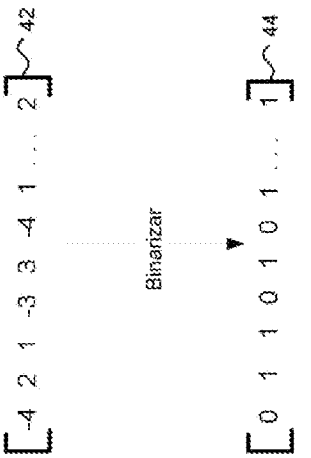

FIGURA 5e

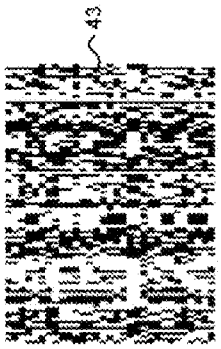

FIGURA 5f

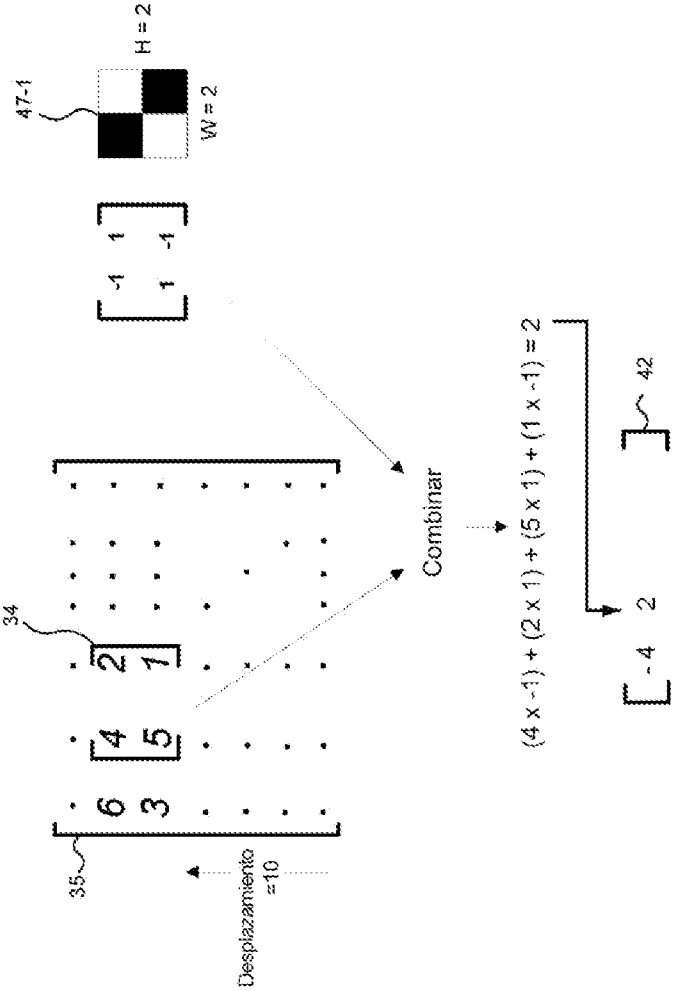

FIGURA 5d

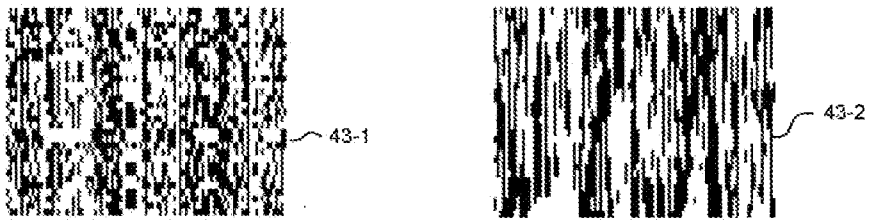

FIGURA 6

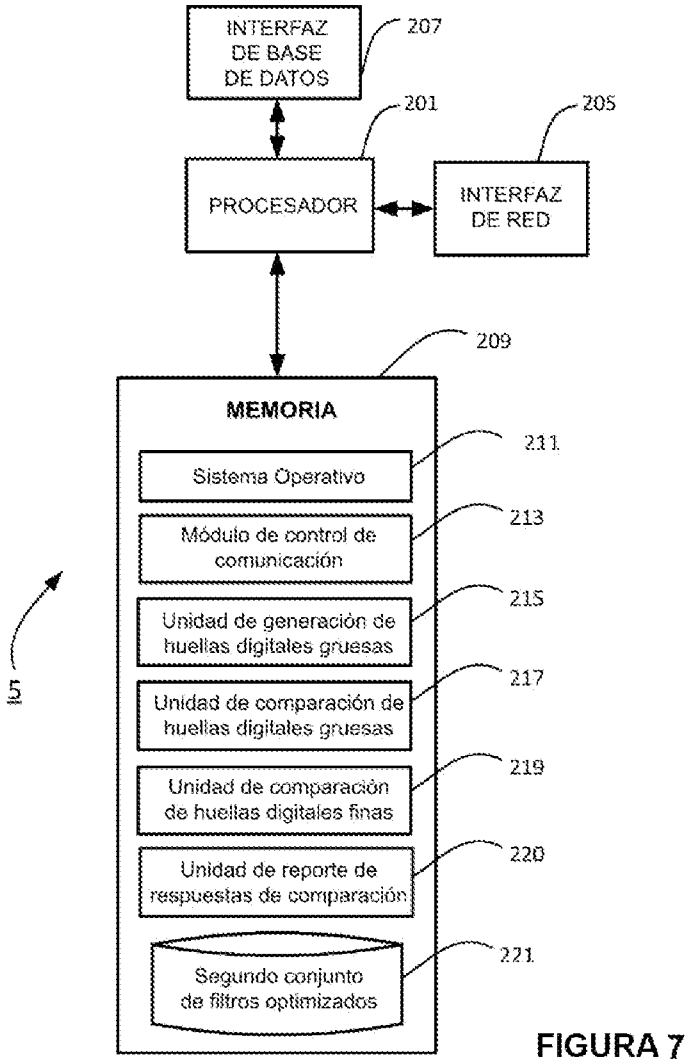

FIGURA 7

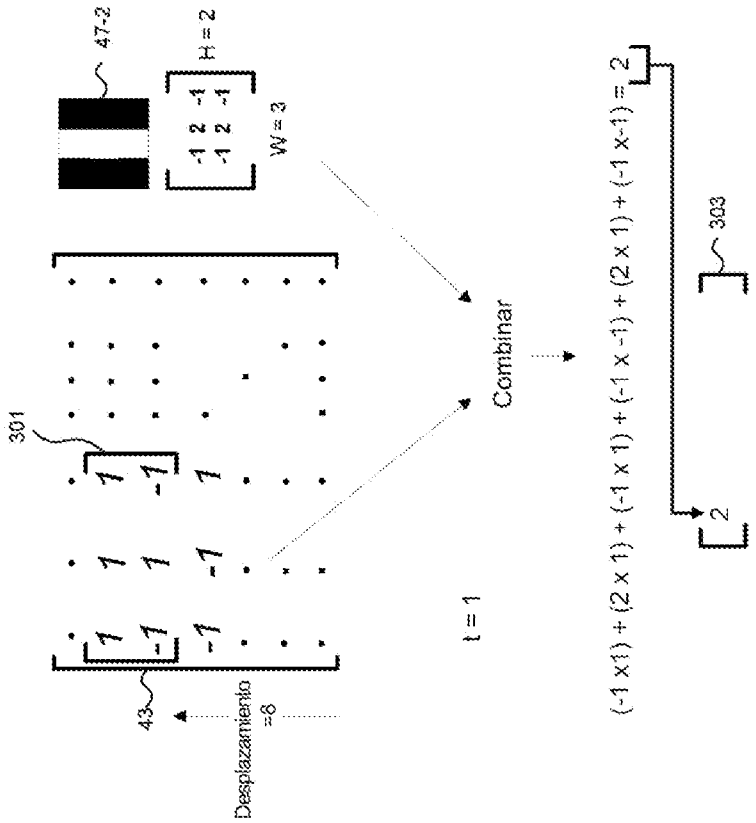
47-2
H = 2
W = 3
301
43
Desplazamiento =8
t = 1
Combinar
(-1 x1) + (2 x 1) + (-1 x 1) + (-1 x -1) + (2 x 1) + (-1 x-1) = 2
303
2

FIGURA 8a

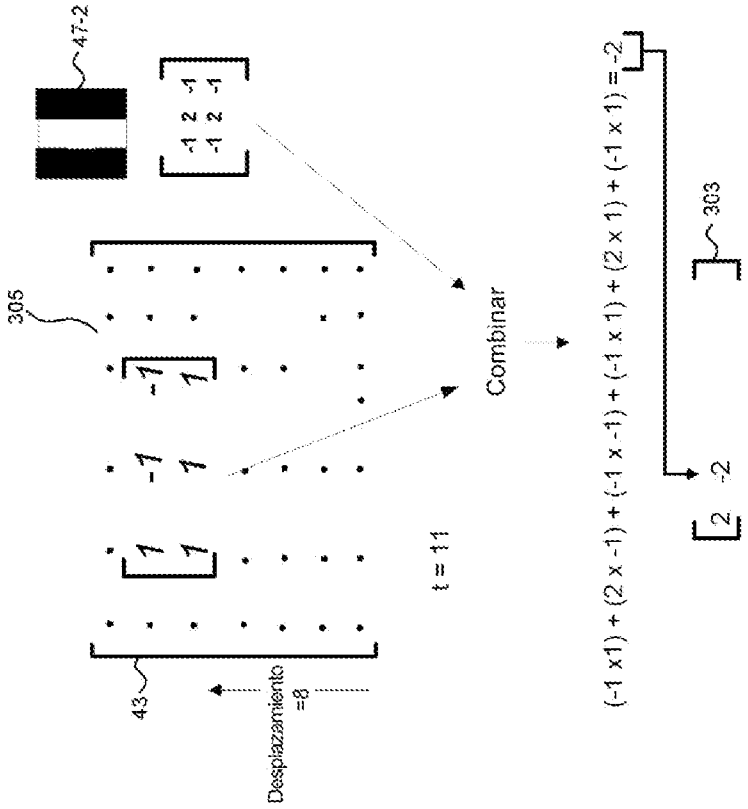
47-2
-1 2 -1
-1 2 -1
305
-1
1
-1
1
1
1
43
Desplazamiento
=8
t = 11
Combinar
(-1 x1) + (2 x -1) + (-1 x -1) + (-1 x 1) + (2 x 1) + (-1 x 1) = -2
303
2
-2

FIGURA 8b

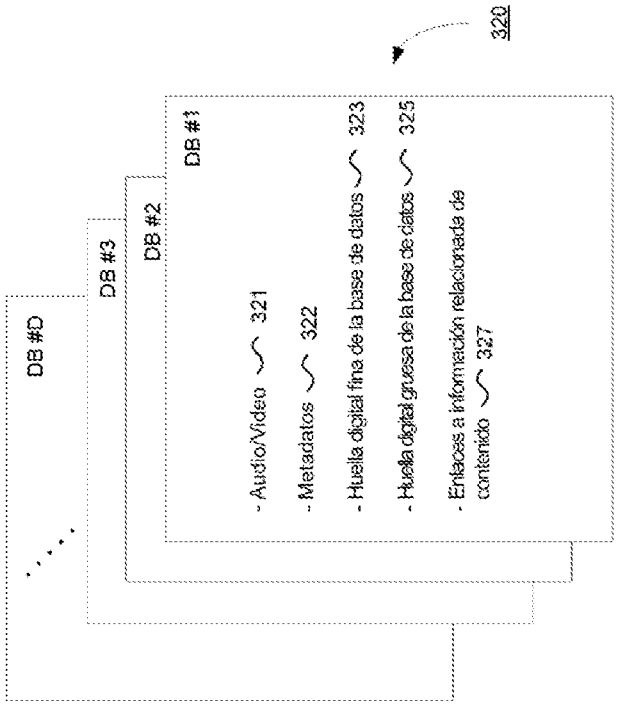

Fig.9

Fig.8c

Fig.8d

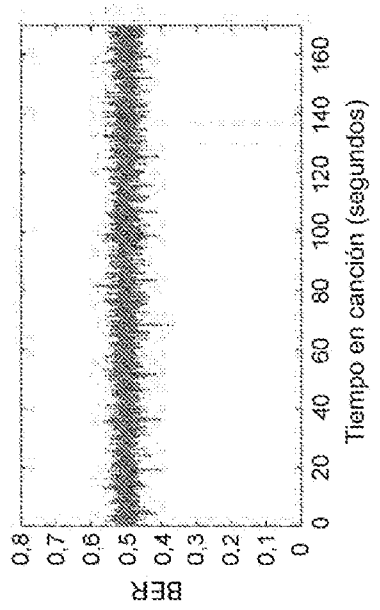

FIGURA 10b

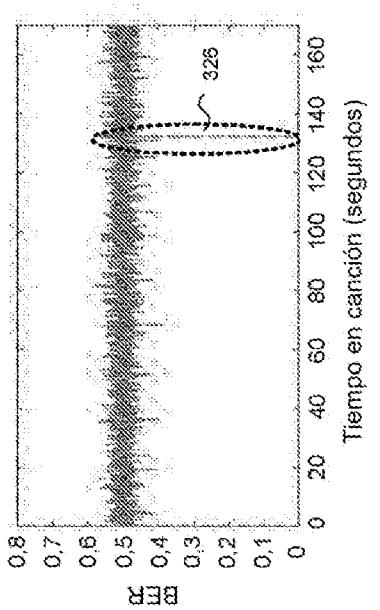

FIGURA 10c

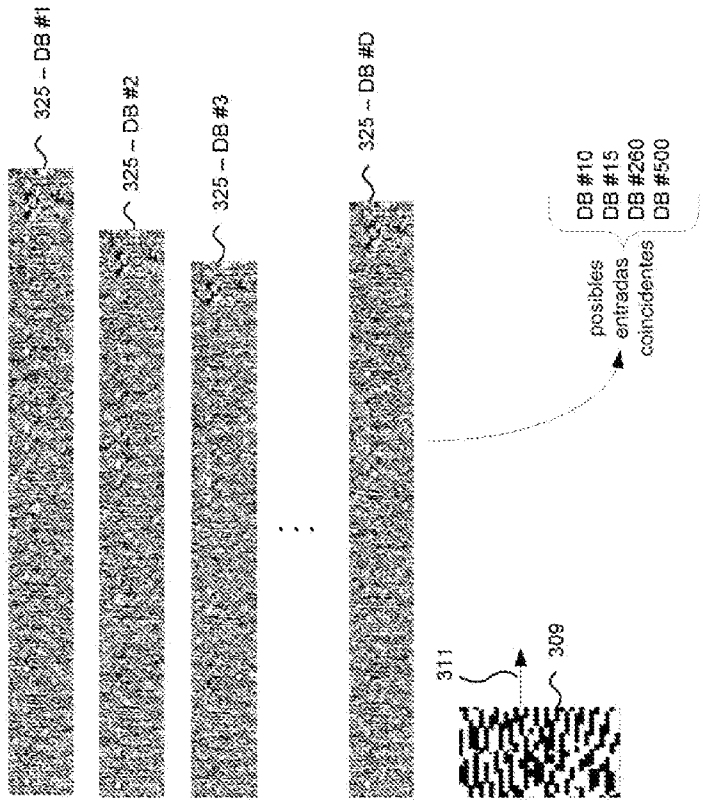

FIGURA 10a

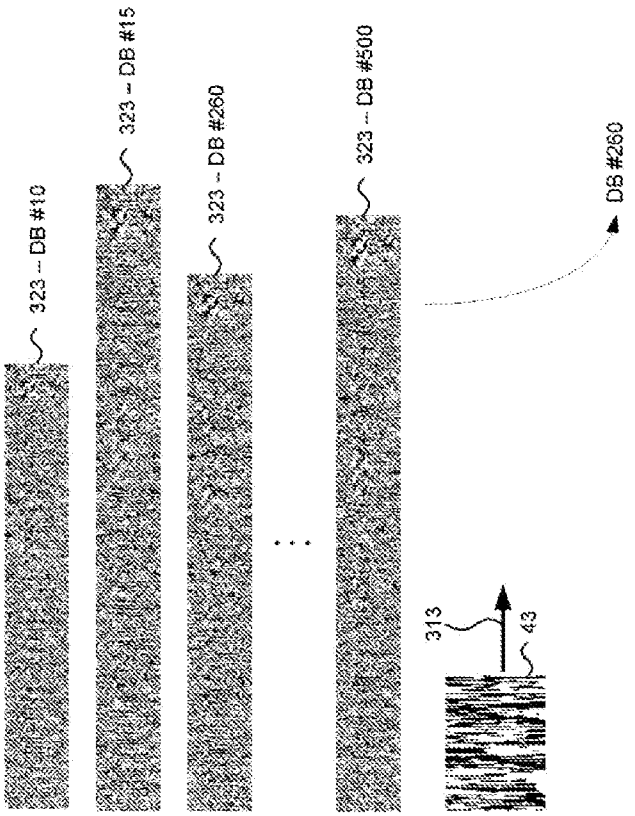

FIGURA 11

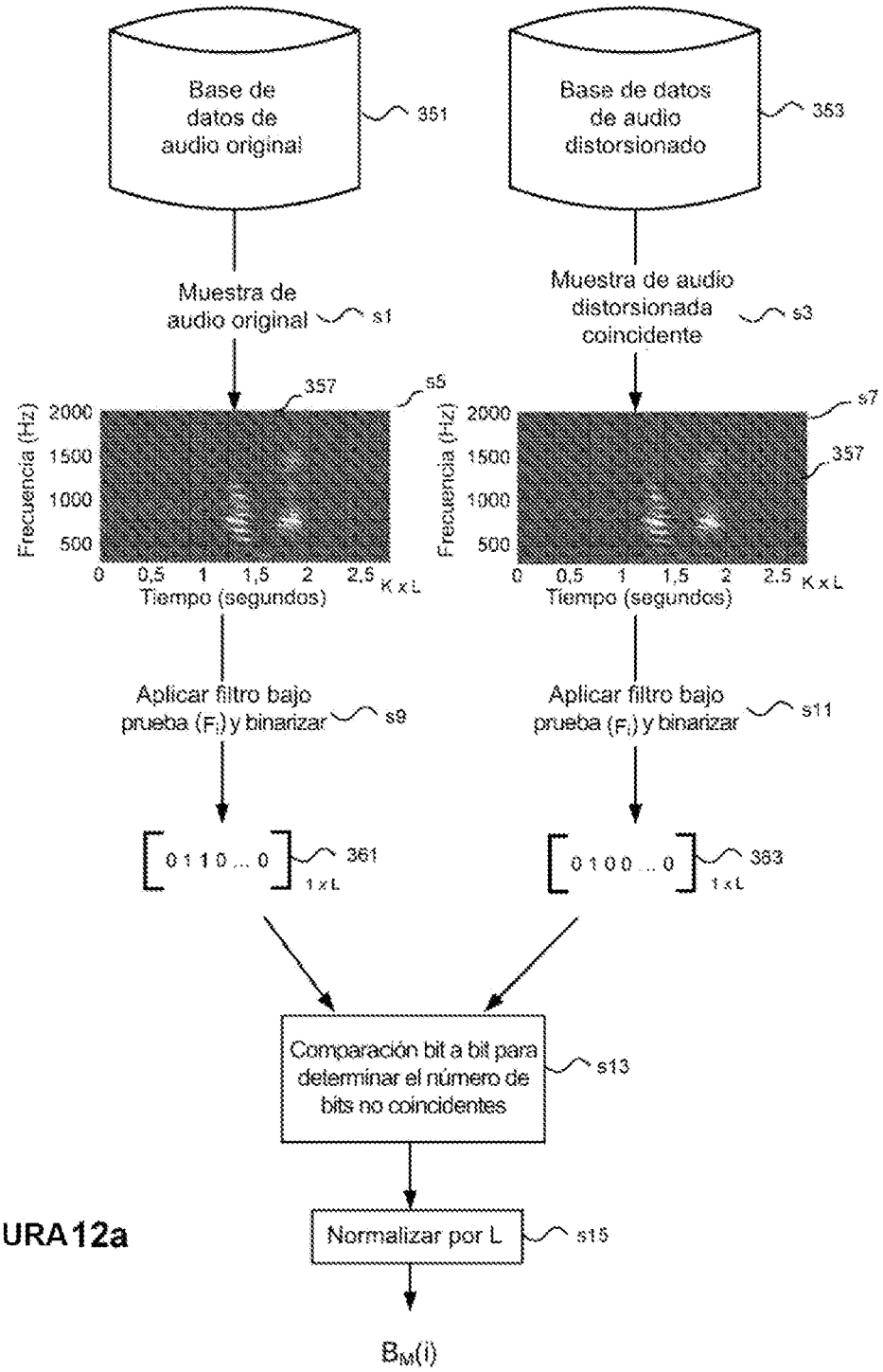
Base de datos de audio original
351
Base de datos de audio distorsionado
353
Muestra de audio original
s1
Muestra de audio distorsionada coincidente
s3
357
s5
s7
Frecuencia (Hz)
2000
1500
1000
500
0
0,5
1
1,5
2
2,5
Tiempo (segundos)
K x L
Aplicar filtro bajo prueba (F_i) y binarizar
s9
s11
0 1 1 0 ... 0
361
0 1 0 0 ... 0
363
1 x L
Comparación bit a bit para determinar el número de bits no coincidentes
s13
Normalizar por L
s15
$B_M(i)$

FIGURA 12a

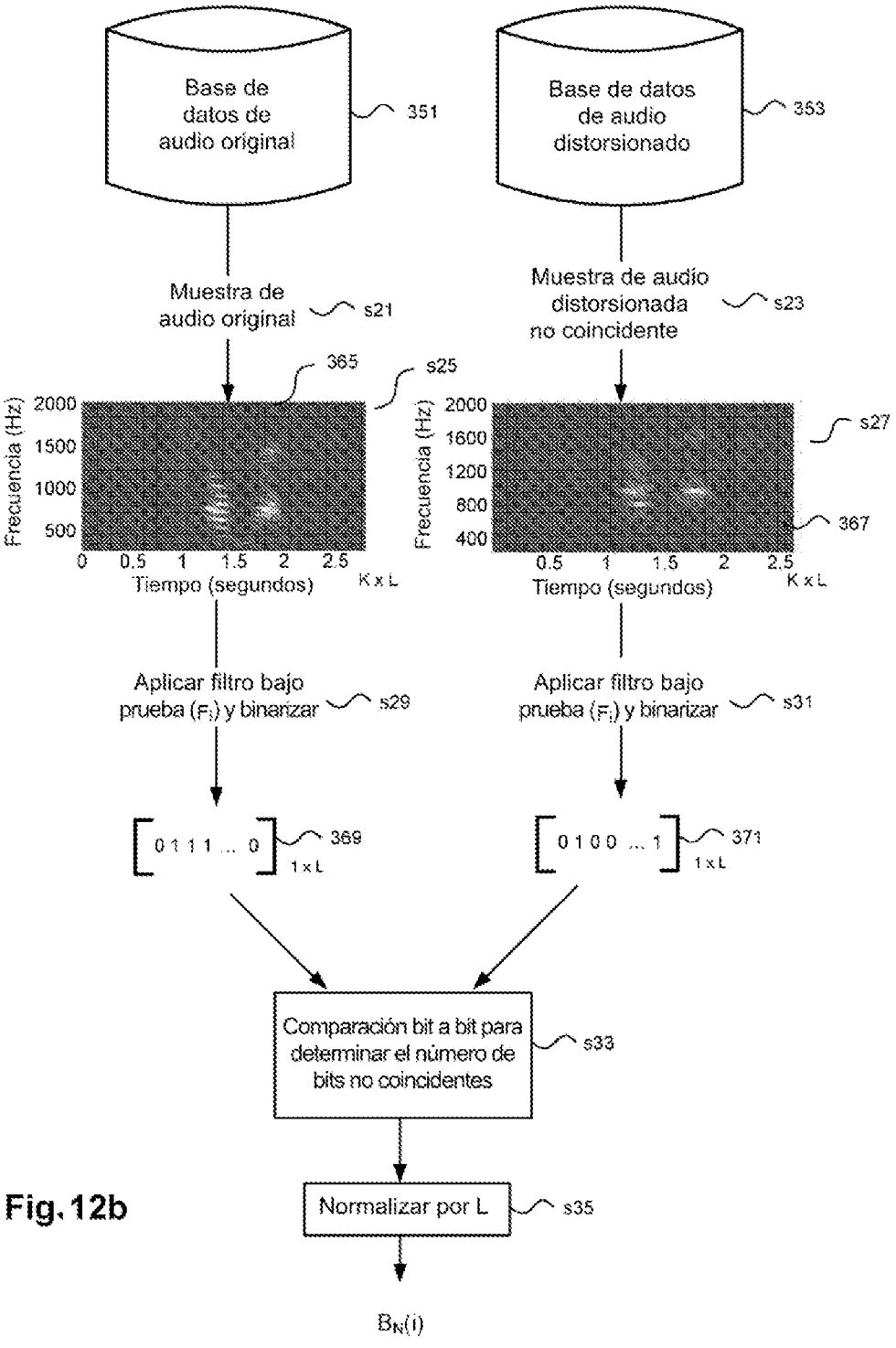
Base de datos de audio original
351
Base de datos de audio distorsionado
353
Muestra de audio original
s21
Muestra de audio distorsionada no coincidente
s23
365
s25
Frecuencia (Hz)
2000
1500
1000
500
0 0.5 1 1.5 2 2.5
Tiempo (segundos)
K x L
Frecuencia (Hz)
2000
1600
1200
800
400
0.5 1 1.5 2 2.5
Tiempo (segundos)
K x L
s27
367
Aplicar filtro bajo prueba (F_i) y binarizar
s29
Aplicar filtro bajo prueba (F_i) y binarizar
s31
0 1 1 1 ... 0
1 x L
369
0 1 0 0 ... 1
1 x L
371
Comparación bit a bit para determinar el número de bits no coincidentes
s33
Normalizar por L
s35
$B_N(i)$

Fig. 12b

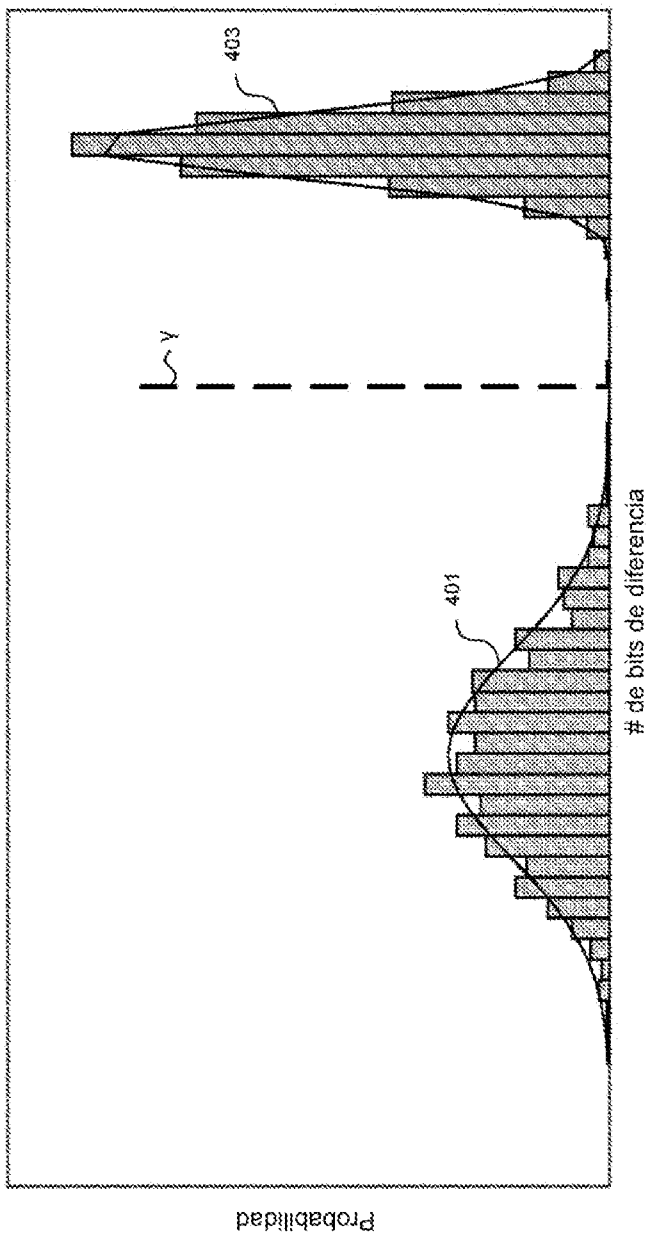

FIGURA 13a

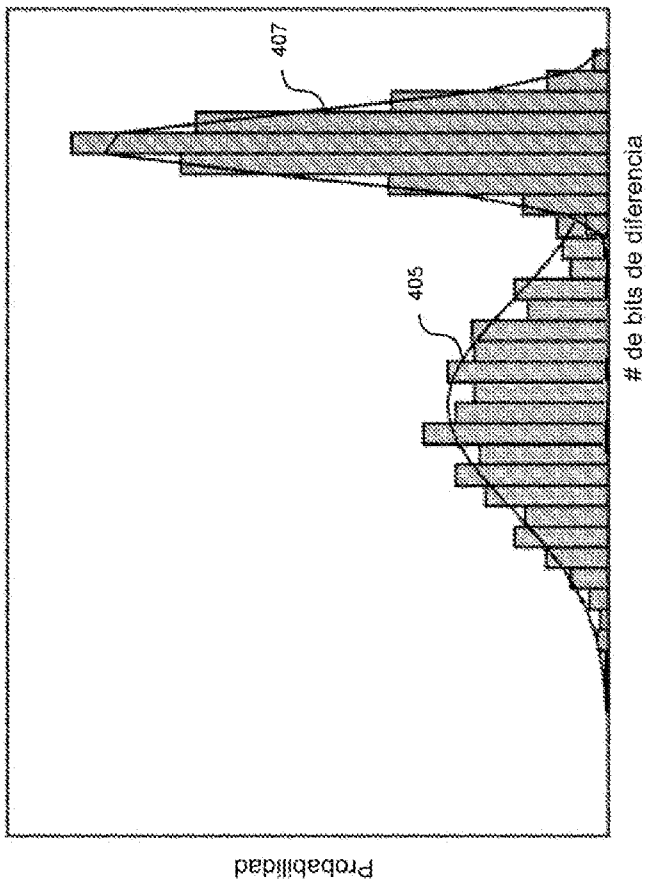

FIGURA 13b

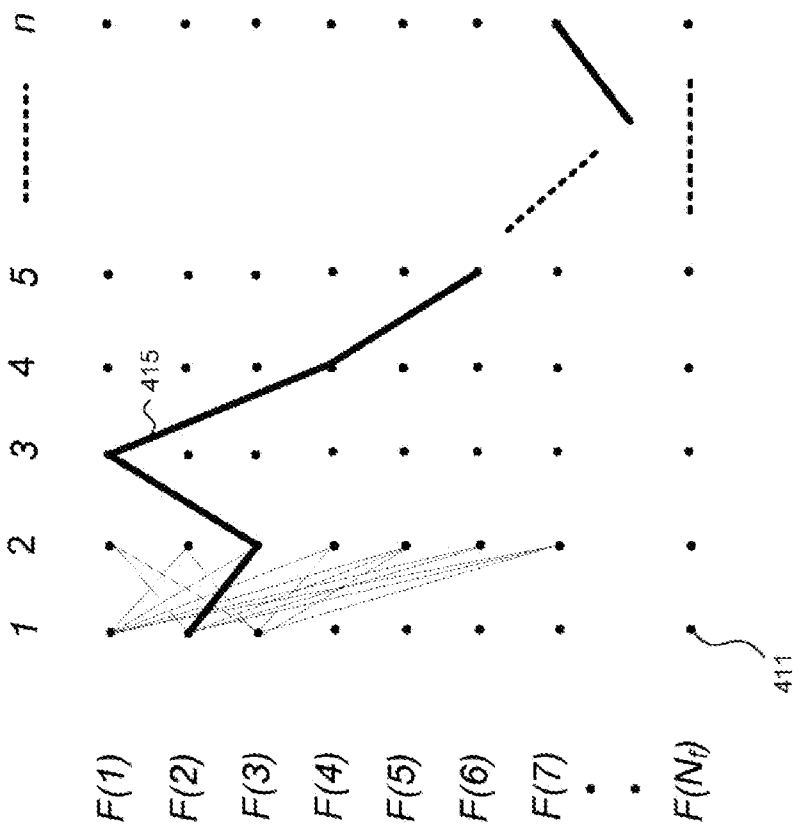

FIGURA 14

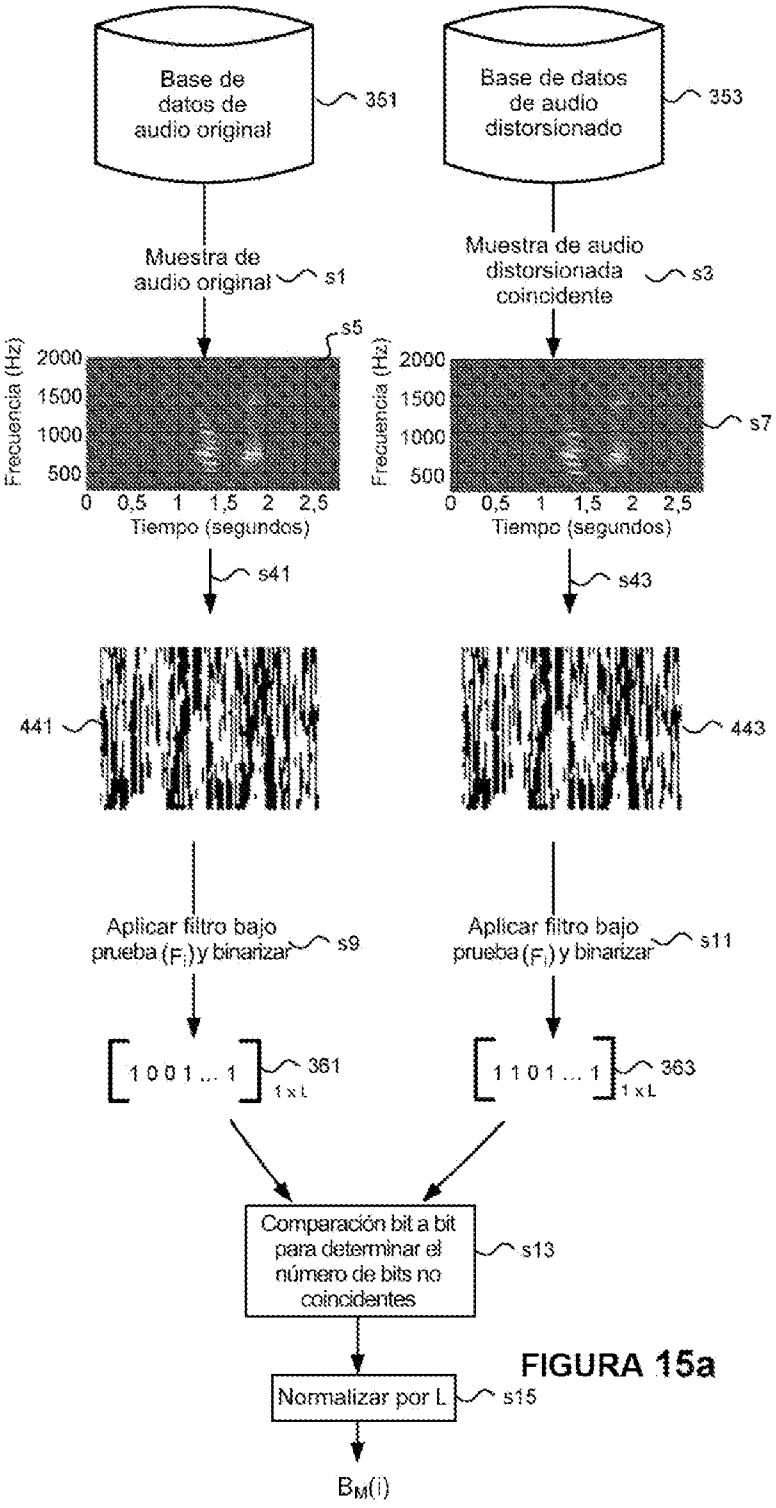
Base de datos de audio original
351
Base de datos de audio distorsionado
353
Muestra de audio original
s1
Muestra de audio distorsionada coincidente
s3
s5
s7
Frecuencia (Hz)
2000
1500
1000
500
0 0,5 1 1,5 2 2,5
Tiempo (segundos)
s41
s43
441
443
Aplicar filtro bajo prueba (F_i) y binarizar
s9
Aplicar filtro bajo prueba (F_i) y binarizar
s11
1 0 0 1 ... 1
1 x L
361
1 1 0 1 ... 1
1 x L
363
Comparación bit a bit para determinar el número de bits no coincidentes
s13
Normalizar por L
s15
$B_M(i)$

FIGURA 15a

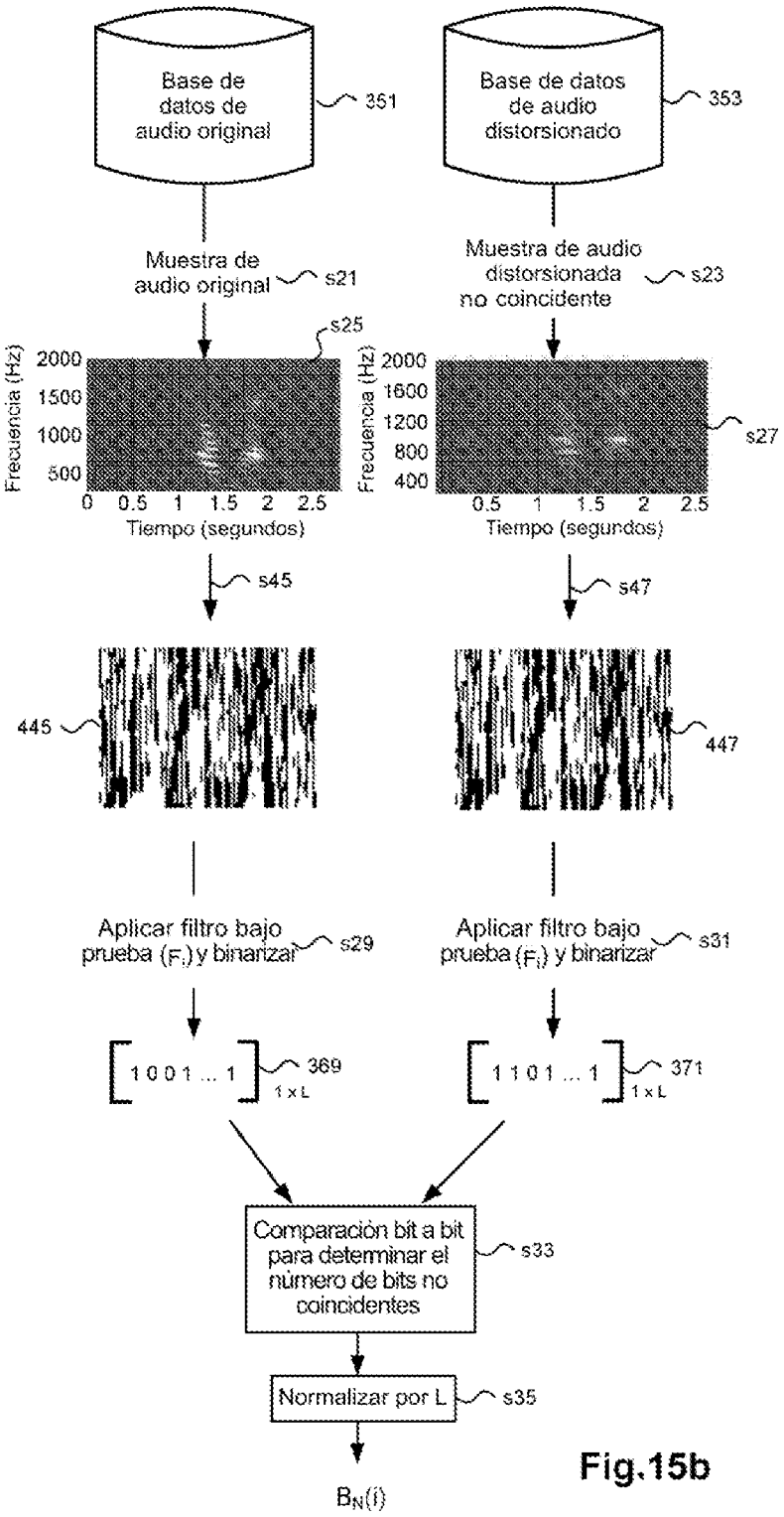
Base de datos de audio original
351
Base de datos de audio distorsionado
353
Muestra de audio original
s21
Muestra de audio distorsionada no coincidente
s23
s25
Frecuencia (Hz)
2000
1500
1000
500
0 0.5 1 1.5 2 2.5
Tiempo (segundos)
Frecuencia (Hz)
2000
1600
1200
800
400
0.5 1 1.5 2 2.5
Tiempo (segundos)
s27
s45
s47
445
447
Aplicar filtro bajo prueba (F_i) y binarizar
s29
Aplicar filtro bajo prueba (F_i) y binarizar
s31
1 0 0 1 ... 1
1 x L
369
1 1 0 1 ... 1
1 x L
371
Comparación bit a bit para determinar el número de bits no coincidentes
s33
Normalizar por L
s35
$B_N(i)$

Fig.15b