(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 459 391**

(51) Int. CI.:

G10L 25/18 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **06.06.2011 E 11725334 (4)**

(97) Fecha y número de publicación de la concesión europea: **22.01.2014 EP 2507790**

(54) Título: **Método y sistema para conseguir hashing de audio invariante al canal**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
09.05.2014

(73) Titular/es:

BRIDGE MEDIATECH, S.L. (100.0%)
Antonio Failde Gago 15-4°
32004 Ourense, ES

(72) Inventor/es:

PÉREZ GONZÁLEZ, FERNANDO;
COMESAÑA ALFARO, PEDRO;
PÉREZ FREIRE, LUIS y
PÉREZ VIEITES, DIEGO

(74) Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

DESCRIPCIÓN

Método y sistema para conseguir hashing de audio invariante al canal

Campo de la Invención

La presente invención se relaciona con el campo del procesado de audio, específicamente con el campo del hashing robusto de audio, también conocido como identificación de audio basada en contenido, hashing de audio perceptual, o fingerprinting de audio.

Antecedentes de la Invención

La identificación de contenidos multimedia, y contenidos de audio en particular, es un campo que atrae considerable atención por tratarse de una tecnología que posibilita muchas aplicaciones, abarcando desde el cumplimiento del derecho de copia o la búsqueda en bases de datos multimedia al enlazado de metadatos, sincronización de audio y vídeo, y la provisión de muchos otros servicios de valor añadido. Muchas de dichas aplicaciones se basan en la comparación de un contenido entre un audio capturado por un micrófono y los valores almacenados en una base de datos de referencia de contenidos de audio. Algunas de estas aplicaciones se ejemplifican a continuación.

Peters et al revelan en US Patent App. No. 10/749,979 un método y sistema para identificar audio ambiente capturado desde un micrófono y mostrar al usuario contenido asociado con dicho audio capturado. Métodos similares se describen en International Patent App. No. PCT/US2006/045551 (asignada a Google) para identificar audio ambiente correspondiente a un medio de difusión, presentando información personalizada al usuario en respuesta al audio identificado, y algunas aplicaciones interactivas adicionales.

US Patent App. No. 09/734,949 (asignada a Shazam) describe un método y sistema para interactuar con los usuarios, en base a una muestra proporcionada por el usuario relacionada con su entorno que es entregada a un servicio interactivo con el fin de disparar eventos, tales como (pero no limitados a) una captura de micrófono.

US Patent App. No. 11/866,814 (asignada a Shazam) describe un método para identificar un contenido capturado de un flujo de datos, que puede ser audio difundido por una fuente de difusión tal como una radio o una estación de televisión. Este método podría ser usado para identificar una canción en una emisión de radio.

Wang et al describen en US Patent App. No. 10/831,945 un método para realizar transacciones, tales como adquisiciones musicales, en base a un sonido capturado utilizando, entre otros, un método de hashing de audio robusto.

El uso de hashing robusto también es considerado por R. Reisman en US Patent App. No. 10/434,032 para aplicaciones de TV interactivas. Lu et al. consideran en US Patent App. No. 11/595,117 el uso de hashes de audio robustos para realizar mediciones de audiencias de programas de difusión.

Existen muchas técnicas para realizar identificación de audio. Cuando se tiene la certeza de que el audio a identificar y la referencia de audio existen y son iguales bit a bit, se pueden utilizar las técnicas tradicionales de hashing criptográfico para realizar búsquedas eficientes. No obstante, si las copias de audio difieren en un único bit esta aproximación falla. Otra técnica para la identificación de audio se basa en metadatos añadidos, pero no son robustos ante conversión de formatos, borrado manual de los metadatos, conversión D/A/D, etc. Cuando el audio puede ser distorsionado leve o fuertemente es obligatorio emplear otras técnicas que sean suficientemente robustas ante dichas distorsiones. Esas técnicas incluyen el watermarking y el hashing robusto de audio. Las técnicas basadas en watermarking asumen que el contenido a identificar lleva incluído cierto código (la marca de agua) que ha sido incrustado a priori. No obstante, la inserción de una marca de agua no siempre es viable, bien por razones de escalabilidad, bien por otros inconvenientes tecnológicos. Es más, dada una copia sin marca de agua de un contenido de audio el detector de la marca no puede extraer ninguna información identificativa de aquél. Por contra, las técnicas de hashing robusto de audio no necesitan ningún tipo de inserción de información en los contenidos de audio, convirtiéndolas en más universales. Las técnicas de hashing de audio robusto analizan el contenido de audio con el fin de obtener un descriptor robusto, normalmente conocido como "hash robusto" o "fingerprint", que puede ser comparado con otros descriptores almacenados en bases de datos.

Existen muchas técnicas de hashing de audio robusto. Se puede encontrar una revisión de los algoritmos más populares en el artículo de Cano et al. titulado “A review of audio fingerprinting”, Journal of VLSI Signal

Processing 41, 271-284, 2005. Algunas de las técnicas actuales persiguen identificar canciones completas o secuencias de audio, o incluso CDs o listas de reproducción. Otras técnicas permiten identificar una canción o una secuencia de audio utilizando únicamente un pequeño fragmento de ella. Normalmente estas últimas pueden adaptarse para realizar identificación en streaming, i.e. capturando fragmentos sucesivos de un flujo de audio y realizando la comparación con bases de datos de tal modo que los contenidos de referencia no están necesariamente sincronizados con aquellos que han sido capturados. Éste es, en general, el modo de operación más habitual para realizar identificación de difusiones de audio y de audio capturado por un micrófono.

La mayoría de métodos para realizar hashing de audio robusto dividen el flujo de audio en bloques contiguos de corta duración, normalmente con un grado significativo de solape. A cada uno de estos bloques se le aplican distintas operaciones con el fin de extraer características distintivas de tal modo que sean robustas frente a un conjunto dado de distorsiones. Estas operaciones incluyen, por un lado, la aplicación de transformaciones de señal como la Fast Fourier Transform (FFT), Modulated Complex Lapped Transform (MCLT), Discrete Wavelet Transform, Discrete Cosine Transform (DCT), Haar Transform o la Walsh-Hadamard Transform, entre otras. Otro procesado que es común a muchos métodos de hashing robusto es la separación de las señales de audio transformado en sub-bandas, emulando propiedades del sistema auditivo humano con el fin de extraer parámetros perceptualmente significativos. Es posible obtener algunos de dichos parámetros de las señales de audio procesadas, tales como los Mel-Frequency Cepstrum Coefficients (MFCC), Spectral Flatness Measure (SFM), Spectral Correlation Function (SCF), la energía de los coeficientes de Fourier, los centroides espectrales, la tasa de cruces por cero, etc. Por otro lado, operaciones habituales también incluyen el filtrado tiempo-frecuencia para eliminar efectos espúreos del canal y para incrementar la decorrelación, y el uso de técnicas de reducción de dimensionalidad como Principal Components Analysis (PCA), Independent Component Analysis (ICA), o la DCT.

Un método conocido para el hashing robusto de audio que se adecúa a la descripción general proporcionada anteriormente se describe en la patente Europea No. 1362485 (asignada a Philips). Los pasos de este método se pueden resumir como sigue: dividir la señal de audio en segmentos enventanados solapantes de longitud fija, calcular los coeficientes del espectrograma de la señal de audio utilizando un banco de filtros de 32 bandas en escala de frecuencias logarítmica, realizar un filtrado 2D de los coeficientes del espectrograma, y cuantificar los coeficientes resultantes con un cuantificador binario según su signo. De esta forma el hash robusto se compone de una secuencia binaria de 0s y 1s. La comparación entre dos hashes robustos tiene lugar calculando su distancia Hamming. Si dicha distancia es menor que un cierto umbral, entonces se decide que los dos hashes robustos representan la misma señal de audio. Este método proporciona un rendimiento razonablemente bueno bajo distorsiones leves, pero en general su funcionamiento empeora bajo las condiciones del mundo real. Un número significativo de trabajos posteriores ha añadido procesado adicional o modificado ciertas partes del método con el fin de mejorar su robustez ante diferentes tipos de distorsión.

El método descrito en EP1362485 ha sido modificado en la solicitud de patente internacional PCT/IB03/03658 (asignada a Philips) con el fin de obtener robustez ante cambios en la velocidad de reproducción de las señales de audio. Con el fin de tratar con los desalineamientos en los dominios del tiempo y de la frecuencia causados por cambios en la velocidad, este método introduce un paso adicional en el método descrito en EP1362485. Este paso consiste en calcular la autocorrelación temporal de los coeficientes de salida del banco de filtros, cuyo número de bandas también es incrementado de 32 a 512. Opcionalmente los coeficientes de autocorrelación pueden ser filtrados pasobajo con el fin de incrementar la robustez.

El artículo de Son et al. titulado “Sub-fingerprint Masking for a Robust Audio Fingerprinting System in a Real-noise Environment for Portable Consumer Devices”, publicado en IEEE Transactions on Consumer Electronics, vol.56, No.1, Febrero 2010, propone una mejora sobre EP1362485 consistente en calcular una máscara para el hash robusto basada en la estimación de las componentes de frecuencia fundamentales de la señal de audio que genera el hash robusto de referencia Esta máscara, que se supone que mejora la robustez frente al ruido del método revelado en EP1362485, tiene la misma longitud que el hash robusto, y puede adoptar los valores 0 ó 1 en cada posición. Para comparar dos hashes robustos primero se multiplican elemento a elemento por la máscara, y luego se calcula sus distancias Hamming como en EP1362485. Park et al. también pretenden conseguir una robustez mejorada ante el ruido en el artículo “Frequency-temporal filtering for a robust audio fingerprinting scheme in real-noise environments”, publicado en ETRI Journal, Vol. 28, No.4, 2006. En dicho artículo los autores estudian el uso de diversos filtros lineales para reemplazar el filtrado 2D empleado en EP1362485, manteniendo inalteradas el resto de componentes.

Otro método de hashing robusto de audio conocido se describe en la patente Europea No. 1307833 (asignada a Shazam). El método calcula una serie de “landmarks” o puntos de referencia (e.g. picos del espectrograma) del audio grabado, y calcula un hash robusto para cada punto de referencia. Con el fin de disminuir la probabilidad de falsa alarma, los puntos de referencia se vinculan con otros landmarks de su entorno. Por tanto, cada grabación de audio se caracteriza por una lista de pares [landmark, hash robusto]. El

método de comparación de señales de audio consiste en dos pasos. El primer paso compara los hashes robustos de cada landmark hallados en los audios de la solicitud y de referencia, y para cada coincidencia almacena un par de sus respectivas ubicaciones temporales. El segundo paso representa los pares de localizaciones temporales en un diagrama de dispersión, y se declara que hay una coincidencia entre las dos señales de audio si dicho diagrama puede ser aproximado por una recta de pendiente unidad. La US patent No. 7627477 (asignada a Shazam) mejora el método descrito en EP1307833, especialmente en lo relativo a la robustez ante cambios de velocidad y la efectividad a la hora de hacer corresponder muestras de audio.

En algunos artículos de investigación recientes, tales como el artículo de Cotton y Ellis “Audio fingerprinting to identify multiple videos of an event” in IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, y Umapathy et al. “Audio Signal Processing Using Time-Frequency Approaches: Coding, Classification, Fingerprinting, and Watermarking”, en EURASIP Journal on Advances in Signal Processing, 2010, el método de hashing robusto de audio descompone la señal de audio en diccionarios de Gabor sobrecompletos con el fin de crear una representación dispersa de la señal de audio.

Los métodos descritos en las patentes y artículos anteriores no consideran explícitamente soluciones para mitigar las distorsiones causadas por la propagación multitrayecto del audio y su ecualización, que son típicos en la identificación de audio captado por un micrófono, y que afecta muy seriamente a la eficacia de la identificación si no son tenidos en cuenta. Este tipo de distorsiones ha sido considerado en el diseño de otros métodos, que se revisan a continuación.

La patente internacional PCT/ES02/00312 (asignada a Universitat Pompeu-Fabra) revela un método de hashing robusto de audio para identificación de canciones en difusión de audio, que modela el canal desde los altavoces hasta el micrófono como un canal convolutivo. El método descrito en PCT/ES02/00312 transforma los coeficientes espectrales extraídos de la señal de audio al dominio logarítmico, con el fin de transformar el efecto del canal en uno aditivo. Después aplica un filtro lineal pasoalto en el eje temporal a los coeficientes transformados, con el fin de eliminar las variaciones lentas que se supone que son causadas por el canal convolutivo. Los descriptores extraídos para componer el hash robusto también incluyen las variaciones de energía así como las derivadas de primer y segundo orden de los coeficientes espectrales. Una diferencia importante entre este método y los métodos referidos anteriormente es que, en vez de cuantificar los descriptores, el método descrito en PCT/ES02/00312 representa los descriptores por medio de Hidden Markov Models (HMM). Los HMMs son obtenidos por medio de una fase de entrenamiento efectuada sobre una base de datos de canciones. La comparación de hashes robustos es realizada por medio del algoritmo de Viterbi. Uno de los inconvenientes de este método es el hecho de que la transformada logarítmica aplicada para eliminar la distorsión convolutiva transforma el ruido aditivo en uno de naturaleza no lineal. Esto causa que el rendimiento de la identificación se vea degradado rápidamente a medida que aumenta el nivel de ruido del audio capturado.

Otros métodos tratan de superar las distorsiones causadas por la captura del micrófono recurriendo a otras técnicas originalmente desarrolladas por la comunidad de visión por computador, tales como el aprendizaje automático. En el artículo “Computer vision for music identification”, publicado en Computer Vision and Pattern Recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, July 2005, Ke et al. generalizan el método revelado en EP1362485. Ke et al. extraen de los ficheros de música una secuencia de energías de las sub-bandas espectrales que son dispuestas en un espectrograma, que a su vez es tratado como una imagen digital. La técnica Adaboost por pares es aplicada en un conjunto de características Viola-Jones (simples filtros 2D, que generalizan el filtro utilizado en EP1362485) con el fin de aprender los descriptores locales y umbrales que mejor identifican los fragmentos musicales. El hash robusto obtenido es una cadena binaria, como en EP1362485, pero el método para comparar hashes robustos es mucho más complejo, calculando una medida de verosimilitud de acuerdo con un modelo de oclusión estimado por el método del algoritmo Expectation Maximization (EM). Tanto las características Viola-Jones seleccionadas como los parámetros del modelo EM se calculan en una fase de entrenamiento que requiere pares de señales de audio limpias y distorsionadas. El rendimiento resultante es altamente dependiente de la fase de entrenamiento, y presumiblemente también de la discrepancia entre las condiciones de entrenamiento y de captura. Es más, la complejidad del método de comparación hace que no sea aconsejable para aplicaciones de tiempo real.

En el artículo “Boosted binary audio fingerprint based on spectral subband moments”, publicado en el IEEE International Conference on Acoustics, Speech and Signal Processing, vol.1, pp.241-244, Abril 2007, Kim y Yoo siguen los mismos principios del método propuesto por Ke et al. Kim y Yoo también hacen referencia a la técnica Adaboost, pero utilizando momentos de sub-banda espectral normalizada en vez de energías de sub-banda espectral.

US patent App. No. 60/823,881 (asignada a Google) también revela un método para hacer hashing robusto basado en técnicas comúnmente utilizadas en el campo de la visión por computador, inspirado por los conocimientos proporcionados por Ke et al. No obstante, en vez de aplicar Adaboost este método aplica el análisis wavelet en 2D en el espectrograma de audio, el cual es considerado como una imagen digital. Se calcula la transformada wavelet del espectrograma, y solamente se conserva un limitado número de coeficientes significativos. Los coeficientes de las Wavelets calculadas se cuantifican de acuerdo a su signo, y se aplica la técnica de Min-Hash para reducir la dimensión del hash robusto final. La comparación de hashes robustos tiene lugar por medio de la técnica Locality-Sensitive-Hashing para que la comparación sea eficiente en bases de datos grandes y la deformación de tiempo dinámica para incrementar la robustez contra los desajustes temporales.

Otros métodos intentan incrementar la robustez contra las distorsiones de frecuencia aplicando alguna normalización a los coeficientes espectrales. El artículo escrito por Sukittanon y Atlas, “Modulation frequency features for audio fingerprinting”, presentado en IEEE International Conference of Acoustics, Speech and Signal Processing, Mayo 2002, está basado en el análisis de la modulación de frecuencia para caracterizar el comportamiento de variación temporal de la señal de audio. Una señal de audio dada primeramente se descompone en un conjunto de sub-bandas de frecuencias, y la modulación de frecuencias para cada sub-banda se estima por medio de un análisis Wavelet en diferentes escalas de tiempo. En este punto, el hash robusto de una señal de audio consiste en un conjunto de características de modulación de frecuencia en diferentes escalas de tiempo en cada sub-banda. Finalmente, para cada sub-banda de frecuencia, las funciones de modulación de frecuencia se normalizan escalándolas uniformemente mediante el sumatorio de los valores de la modulación de frecuencia calculadas para un fragmento de audio dado. Esta aproximación tiene varios inconvenientes. Por un lado, asume que la distorsión es constante a lo largo de la duración de todo el fragmento de audio. Por consiguiente, las variaciones en la ecualización o volumen que ocurren en el medio del fragmento analizado repercutirán negativamente en su eficacia. Por otro lado, para mejorar la normalización es necesario esperar hasta que todo el fragmento de audio sea recibido y se extraigan sus características. Estos inconvenientes hacen el método no recomendable para tiempo-real o aplicaciones streaming.

US patent No. 7328153 (asignada a Gracenote) describe un método de hashing robusto de audio que descompone los segmentos enventanados de las señales de audio en un conjunto de bandas espectrales. Se construye una matriz tiempo-frecuencia en cada una de las bandas espectrales. Las características utilizadas son bien los coeficientes de la DCT o bien los coeficientes wavelet para un conjunto de escalas wavelet. La normalización aproximada es muy similar en el método descrito por Sukittanon y Atlas: para mejorar la robustez contra la ecualización de frecuencia, los elementos de la matriz tiempo-frecuencia se normalizan en cada banda por el valor de la fuente principal en cada banda. La misma normalización aproximada se describe en la US patent App. No. 10/931,635.

Con el fin de mejorar la robustez contra las distorsiones, muchos métodos de hashing robusto de audio aplican en sus pasos finales un cuantificador a las características extraídas. Las características cuantificadas son también ventajosas para implementaciones de hardware simplificadas y requisitos reducidos de memoria. Habitualmente, estos cuantificadores son cuantificadores escalares binarios simples aunque los cuantificadores vectoriales, Gaussian Mixture Models y Hidden Markov Models, también se incluyen en el arte previo.

En general, y en particular cuando se utilizan cuantificadores escalares, los cuantificadores no se diseñan óptimamente para maximizar la función de identificación de los métodos de hash robusto de audio. Además, por razones computacionales, los cuantificadores escalares son preferidos dado que la cuantificación vectorial tiene un alto consumo de tiempo, especialmente cuando el cuantificador es no estructurado. El uso de cuantificadores multinivel (e.g. con más de dos celdas de cuantificación) es deseable para incrementar la precisión del hash robusto. Aún así, la cuantificación multinivel es particularmente sensible a distorsiones tales como la ecualización de frecuencia, la propagación multitrayecto y los cambios de volumen, los cuales ocurren en escenarios de identificación por grabación con micrófono. Por tanto, los cuantificadores multinivel no pueden aplicarse en estos escenarios a menos que el método de hashing sea robusto por construcción para aquellas distorsiones. Algunos trabajos describen los métodos de cuantificación escalar adaptados a la señal de entrada.

US patent App. No. 10/994,498 (asignada a Microsoft) describe un método de hashing robusto de audio que realiza el cálculo de los estadísticos de primer orden de segmentos de audio transformados mediante MCLT, realiza un paso intermedio de cuantificación utilizando un cuantificador adaptativo de N niveles obtenido del histograma de las señales, y finalmente cuantifica el resultado utilizando un decodificador con corrección de errores, el cual es una forma de cuantificador vectorial. Además, considera una aleatorización en el cuantificador dependiendo de una clave secreta.

Allamanche et al. describe en US patent App. No. 10/931,635 un método que utiliza un cuantificador escalar adaptado a la señal de entrada. En pocas palabras, el paso de cuantificación es mayor para valores de entrada de señal que ocurren con menor frecuencia, y menor para valores de señal de entrada que ocurren con mayor frecuencia.

El principal inconveniente de los métodos descritos en US patent App. No. 10/931,63 y US patent App. No. 10/994,498 es que el cuantificador optimizado es siempre dependiente de la señal de entrada, haciéndolo solamente apropiado para tratar con distorsiones leves. Una distorsión moderada o fuerte causarán que las características cuantificadas sean diferentes para el audio bajo test y el audio de referencia, incrementando de este modo la probabilidad de perder correlaciones correctas de audio.

Como se ha mencionado, los métodos existentes de hash robusto de audio presentan numerosas deficiencias que los hacen no aptos para identificación en tiempo real o audio capturado vía streaming con micrófonos. En este escenario, un esquema de hashing robusto de audio deberá cumplir varios requisitos:

• Eficiencia computacional en la generación del hash robusto. En muchos casos, la tarea de calcular los hashes robustos de audio debe efectuarse en dispositivos electrónicos realizando un número de diferentes tareas simultaneas y con poca potencia computacional (e.g. un ordenador portátil, un dispositivo móvil o un dispositivo empotrado). Por tanto, resulta especialmente interesante mantener baja la complejidad computacional en el cómputo del hash robusto.

• Eficiencia computacional en la comparación del hash robusto. En algunos casos, la comparación del hash robusto debe ejecutarse en grandes bases de datos, demandando por consiguiente algoritmos eficientes de búsqueda y detección. Un número significativo de métodos satisfacen esta característica. Sin embargo, hay otro escenario relacionado que no encaja bien en el arte anterior: un gran número de usuarios concurrentes realizando consultas a un servidor, cuando el tamaño de la base de datos de referencia no es necesariamente grande. Éste es el caso, por ejemplo, de una medición de audiencia basada en hash robusto para transmisiones de radiodifusión, o servicios interactivos basados en hash robusto, donde tanto el número de usuarios como la cantidad de consultas por segundo al servidor pueden ser muy altas. En este caso, se debe poner énfasis en la eficiencia del método de comparación antes que en el método de búsqueda. Por lo tanto, este último escenario necesita que la comparación del hash robusto sea lo más simple posible, para minimizar el número de operaciones de comparación.

• Mayor robustez para canales captados por micrófono. Cuando capturamos una transmisión de audio con micrófonos, el audio está sujeto a distorsiones como la adición de eco (debido a la propagación multitrayecto del audio), ecualización o ruido ambiental. Además, el dispositivo de captura, por ejemplo un micrófono integrado en un dispositivo electrónico como un teléfono móvil o un ordenador portátil, introduce más ruido aditivo y posiblemente distorsiones no-lineales. Por consiguiente, la relación señal a ruido (SNR) esperada en esta clase de aplicaciones es muy baja (generalmente en orden de 0 dBs o menos). Una de las principales dificultades es encontrar un método de hashing robusto que sea altamente robusto al multitrayecto y a la ecualización y cuyo rendimiento no se vea degradado críticamente por bajas SNRs. Como hemos visto, ninguno de los métodos existentes de hashing robusto es capaz de cumplir completamente este requisito.

• Fiabilidad. La fiabilidad es medida en términos de probabilidad de un falso positivo (P_{FP}) y de no detección (P_{MD}). P_{FP} mide la probabilidad de que un contenido de audio sea identificado incorrectamente, es decir, que sea relacionado con otro contenido de audio que en realidad no tiene que ver con la muestra de audio. Si P_{FP} es alto, entonces el esquema de hashing robusto de audio no es lo suficientemente discriminativo. P_{MD} mide la probabilidad de que el hash robusto extraído del contenido de una muestra de audio no encuentre ninguna correspondencia en la base de datos de hashes robustos de referencia cuando dicha correspondencia existe. Cuando P_{MD} es alto, el esquema de hashing robusto de audio se dice que no es lo suficientemente robusto. A pesar de que es deseable mantener P_{MD} tan bajo como sea posible, el coste de los falsos positivos es en general mucho más alto que el de las no detecciones. Por tanto, para muchas aplicaciones es preferible mantener la probabilidad de falsa alarma muy baja, siendo aceptable tener una probabilidad moderadamente alta de no detección.

Descripción de la Invención

La presente invención describe un método y un sistema de hashing robusto de audio, tal y como se define en las reivindicaciones. El núcleo de esta invención es un método de normalización que hace a las características extraídas de señales de audio aproximadamente invariantes a distorsiones causadas por canales captados mediante micrófono. La invención es aplicable a numerosos escenarios de identificación de

audio, pero es particularmente apropiada para la identificación captada mediante micrófono o señales de audio en streaming filtradas linealmente en tiempo real, para aplicaciones como mediciones de audiencia o la provisión de interactividad a los usuarios.

La presente invención supera los problemas identificados en el análisis del estado del arte con el fin de conseguir una rápida y fiable identificación de audio en streaming capturado en tiempo real, proporcionando un alto grado de robustez contra las distorsiones causadas por el canal de captura mediante micrófono. La presente invención extrae de las señales de audio una secuencia de vectores característicos que es altamente robusta, por construcción, ante la propagación de audio multitrayecto, ecualización de frecuencia y SNR extremadamente baja.

La presente invención comprende un método para computar hashes robustos provenientes de señales de audio, y un método para comparar hashes robustos. El método para computar un hash robusto está compuesto de tres bloques principales: transformación, normalización, y cuantificación. El bloque de transformación abarca una amplia variedad de transformaciones de señal y técnicas de reducción de dimensionalidad. La normalización está especialmente diseñada para hacer frente a las distorsiones del canal de captura mediante micrófono, mientras que la cuantificación está pensada para conseguir un alto grado de discriminación y compactación del hash robusto. El método para la comparación del hash robusto es muy simple además de eficaz.

Las principales ventajas del método revelado aquí son las siguientes:

• La computación del hash robusto es muy simple, permitiendo implementaciones ligeras en dispositivos con recursos limitados.

• Las características extraídas de las señales de audio pueden ser normalizadas sobre la marcha, sin la necesidad de esperar por grandes fragmentos de audio. Por tanto, el método es apropiado para la identificación de flujos de audio y aplicaciones en tiempo real.

• El método puede tratar con variaciones temporales en la distorsión de canal, haciéndolo muy apropiado para la identificación de audio en streaming.

• Los hashes robustos son muy compactos, y el método de comparación es muy simple, permitiendo arquitecturas servidor-cliente en escenarios de gran escala.

• Alto rendimiento en la identificación: los hashes robustos son altamente discriminativos y altamente robustos, incluso para longitudes cortas.

De acuerdo con un aspecto de la presente invención se proporciona un método de hashing robusto de audio, incluyendo un paso de extracción de hash robusto en el que se extrae un hash robusto (110) a partir de contenido de audio; dicho paso de extracción de hash robusto incluye:

- dividir el contenido de audio (102, 106) en, al menos, una trama;

- aplicar un proceso de transformación (206) en la trama para calcular, en dicha trama, un conjunto de coeficientes transformados (208);

- aplicar un proceso de normalización (212) a los coeficientes transformados (208) para obtener un conjunto de coeficientes normalizados (214), en el que dicha normalización (212) implica el cálculo del producto del signo de cada coeficiente de dichos coeficientes transformados (208) por el cociente de dos funciones homogéneas de cualquier combinación de dichos coeficientes transformados (208), donde ambas funciones homogéneas son del mismo orden;

- aplicar un procedimiento de cuantificación (220) sobre dichos coeficientes normalizados (214) para obtener el hash robusto (110) del contenido de audio (102,106).

Por ejemplo, el método implica a mayores un paso de preprocesado en el que el contenido de audio es primeramente procesado para proporcionar un contenido de audio preprocesado en un formato adecuado para el paso de extracción de hash robusto. El paso de preprocesado puede incluir alguna de las siguientes operaciones:

- conversión a formato Pulse Code Modulation (PCM);

- conversión a un único canal en el caso de audio multicanal;
- conversión de la tasa de muestreo.

El paso de extracción de hash robusto preferiblemente implica un procedimiento de enventanado para convertir las tramas en tramas enventanadas para el procedimiento de transformación.

En otro ejemplo el paso de extracción de hash robusto implica además un procedimiento de postprocesado para convertir los coeficientes normalizados en coeficientes postprocesados para el procedimiento de cuantificación. El procedimiento de postprocesado puede incluir al menos una de las siguientes operaciones:

- filtrado de otras distorsiones;
- suavizado de las variaciones de los coeficientes normalizados;
- reducción de la dimensionalidad de los coeficientes normalizados.

El procedimiento de normalización se aplica preferiblemente en los coeficientes transformados dispuestos en una matriz de tamaño F×T para obtener una matriz de coeficientes normalizados de tamaño F' × T', con F' = F, T' ≤ T, cuyos elementos Y(f', t') se calculan de acuerdo con la siguiente regla:

$$Y(f',t') = \frac{\mathrm{sign}(X(f', M(t'))) \times H(\mathbf{X}_{f'})}{G(\mathbf{X}_{f'})},$$

donde X(f', M(t')) son los elementos de la matriz de coeficientes transformados, $\mathbf{X}_{f'}$ es la fila f-ésima de la matriz de coeficientes transformados, M() es una función que hace corresponder a los índices de {1,…, T'} a índices de {1,…, T}, y tanto H() como G() son funciones homogéneas del mismo orden.

Las funciones H() y G() pueden ser obtenidas como combinación lineal de funciones homogéneas. Las funciones H() y G() pueden ser tales que el conjunto de elementos de $\mathbf{X}_{f'}$ usados en el numerador y el denominador sean disjuntos, o tales que el conjunto de elementos de $\mathbf{X}_{f'}$ usados en el numerador y el denominador sean disjuntos y correlativos. En una realización preferente las funciones homogéneas H() y G() son tales que:

$$H(\mathbf{X}_{f'}) = H(\overline{\mathbf{X}}_{f',M(t')}), \;\; G(\mathbf{X}_{f'}) = G(\underline{\mathbf{X}}_{f',M(t')}),$$

con

$$\overline{\mathbf{X}}_{f',M(t')} = [X(f', M(t')), X(f', M(t')+1), \ldots, X(f', k_u)],$$
$$\underline{\mathbf{X}}_{f',M(t')} = [X(f', k_l), \ldots, X(f', M(t')-2), X(f', M(t')-1)],$$

donde k_l es el máximo de $\{M(t')-L_l, 1\}$, k_u es el mínimo de $\{M(t')+L_u-1, T\}$, M(t')>1, y L_l>1, L_u>0.

Preferiblemente, M(t')=t'+1 y $H(\overline{\mathbf{X}}_{f',M(t')}) = \mathrm{abs}(X(f', t'+1))$, resultando en la siguiente regla de normalización:

$$Y(f',t') = \frac{X(f',t'+1)}{G(\underline{\mathbf{X}}_{f',t'+1})},$$

En una realización preferente, G() es escogida tal que

$$G(\underline{\mathbf{X}}_{f',t'+1}) = L^{-\frac{1}{p}} \times (a(1) \times |X(f',t')|^p + a(2) \times |X(f',t'-1)|^p + \ldots + a(L) \times |X(f',t'-L+1)|^p)^{\frac{1}{p}},$$

donde L_l=L, a=[a(1), a(2), …, a(L)] es un vector de ponderación y p es un número real positivo.

En otra realización preferente el procedimiento de normalización puede aplicarse a los coeficientes transformados dispuestos en una matriz de tamaño F×T para obtener una matriz de coeficientes normalizados de tamaño F' × T', con F' ≤ F, T' = T, cuyos elementos Y(f', t') son calculados de acuerdo con la siguiente regla:

$$Y(f',t') = \frac{\text{sign}(X(M(f'),t')) \times H(\mathbf{X}_{t'})}{G(\mathbf{X}_{t'})},$$

donde X(M(f'), t') son los elementos de la matriz de coeficientes transformados, $\mathbf{X}_{t'}$ es la t'-ésima columna de la matriz de coeficientes transformados, M() es una función que va del conjunto de índices {1,…, F'} al conjunto de índices {1,…, F}, y tanto H() como G() son funciones homogéneas del mismo orden.

Para realizar la normalización se puede utilizar un buffer para almacenar una matriz de coeficientes transformados pasados de contenidos de audio previamente procesados.

El procedimiento de transformación puede implicar una descomposición en sub-bandas espectrales de cada trama. El procedimiento de transformación preferiblemente implica una transformación lineal para reducir el número de coeficientes transformados. El procedimiento de transformación puede implicar a mayores dividir el espectro en al menos una banda espectral y calcular cada coeficiente transformado como la energía de la trama correspondiente en la banda espectral correspondiente.

En el procedimiento de cuantificación se puede emplear al menos un cuantificador multinivel obtenido mediante un método de entrenamiento. El método de entrenamiento para obtener los cuantificadores multinivel preferiblemente implica:

calculo de la partición: obtención de Q intervalos de cuantificación disjuntos a base de maximizar una función de coste predefinida que depende de los estadísticos de un conjunto de coeficientes normalizados calculados a partir de un conjunto de entrenamiento de fragmentos de audio; y

calculo de los símbolos: asociación de un símbolo a cada intervalo calculado.

En el conjunto de entrenamiento para obtener los cuantificadores multinivel los coeficientes calculados a partir de un conjunto de entrenamiento preferiblemente se disponen en una matriz y se optimiza un cuantificador para cada fila de dicha matriz.

Los símbolos pueden ser calculados de acuerdo con cualquiera de las siguientes reglas:

- calcular el centroide que minimiza la distorsión media para cada intervalo de cuantificación;

- asignar a cada intervalo de la partición un valor fijo de acuerdo con una modulación de pulsos en amplitud de Q niveles.

En una realización preferente la función de coste es la entropía empírica de los coeficientes cuantificados, calculada de acuerdo con la siguiente fórmula:

$$\text{Ent}(\mathcal{P}_f) = -\sum_{i=1}^{Q} (N_{i,f}/L_c) \log(N_{i,f}/L_c),$$

donde $N_{i,f}$ es el número de coeficientes de la fila f-ésima de la matriz de coeficientes postprocesados asignados al i-ésimo intervalo de la partición, y L_c es la longitud de cada fila.

Se puede emplear una medida de similitud, preferiblemente la correlación normalizada, en el paso de comparación entre el hash robusto y los hashes de referencia. El paso de comparación preferiblemente implica, para cada hash de referencia:

- extraer del correspondiente hash de referencia al menos un sub-hash con la misma longitud J que la longitud del hash robusto;

- convertir el hash robusto y cada uno de los sub-hashes en los correspondientes símbolos de reconstrucción dados por el cuantificador;

- calcular una medida de similitud de acuerdo con la correlación normalizada entre el hash robusto y cada uno de los sub-hashes de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash procesado de longitud J, h_r un sub-hash de referencia de la misma longitud J, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}}.$$
;

- comparar una función de dichas medidas de similitud con un umbral predeterminado;

- decidir, en base a dicha comparación, si el hash robusto y el hash de referencia representan el mismo contenido de audio.

Una realización preferida de la presente invención es un método para decidir si dos hashes robustos calculados mediante el anterior método de extracción de hash robusto representa el mismo contenido de audio. Dicho método comprende:

- extraer del hash más largo al menos un sub-hash con la misma longitud J que la longitud del hash más corto;

- convertir el hash más corto y cada uno de dichos sub-hashes en los correspondientes símbolos de reconstrucción dados por el cuantificador;

- calcular una medida de similitud de acuerdo con la correlación normalizada entre el hash más corto y cada uno de dichos sub-hashes de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash de la solicitud de longitud J, h_r un sub-hash de referencia de la misma longitud J, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}}.$$
;

- comparar una función (preferiblemente el máximo) de dicha medida de similitud con un umbral predefinido;

- decidir, en base a dicha comparación, si dos hashes robustos representan el mismo contenido de audio.

De acuerdo con otro aspecto de la presente invención se proporciona un sistema de hashing robusto de audio, caracterizado por que comprende un módulo de extracción de hash robusto (108) para extraer un hash robusto (110) a partir de contenido de audio (102,106), el módulo de extracción de hash robusto (108) comprendiendo medios de procesamiento de datos configurados para:

- la división del contenido de audio (102, 106) en al menos una trama;

- la aplicación de un proceso de transformación (206) sobre dichas tramas para calcular, para cada una de ellas, un conjunto de coeficientes transformados (208);

- la aplicación de un proceso de normalización (212) sobre los coeficientes transformados (208) para obtener un conjunto de coeficientes normalizados (214), donde dicho proceso de normalización (212) comprende el cálculo del producto del signo de cada coeficiente transformado (208) por el cociente de dos funciones homogéneas de cualquier combinación de los coeficientes transformados (208), donde ambas funciones homogéneas son del mismo orden;

- la aplicación de un proceso de cuantificación (220) en dichos coeficientes normalizados (214) para obtener un hash robusto (110) del contenido de audio (102, 106).

Una realización preferida de la presente invención es un sistema para decidir si dos hashes robustos calculados mediante el anterior sistema de hashing robusto de audio representan el mismo contenido de audio. Dicho sistema comprende medios de procesamiento de datos configurados para:

- la extracción del hash más largo de al menos un sub-hash con la misma longitud J que la longitud del hash más corto;

- la conversión del hash más corto y cada uno de dichos sub-hashes en los correspondientes símbolos de reconstrucción dados por el cuantificador;

- el cálculo de una medida de similitud de acuerdo con la correlación normalizada entre el hash más corto y cada uno de dichos sub-hashes de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash a comparar (110) de longitud J, h_r un sub-hash de referencia de la misma longitud J, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}};$$

- la comparación de una función de dicha media de similitud con un umbral predefinido;

- la decisión, en base a dicha comparación, de si dos hashes robustos representan el mismo contenido de audio.

Breve Descripción de las Figuras

A continuación se describe brevemente un conjunto de figuras que ayudan a entender mejor la invención. Las descripciones se presentan relacionadas con la realización de dicha invención.

La Fig. 1 muestra un diagrama de bloques esquemático de un sistema de hashing robusto de acuerdo con la presente invención.

La Fig. 2 es un diagrama de bloques representando el método para calcular un hash robusto a partir de un contenido de audio de muestra.

La Fig. 3 ilustra el método para comparar un hash robusto extraído de un fragmento de un contenido de audio con un hash dado contenido en una base de datos.

La Fig. 4 es un diagrama de bloques representando el método de normalización.

La Fig. 5 ilustra las propiedades de la normalización utilizada en la presente invención.

La Fig. 6 es un diagrama de bloques ilustrando el método para entrenar el cuantificador.

La Fig. 7 muestra la Receiver Operating Characteristic (ROC) para la realización preferente.

La Fig. 8 muestra P_{FP} y P_{MD} para la disposición preferente.

La Fig. 9 es un diagrama de bloques ilustrando la realización de la invención para identificar un flujo de audio.

La Fig. 10 muestra gráficas de la probabilidad de una ejecución correcta y las diferentes probabilidades de error al usar la realización de la invención para identificar un flujo de audio.

Descripción de una Realización Preferente de la Invención

La Fig. 1 representa el diagrama de bloques general de un sistema de identificación de audio basado en hashing robusto de audio de acuerdo con la presente invención. El contenido de audio **102** puede originarse en cualquier fuente: puede ser un fragmento extraído de un fichero de audio obtenido de cualquier sistema de almacenamiento, una captura de micrófono de una transmisión de difusión (radio o TV, por ejemplo), etc. El contenido de audio **102** es preprocesado por un módulo de preprocesado **104** con el fin de proporcionar un contenido de audio preprocesado **106** en un formato que pueda entregarse al módulo de extracción de hash robusto **108**. Las operaciones realizadas por el módulo de preprocesado **104** incluyen lo siguiente: conversión a formato Pulse Code Modulation (PCM), conversión a un único canal en el caso de audio multicanal, y conversión de la tasa de muestreo si fuese necesaria. El módulo de extracción de hash robusto **108** analiza el contenido de audio preprocesado **106** para extraer el hash robusto **110**, que es un vector de características distintivas que es usado en el módulo de comparación **114** para encontrar posibles correspondencias. El módulo de comparación **114** compara el hash robusto **110** con los hashes de referencia almacenados en una base de datos de hashes **112** para encontrar posibles correspondencias.

En un primer ejemplo, la invención realiza la identificación de un contenido de audio dado extrayendo de dicho contenido de audio un vector de características que pueda ser comparado con otros hashes robustos de referencia almacenados en una base de datos dada. Con el fin de realizar dicha identificación se procesa el contenido de audio de acuerdo con el método mostrado en la Fig. 2. El contenido de audio preprocesado **106** se divide primeramente en tramas solapantes $\{fr_t\}$, con $1 \leq t \leq T$, de tamaño N muestras $\{s_n\}$, con $1 \leq n \leq N$. El grado de solapamiento debe ser significativo, con el fin de hacer que el hash sea robusto ante desalineamientos temporales. El número total de tramas, T, dependerá de la longitud del contenido de audio preprocesado **106** y el grado de solapamiento. Como es habitual en procesado de audio, cada trama se multiplica por una ventana predefinida –procedimiento de enventanado **202** (e.g. Hamming, Hanning, Blackman, etc.)–, con el fin de reducir los efectos del entramado en el dominio de la frecuencia.

En el siguiente paso, las tramas enventanadas **204** experimentan un proceso de transformación **206** que transforma dichas tramas en una matriz de coeficientes transformados **208** de tamaño F×T. Más específicamente, se calcula un vector de F coeficientes transformados para cada trama y se disponen en un vector columna. Por tanto, la columna de la matriz de los coeficientes transformados **208** de índice t, con $1 \leq t \leq T$, contiene todos los coeficientes transformados para la trama con ese mismo índice temporal. De forma similar, la fila de índice f, con $1 \leq f \leq F$, contiene la evolución temporal del coeficiente transformado de ese mismo índice f. El cálculo de los elementos X(f, t) de la matriz de coeficientes transformados **208** se explica a continuación. De forma opcional, la matriz de coeficientes transformados **208** puede ser almacenada completa o en parte en un buffer **210**. La utilidad de dicho buffer **210** se muestra a continuación en la descripción de otra realización de la siguiente invención.

A los elementos de la matriz de coeficientes transformados **208** se les aplica un procedimiento de normalización **212** que es clave para asegurar el buen comportamiento de la presente invención. La normalización considerada en esta invención está destinada a crear una matriz de coeficientes normalizados **214** de tamaño F'×T', donde $F' \leq F$, $T' \leq T$, con elementos Y(f', t'), más robusta ante distorsiones causadas por canales de captura con micrófono. La distorsión más importante en estos canales proviene de la propagación multitrayecto del audio, que introduce ecos, produciendo distorsiones importantes en el audio capturado.

Además, la matriz de coeficientes normalizados **214** es la entrada de un procedimiento de postprocesado **216** que se puede aplicar, por ejemplo, para filtrar otras distorsiones, suavizar las variaciones en la matriz de coeficientes normalizados **214**, o reducir su dimensionalidad utilizando Principal Component Analysis (PCA), Independent Component Analysis (ICA), la Discrete Cosine Transform (DCT), etc. Los coeficientes postprocesados así obtenidos se disponen en una matriz de coeficientes postprocesados **218**, aunque posiblemente de menor tamaño que el de la matriz de coeficientes normalizados **214**.

Finalmente, a los coeficientes postprocesados **218** se les aplica un procedimiento de cuantificación **220**. Los objetivos de la cuantificación son dos: producir un hash más compacto, e incrementar la robustez frente al ruido. Por las razones explicadas anteriormente es preferible que el cuantificador sea escalar, es decir, que cuantifique cada coeficiente de forma independiente a los demás. Al contrario que muchos cuantificadores utilizados en métodos de hashing robusto existentes, el cuantificador utilizado en esta invención no es necesariamente binario. En efecto, el mejor funcionamiento de la presente invención se obtiene utilizando un cuantificador multinivel, lo que hace el hash más discriminativo. Como se ha explicado anteriormente, una condición para la efectividad de dicho cuantificador multinivel es que la entrada debe ser (al menos aproximadamente) invariante a distorsiones causadas por la propagación multitrayecto. Por tanto, la normalización **212** es clave para garantizar el buen funcionamiento de la invención.

Se aplica el proceso de normalización **212** en los coeficientes transformados **208** para obtener una matriz de coeficientes normalizados **214**, que en general es de tamaño F'×T'. La normalización **212** implica calcular el producto del signo de cada coeficiente de dicha matriz de coeficientes transformados **208** por una función invariante al escalado en amplitud de alguna combinación de dicha matriz de coeficientes transformados **208**.

En una realización preferente, la normalización **212** produce una matriz de coeficientes normalizados **214** de tamaño F' × T', con F' = F, T' ≤ T, cuyos elementos se calculan de acuerdo con la siguiente regla:

$$Y(f',t') = \frac{\text{sign}(X(f', M(t'))) \times H(\mathbf{X}_{f'})}{G(\mathbf{X}_{f'})}, \quad (1)$$

donde $\mathbf{X}_{f'}$ es la f'-ésima fila de la matriz de coeficientes transformados **208**, M() es una función que hace corresponder a índices del intervalo {1,…, T'} otros índices del intervalo {1,…, T}, es decir, se encarga de cambios en los índices de la trama debidos a la posible reducción del número de tramas, y tanto H() como G() son funciones homogéneas del mismo orden. Una función homogénea de orden n es una función que, para cualquier número positivo ρ, satisface la siguiente relación:

$$G(\rho \mathbf{X}_{f'}) = \rho^n G(\mathbf{X}_{f'}). \quad (2)$$

El objetivo de la normalización es hacer los coeficientes Y(f', t') invariantes al escalado. Esta propiedad de invarianza realmente mejora la robustez ante distorsiones tales como la propagación multitrayecto del audio y la ecualización en frecuencia. De acuerdo con la ecuación (1), la normalización del elemento X(f, t) usa sólo elementos de la misma fila f de la matriz de coeficientes transformados **208**. No obstante, esta realización no debería ser considerada como limitante, dado que en un escenario más general la normalización **212** podría usar cualquier elemento de la matriz completa **208**, como se explica a continuación.

Existen numerosas realizaciones de la normalización que se adecúan a los propósitos contemplados. En cualquier caso, las funciones H() y G() deben ser escogidas adecuadamente de tal manera que la normalización sea efectiva. Una posibilidad es hacer que los conjuntos de elementos de $\mathbf{X}_f$ usados en el numerador y denominador sean disjuntos. Existen múltiples combinaciones de elementos que cumplen esta condición. Una de ellas viene dada mediante la siguiente elección:

$$H(\mathbf{X}_{f'}) = H(\overline{\mathbf{X}}_{f',M(t')}), \quad G(\mathbf{X}_{f'}) = G(\underline{\mathbf{X}}_{f',M(t')}), \quad (3)$$

con

$$\overline{\mathbf{X}}_{f',M(t')} = [X(f', M(t')), X(f', M(t')+1), \ldots, X(f', k_u)], \quad (4)$$

$$\underline{\mathbf{X}}_{f',M(t')} = [X(f', k_l), \ldots, X(f', M(t')-2), X(f', M(t')-1)], \quad (5)$$

donde k_l es el máximo de $\{M(t')-L_l,1\}$, k_u es el mínimo de $\{M(t')+L_u-1,T\}$, $M(t')>1$, y $L_l>1$, $L_u>0$. Con esta elección, se usan a lo sumo L_u elementos de $\mathbf{X}_{f'}$ en el numerador de (1), y a lo sumo L_l elementos de $\mathbf{X}_{f'}$ en el denominador. Además, no sólo los conjuntos de coeficientes usados en numerador y denominador son disjuntos, sino que también son correlativos. Otra ventaja fundamental de la normalización usada en estos conjuntos de coeficientes es que se adapta dinámicamente a variaciones temporales del canal de captura del micrófono, dado que la normalización sólo tiene en cuenta los coeficientes en una ventana deslizante de duración L_l+L_u.

La Fig. 4 muestra un diagrama de bloques de la normalización de acuerdo con esta realización, en la que la función de correspondencia se fija a M(t') = t'+1. Un buffer de coeficientes pasados **404** almacena los L_l elementos de la f-ésima fila **402** de la matriz de coeficientes transformados **208** de X(f', t'+1−L_l) a X(f', t'), y es la entrada de la función G() **410**. De forma similar, un buffer de coeficientes futuros **406** almacena los L_u elementos de X(f', t'+1) a X(f', t'+L_u) y es la entrada de la función H() **412**. La salida de la función H() se multiplica por el signo del coeficiente actual X(f', t'+1) calculado en **408**. El número resultante se divide finalmente por la salida de la función G() **412**, obteniéndose el coeficiente normalizado Y(f', t').

Si las funciones H() y G() se escogen adecuadamente, a medida que L_l y L_u aumentan la variación de los coeficientes Y(f',t') se hace más suave, incrementando de esta manera la robustez frente al ruido, el cual es otro objetivo perseguido por la presente invención. El inconveniente de incrementar L_l y L_u es que el tiempo para adaptarse a los cambios en el canal se incrementa igualmente. Se da, por tanto, un compromiso entre el tiempo de adaptación y la robustez frente al ruido. Los valores óptimos de L_l y L_u dependen de la SNR esperada y de la tasa de variación del canal de captura del micrófono.

En un caso concreto de normalización, la ecuación (1), que es particularmente útil para aplicaciones de flujo de audio, se obtiene fijando $H(\underline{\mathbf{X}}_{f',M(t')}) = \mathrm{abs}(X(f',t'+1))$, obteniéndose

$$Y(f',t') = \frac{X(f',t'+1)}{G(\underline{\mathbf{X}}_{f',t'+1})}, \qquad (6)$$

con L_l=L. Por tanto, la normalización hace que el coeficiente Y(f', t') sea dependiente de, a lo sumo, L muestras de audio pasadas. Aquí el denominador G($\underline{\mathbf{X}}_{f',t'+1}$) puede ser considerado como un factor de normalización. A medida que L aumenta, el factor de normalización varía más suavemente, aumentando a su vez el tiempo para adaptarse a los cambios del canal. La realización de la ecuación (6) es particularmente adecuada para aplicaciones de tiempo real, dado que puede realizarse sobre la marcha fácilmente a medida que se procesan las tramas del fragmento de audio, sin necesidad de esperar por el procesado del fragmento completo de tramas futuras.

Una familia particular de funciones homogéneas de orden 1 que es adecuada para realizaciones prácticas es la familia de normas p ponderadas, que se ejemplifica aquí para G($\underline{\mathbf{X}}_{f',t'+1}$):

$$G(\underline{\mathbf{X}}_{f',t'+1}) = L^{-\frac{1}{p}} \times (a(1) \times |X(f',t')|^p + a(2) \times |X(f',t'-1)|^p + \ldots + a(L) \times |X(f',t'-L+1)|^p)^{\frac{1}{p}}, \qquad (7)$$

donde **a**=[a(1), a(2), …, a(L)] es el vector de ponderación, y p puede adoptar cualquier valor positivo (no necesariamente un entero). El parámetro p puede ser ajustado para optimizar la robustez del sistema de hashing robusto. El vector de pesos se puede usar para ponderar los coeficientes del vector $\underline{\mathbf{X}}_{f,t'+1}$ de acuerdo por ejemplo a una métrica de fiabilidad dada como sus amplitudes (aquellos coeficientes con menor amplitud podrían tener menos peso en la normalización, puesto que se consideran poco fiables). Otro uso del vector de pesos es implementar un factor de olvido. Por ejemplo, si **a** = [γ, γ^2, γ^3, …,γ^L], con |γ| < 1, el peso de los coeficientes en la ventana de normalización decae exponencialmente a medida que se alejan en el tiempo. El factor de olvido se puede usar para aumentar la longitud de la ventana de normalización sin hacer demasiado lenta la adaptación a cambios en el canal de captura de micrófono.

En otra realización, las funciones H() y G() se obtienen como combinación lineal de funciones homogéneas. Un ejemplo compuesto de la combinación de normas p ponderadas para la función G() se muestra a continuación:

$$G(\underline{\mathbf{X}}_{f,t}) = w_1 \times G_1(\underline{\mathbf{X}}_{f,t}) + w_2 \times G_2(\underline{\mathbf{X}}_{f,t}), \qquad (8)$$

donde

$$G_1(\underline{X}_{f,t}) = L^{-\frac{1}{p_1}} \times (a_1(1) \times |X(f,t-1)|^{p_1} + a_1(2) \times |X(f,t-2)|^{p_1} + \ldots + a_1(L) \times |X(f,t-L)|^{p_1})^{\frac{1}{p_1}}, \qquad (9)$$

$$G_2(\underline{X}_{f,t}) = L^{-\frac{1}{p_2}} \times (a_2(1) \times |X(f,t-1)|^{p_2} + a_2(2) \times |X(f,t-2)|^{p_2} + \ldots + a_2(L) \times |X(f,t-L)|^{p_2})^{\frac{1}{p_2}}, \qquad (10)$$

donde w_1 y w_2 son factores de ponderación. En este caso, los elementos de los vectores de pesos $\mathbf{a}_1$ y $\mathbf{a}_2$ sólo adoptan valores 0 ó 1, de tal manera que $\mathbf{a}_1 + \mathbf{a}_2 = [1, 1, \ldots, 1]$. Esto es equivalente a dividir los coeficientes $\underline{\mathbf{X}}_{f,t}$ en dos conjuntos disjuntos, de acuerdo con aquellos índices de $\mathbf{a}_1$ y $\mathbf{a}_2$ que tomen el valor 1. Si $p_1<p_2$, entonces los coeficientes indexados por $\mathbf{a}_1$ tienen menos influencia en la normalización. Esta característica es útil para reducir el impacto negativo de coeficientes poco fiables, tales como aquellos con amplitudes pequeñas. Los valores óptimos para los parámetros w_1, w_2, p_1, p_2, $\mathbf{a}_1$ y $\mathbf{a}_2$ pueden obtenerse por medio de técnicas de optimización habituales.

Todas las realizaciones de la normalización **212** que han sido descritas anteriormente se ciñen a la ecuación (1), es decir, la normalización tiene lugar sobre la matriz de coeficientes transformados **208**. En otra realización, la normalización se efectúa por columnas para llegar a una matriz de coeficientes normalizados de tamaño F'×T', con F' ≤ F, T' = T. De forma similar a la ecuación (1), los elementos normalizados se calculan como:

$$Y(f',t') = \frac{\text{sign}(X(M(f'),t')) \times H(\mathbf{X}_{t'})}{G(\mathbf{X}_{t'})},$$

donde $\mathbf{X}_{t'}$ es la columna t'-ésima de la matriz de coeficientes transformados **208**, M() es una función que hace corresponder índices del conjunto {1,…, F'} al conjunto {1,…, F}, es decir, se encarga de los cambios en los coeficientes transformados debidos a la posible reducción en el número de coeficientes transformados por trama, y tanto H() como G() son funciones homogéneas del mismo orden. Un caso en que la aplicación de esta normalización es particularmente útil es aquél en que el contenido de audio puede estar sujeto a cambios de volumen. En el caso límite T=1 (es decir, el contenido de audio total es tomado como una trama) la matriz resultante de coeficientes transformados **208** es un vector columna F-dimensional, y esta normalización puede hacer que los coeficientes normalizados sean invariantes ante cambios de volumen.

Hay numerosas realizaciones de la transformación **206** que pueden aprovechar las propiedades de la normalización descritas anteriormente. De acuerdo con un ejemplo de realización de la invención, cada coeficiente transformado es considerado un coeficiente de la DFT. La transformación **206** simplemente calcula la transformada discreta de Fourier (DFT) de tamaño M_d para cada trama enventanada **204**. Para un conjunto de índices DFT en un rango predeterminado de i_1 a i_2 se calcula su módulo al cuadrado. El resultado es almacenado en cada elemento X(f, t) de la matriz de coeficientes transformados **208**, que se puede ver como una matriz tiempo-frecuencia. Por tanto, $X(f, t) = |v(f,t)|^2$, con v(f, t) el coeficiente DFT de la trama t en el índice de frecuencia f. Si X(f, t) es un coeficiente de la matriz tiempo-frecuencia obtenido a partir del contenido de un audio de referencia, y X*(f, t) es el coeficiente obtenido a partir del mismo contenido distorsionado por propagación multitrayecto del audio, entonces se cumple que

$$X^*(f,t) \approx C_f \times X(f,t),\ 1 \le t \le T \qquad (11)$$

donde C_f es una constante dada por el cuadrado de la amplitud del canal multitrayecto en la frecuencia de índice f. La aproximación de (11) se deriva del hecho de que la transformación **206** funciona con tramas del contenido de audio, lo que hace que la propagación multitrayecto no pueda ser modelada exactamente como un efecto puramente multiplicativo. Por tanto, como resultado de la normalización **212**, resulta evidente que la salida Y(f', t') **214**, obtenida de acuerdo con la fórmula (1), es aproximadamente invariante ante distorsiones causadas por la propagación multitrayecto del audio, dado que las dos funciones, H() en el numerador, y G() en el denominador son homogéneas del mismo orden y por tanto $C_{f'}$ prácticamente se cancela para cada frecuencia de índice f'. En la Fig. 5 se muestra un gráfico de dispersión **52** de X(f, t) frente a X*(f, t) para un índice de la DFT. Esta realización no es la más ventajosa, dado que realizar la normalización en todos los canales DFT es costoso debido al hecho de que el tamaño de la matriz de coeficientes transformados **208** será, en general, grande. Por tanto, es preferible realizar la normalización en un número reducido de coeficientes transformados.

De acuerdo con un ejemplo de realización de la invención, la transformación **206** divide el espectro en un número predeterminado M_b de bandas espectrales, posiblemente solapadas. Cada coeficiente transformado X(f, t) se calcula como la energía de la trama t en la correspondiente banda f, con $1 \le f \le M_b$. Por tanto, en esta realización los elementos de la matriz de coeficientes transformados **208** vienen dados por

$$X(f,t) = \sum_{i=1}^{M_d} e_f(i) \times v_t(i), \qquad (12)$$

que en notación matricial se puede escribir de forma compacta como X(f,t) = $\mathbf{e}_f^T\mathbf{v}_t$, donde:

- $\mathbf{v}_t$ es un vector con los coeficientes DFT de la trama de audio t,

- $\mathbf{e}_f$ es un vector con todos los elementos puestos a 1 en aquellos índices que se correspondan con la banda espectral f, y 0 en caso contrario.

Esta segunda realización se puede ver como una forma de reducción de dimensionalidad por medio de una transformación lineal aplicada sobre la primera realización. Esta transformación lineal se define mediante la matriz de proyección

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{M_b}]. \qquad (13)$$

Por tanto se construye una matriz de coeficientes transformados más pequeña **208**, en la que cada elemento es ahora la suma de un subconjunto dado de los elementos de la matriz de coeficientes transformados construida en la realización anterior. En el caso límite en el que M_b=1, la matriz resultante de coeficientes transformados **208** es un vector fila T-dimensional, donde cada elemento es la energía de la correspondiente trama.

Tras la distorsión de un canal multitrayecto, los coeficientes de la matriz de coeficientes transformados **208** son multiplicados por las correspondientes ganancias del canal en cada banda espectral. En notación matricial, X(f,t) ≈ $\mathbf{e}_f^T\mathbf{D}\mathbf{v}_t$, donde **D** es la matriz diagonal cuya diagonal principal viene dada por el módulo al cuadrado de los coeficientes DFT del canal multitrayecto. Si la variación de la magnitud de la respuesta en frecuencia del canal multitrayecto en el rango de cada banda espectral no es demasiado abrupto, entonces se cumple la condición (11) y por tanto se asegura la invarianza aproximada ante la distorsión multitrayecto. Si la respuesta en frecuencia es abrupta, como es habitual en el caso de canales multitrayecto, entonces es preferible aumentar la longitud de las ventanas de normalización L_l y L_u con el fin de mejorar la robustez frente al multitrayecto. Al usar la normalización (6) y la definición (7) de la función G() para p=2 y **a** = [1, 1, …, 1], entonces G($\underline{\mathbf{X}}_{f,t}$) es la potencia del coeficiente transformado de índice f (que en este caso se corresponde con la f-ésima banda espectral) promediada en las últimas L tramas. En notación matricial, esto puede escribirse como

$$G(\mathbf{X}_{f,t}) = \left(\mathbf{e}_f^T\left(\frac{1}{L}\sum_{i=1}^{L}\mathbf{v}_{t-i}\mathbf{v}_{t-i}^T\right)\mathbf{e}_f\right)^{\frac{1}{2}} = \left(\mathbf{e}_f^T\mathbf{R}_t\mathbf{e}_f\right)^{\frac{1}{2}}. \qquad (14)$$

Si el contenido de audio se ve distorsionado por un canal multitrayecto, entonces

$$G(\mathbf{X}^*_{f,t}) \approx \left(\mathbf{e}_f^T\left(\mathbf{D}\mathbf{R}_t\mathbf{D}\right)\mathbf{e}_f\right)^{\frac{1}{2}}. \qquad (15)$$

Cuanto más grande sea L, más estables son los valores de la matriz $\mathbf{R}_t$, y mejor es por tanto el funcionamiento del sistema. En la Fig. **¡Error! No se encuentra el origen de la referencia.** se muestra un gráfico de dispersión **54** de Y(f', t') frente a Y*(f', t') obtenido con L=20 para una determinada banda f y la función G mostrada en (7). Como se puede apreciar, los valores representados se concentran todos alrededor de la pendiente unidad, mostrando así la propiedad de cuasi-invarianza conseguida mediante la normalización.

En otra realización, la transformación **206** aplica una transformación lineal que generaliza la descrita en la realización anterior. Esta transformación lineal considera una matriz de proyección arbitraria **E**, que puede ser generada aleatoriamente por medio de PCA, ICA o algún procedimiento de reducción de dimensionalidad similar. En cualquier caso, esta matriz no depende de cada matriz de entrada de coeficientes transformados **208** sino que se calcula de antemano, por ejemplo durante la fase de entrenamiento. El objetivo de esta transformación lineal es realizar una reducción de dimensionalidad en la matriz de coeficientes transformados, que según las realizaciones previas podría componerse de los cuadrados de los módulos de los coeficientes de la DFT $\mathbf{v}_t$ o la energía de las bandas espectrales según la ecuación (12). La última opción

es preferible en general, dado que el método, especialmente en su fase de entrenamiento, es computacionalmente más asequible puesto que el número de bandas espectrales es normalmente mucho menor que el número de coeficientes de la DFT. Los coeficientes normalizados **214** tienen propiedades similares a aquellas mostradas en realizaciones anteriores. En la Fig. **¡Error! No se encuentra el origen de la referencia.** la gráfica de dispersión **56** muestra Y(f', t') frente a Y*(f', t') para una banda dada f cuando $G(\underline{\mathbf{X}}_{f,t})$ se ajusta según la ecuación (7), L=20, y la matriz de proyección **E** se obtiene por medio de PCA. Esto muestra de nuevo la propiedad de cuasi invarianza conseguida por la normalización.

En otra realización preferente, el bloque de transformación **206** simplemente calcula la transformada DFT de las tramas de audio enventanadas **204**, y el resto de operaciones se posponen hasta la etapa de postprocesado **216**. No obstante, es preferible realizar la normalización **212** en una matriz de coeficientes transformados tan pequeña como sea posible para ahorrar operaciones. Además, realizar la reducción de dimensionalidad antes de la normalización tiene el efecto positivo de eliminar componentes que son demasiado sensibles al ruido, mejorando de esta manera la efectividad de la normalización y el rendimiento del sistema global.

Son posibles otras realizaciones con diferentes transformaciones **206**. Otro ejemplo de realización efectúa las mismas operaciones que las realizaciones descritas anteriormente, pero reemplazando la DFT por la transformada discreta del coseno (DCT). La correspondiente gráfica de dispersión **58** se muestra en la Fig. 5 cuando $G(\underline{\mathbf{X}}_{f,t})$ se asigna de acuerdo con la ecuación (7), L=20, p=2, y la matriz de proyección viene dada por la matriz mostrada en (13). La transformada puede ser también la transformada wavelet discreta (DWT). En este caso, cada fila de la matriz de coeficientes transformados **208** se correspondería con una escala wavelet diferente.

En otra realización, la invención opera completamente en el dominio temporal, aprovechándose del teorema de Parseval. La energía de cada sub-banda se calcula filtrando las tramas de audio enventanadas **204** con un banco de filtros en el que cada filtro es un filtro pasobanda de cada sub-banda. El resto de operaciones de **206** se realizan de acuerdo con las descripciones proporcionadas anteriormente. Este modo de operación puede ser particularmente útil para sistemas con recursos computacionales limitados.

Cualquiera de las realizaciones de **206** descritas anteriormente puede aplicar operaciones lineales adicionales a la matriz de coeficientes transformados **208**, dado que en general esto no tendrá ningún impacto negativo en la normalización. Un ejemplo de operación lineal útil es un filtrado lineal pasoalto de los coeficientes transformados con el fin de eliminar variaciones de baja frecuencia a lo largo del eje t de la matriz de coeficientes transformados, que no contienen información.

Con respecto a la cuantificación **220**, la elección del cuantificador más apropiado se puede realizar de acuerdo con diferentes requisitos. La invención se puede configurar para trabajar con cuantificadores vectoriales, pero las disposiciones descritas aquí consideran únicamente cuantificadores escalares. Una de las principales razones de esta elección es computacional, como se explicó anteriormente. Para un entero positivo $Q > 1$ se define un cuantificador escalar de Q niveles mediante un conjunto de Q-1 umbrales que dividen el eje real en Q intervalos disjuntos (también conocidos como celdas), y mediante un símbolo (también llamado nivel de reconstrucción o centroide) asociado a cada intervalo de cuantificación. El cuantificador asigna a cada coeficiente postprocesado un índice q en el alfabeto {0, 1, …, Q−1}, dependiendo del intervalo en el que esté contenido. La conversión del índice q en el correspondiente símbolo S_q es necesaria únicamente para la comparación de hashes robustos que se describe a continuación. A pesar de que el cuantificador pueda ser escogido arbitrariamente, la presente invención considera un método de entrenamiento para construir un cuantificador optimizado que consta de los siguientes pasos, ilustrados en la Fig. 6.

En primer lugar se compila un conjunto de entrenamiento **602** consistente en un gran número de fragmentos de audio. Estos fragmentos de audio no necesitan contener muestras distorsionadas, pero pueden ser tomadas como fragmentos de audio de referencia (es decir, originales).

El segundo paso **604** aplica los procedimientos ilustrados en la Fig. 2 (enventanado **202**, transformación **206**, normalización **212**, postprocesado **216**), de acuerdo con la descripción anterior, a cada fragmento de audio del conjunto de entrenamiento. Por tanto, para cada fragmento de audio se obtiene una matriz de coeficientes postprocesados. Las matrices calculadas para cada fragmento de audio se concatenan a lo largo de la dimensión t para crear una única matriz de coeficientes postprocesados **606** conteniendo información de todos los fragmentos. Cada fila $\mathbf{r}_{f'}$, con $1 \leq f' \leq F'$, tiene longitud L_c.

Para cada fila $\mathbf{r}_f$ de la matriz de coeficientes postprocesados **606** se calcula la partición P_f del eje real en Q intervalos disjuntos **608** de tal modo que la partición maximice una función de coste predefinida. Una función de coste adecuada es la entropía empírica de los coeficientes cuantificados, que se calcula mediante la siguiente fórmula:

$$\mathrm{Ent}(\mathcal{P}_f) = -\sum_{i=1}^{Q} (N_{i,f}/L_c) \log(N_{i,f}/L_c), \quad (16)$$

donde $N_{i,f}$ es el número de coeficientes de la f-ésima fila de la matriz de coeficientes postprocesados **606** asignada al i-ésimo intervalo de la partición P_f. Cuando (16) es máxima (es decir, se aproxima a log(Q)), la salida del cuantificador contiene toda la información posible, maximizando de esta manera la discriminabilidad del hash robusto. Por tanto se construye una partición optimizada para cada fila de la matriz concatenada de coeficientes postprocesados **606**. Esta partición consiste en la secuencia de Q−1 umbrales **610** dispuestos en orden ascendente. Obviamente, el parámetro Q puede ser diferente para el cuantificador de cada fila.

Finalmente, para cada partición obtenida en el paso previo **608**, se calcula un símbolo asociado con dicho intervalo **612**. Se pueden considerar varios métodos para calcular dichos símbolos **614**. La presente invención considera, entre otros, el centroide que minimiza la distorsión media para cada intervalo de cuantificación, que puede calcularse de forma sencilla obteniendo la media condicional de cada intervalo de cuantificación de acuerdo con el conjunto de entrenamiento. Otro método para calcular los símbolos, que obviamente también está dentro del ámbito de la presente invención, consiste en asignar a cada partición un valor fijo de acuerdo con una modulación Q-PAM (Pulse Amplitude Modulation de Q niveles). Por ejemplo, para Q=4 los símbolos serían $\{-c_2, -c_1, c_1, c_2\}$ con c_1 y c_2 dos números reales positivos.

Con el método descrito arriba se consigue un cuantificador optimizado para cada fila de la matriz de coeficientes postprocesados **218**. El conjunto resultante de cuantificadores puede ser no uniforme y no simétrico, dependiendo de las propiedades de los coeficientes a cuantificar. El método descrito arriba da soporte, no obstante, a más cuantificadores estándar simplemente escogiendo funciones de coste apropiadas. Por ejemplo, las particiones pueden restringirse a ser simétricas con el fin de facilitar las implementaciones hardware. También, en aras de la simplicidad, las filas de la matriz de coeficientes postprocesados **606** pueden ser concatenadas con el fin de obtener un único cuantificador que será aplicado a todos los coeficientes postprocesados.

En ausencia de normalización **212**, el uso de un cuantificador multinivel puede degradar el rendimiento de forma drástica dado que las fronteras entre los intervalos de cuantificación no se adaptarían a las distorsiones introducidas por el canal de captura de micrófono. Gracias a las propiedades debidas a la normalización **212** se garantiza que el procedimiento de cuantificación es eficaz incluso en este caso. Otra ventaja de la presente invención es que al hacer el cuantificador dependiente del conjunto de entrenamiento y no del contenido de audio concreto cuyo hash se quiere calcular, la robustez ante distorsiones severas aumenta considerablemente.

Tras realizar la cuantificación **220**, los elementos de la matriz de coeficientes postprocesados cuantificados se disponen por columnas en un vector. Los elementos del vector resultante, que son los índices de los correspondientes intervalos de cuantificación, se convierten finalmente en una representación binaria en aras de obtener una representación compacta. El vector resultante constituye el hash final **110** del contenido de audio **102**.

El objetivo de comparar dos hashes robustos es decidir si representan el mismo contenido de audio. El método de comparación se ilustra en la Fig. **¡Error! No se encuentra el origen de la referencia.**. La base de datos **112** contiene hashes de referencia almacenados como vectores, que fueron precalculados para los correspondientes contenidos de audio de referencia. El método para calcular estos hashes de referencia es el mismo que se ha descrito anteriormente y que se muestra en la Fig. 2. En general, los hashes de referencia pueden ser más largos que el hash extraído del contenido de audio a analizar, que normalmente es un fragmento de audio pequeño. En lo que sigue supondremos que la longitud temporal del hash **110** extraído del audio a analizar es J, que es más pequeño que la de los hashes de referencia. Una vez se escoge un hash de referencia **302** en **112,** el método de comparación comienza con la extracción **304** de un sub-hash más corto **306** de longitud J a partir de aquél. El primer elemento del primer sub-hash se indexa mediante un puntero **322**, que se inicializa al valor 1. A continuación, los elementos del hash de referencia **302** en las posiciones de la 1 a J se leen en orden para componer el primer sub-hash de referencia **306**.

A diferencia de la mayoría de métodos de comparación enumerados en el arte existente, que usan la distancia Hamming para comparar hashes, aquí se usa la correlación normalizada como una medida eficaz de similitud. Se ha comprobado experimentalmente en nuestra aplicación que la correlación normalizada mejora significativamente el comportamiento de las distancias de norma p o el de la distancia Hamming. La correlación normalizada mide la similitud entre dos hash como el coseno de su ángulo en un espacio J-dimensional. Antes de calcular la correlación normalizada es necesario convertir **308** los elementos binarios del sub-hash **306** y el hash de la solicitud **110** en símbolos con valor real (es decir, los valores de reconstrucción) dados por el cuantificador. Una vez se ha realizado esta conversión se puede calcular la correlación normalizada. En lo que sigue denotaremos el hash de la solicitud **110** por $\mathbf{h}_q$, y los sub-hashes de referencia **306** por $\mathbf{h}_r$. La correlación normalizada **310** calcula la medida de similitud **312**, que siempre se encuentra en el rango [−1,1], de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)}, \qquad (17)$$

donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}}. \qquad (18)$$

Cuanto más próximo a 1 sea el valor de esta expresión, mayor similitud hay entre los dos hashes. Recíprocamente, cuanto más próximo sea a -1, más diferentes son.

El resultado de la correlación normalizada **312** se almacena temporalmente en un buffer **316**. A continuación se comprueba **314** si el hash de referencia **302** contiene más sub-hashes que comparar. Si éste fuera el caso se extrae un nuevo sub-hash **306** incrementando el puntero **322** y obteniendo un nuevo vector de J elementos de **302**. El valor del puntero **322** se incrementa en una cantidad tal que el primer elemento del siguiente sub-hash se corresponda con el comienzo de la siguiente trama de audio. Por tanto, dicha cantidad depende tanto de la duración de la trama como del solapamiento entre tramas. Para cada nuevo sub-hash se calcula un nuevo valor de correlación normalizada **312** y se almacena en el buffer **316**. Una vez no haya mas sub-hashes que extraer del hash de referencia **302**, se calcula **318** una función de los valores almacenados en el buffer **316** y se compara **320** con un umbral. Si el resultado de dicha función es mayor que este umbral, se decide que los hashes comparados representan el mismo contenido de audio. En caso contrario, se considera que los hashes comparados pertenecen a diferentes contenidos de audio. Existen numerosas alternativas para la función que se calcula en los valores de la correlación normalizada. Una de ellas es el máximo –como se muestra en la Fig. 3–, pero otras alternativas (el valor medio, por ejemplo) también serían adecuadas. El valor apropiado para el umbral normalmente se asigna en base a observaciones empíricas, y se comentará a continuación.

El método descrito anteriormente para realizar la comparación se basa en una búsqueda exhaustiva. Una persona experimentada en la técnica puede percatarse de que dicho método basado en el cálculo de la correlación normalizada puede ser realizado con métodos más eficientes para buscar en bases de datos grandes, como se describe en el arte existente, si fuese necesario cumplir restricciones de eficiencia específicas.

Preferiblemente se configura la invención de acuerdo con los siguientes parámetros, que han demostrado un muy buen comportamiento en sistemas prácticos. En primer lugar se remuestrea el audio de la solicitud **102** a 11025 Hz. La duración de un fragmento de audio para realizar la solicitud se establece a 2 segundos. El solapamiento entre tramas se configura a un 90%, con el fin de tratar con fallos de sincronismo, y cada trama $\{fr_t\}$, con $1 \leq t \leq T$ se enventana con una ventana Hanning. La longitud N de cada trama fr_t se establece en 4096 muestras, obteniéndose 0.3641 segundos. En el procedimiento de transformación **206** se transforma cada trama por medio de una transformada rápida de Fourier (FFT) de tamaño 4096. Los coeficientes FFT se agrupan en 30 sub-bandas críticas en el rango $[f_1, f_c]$ (Hz). Los valores para las frecuencias de corte son f_l=300 Hz, f_c=2000 Hz, por dos motivos:

1. La mayor parte de la energía de las señales de audio naturales se concentran en frecuencias bajas, típicamente por debajo de los 4 KHz, y las distorsiones no lineales introducidas por los sistemas de reproducción de sonido y adquisición son mayores para frecuencias altas.

2. Las frecuencias muy bajas son imperceptibles para los humanos, y normalmente contienen información espúrea. En el caso de la captura de audio con micrófonos integrados en ordenadores portátiles, las componentes en frecuencia por debajo de los 300 Hz contienen típicamente una gran cantidad de ruido del ventilador.

Los límites de cada banda crítica se calculan de acuerdo con la conocida escala Mel, que imita las propiedades del sistema auditivo humano. Para cada una de las 30 sub-bandas críticas se calcula la energía de los coeficientes DFT. Por tanto se construye una matriz de coeficientes transformados de tamaño 30×44, donde 44 es el número de tramas T incluidas en el contenido de audio **102**. A continuación se aplica un filtro lineal pasobanda a cada fila de la matriz tiempo-frecuencia con el fin de filtrar efectos espúreos tales como valores medios no nulos o variaciones de alta frecuencia. A continuación a la matriz filtrada de coeficientes transformados se le aplica un procesado de reducción de dimensionalidad utilizando una aproximación PCA modificada que consiste en la maximización de los momentos de cuarto orden en un conjunto de entrenamiento de contenidos de audio. La matriz de coeficientes transformados resultante **208** del último fragmento de 2 segundos es de tamaño F×44, con F ≤ 30. La reducción de dimensionalidad permite reducir F hasta 12, pero manteniendo un alto rendimiento en la identificación de audio.

Para la normalización **212** se usa la función (6), junto con la función G() dada por (7), obteniéndose una matriz de coeficientes normalizados de tamaño F×43, con F ≤ 30. Como se ha explicado anteriormente el parámetro p puede adoptar cualquier valor real positivo. Se ha comprobado experimentalmente que la elección óptima de p, en el sentido de minimizar las probabilidades de error, se encuentra en el rango [1,2]. En particular, la realización preferente usa la función con p=1,5. El vector de pesos se fija a **a** = [1, 1, …, 1]. Falta asignar el valor del parámetro L, que es la longitud de la ventana de normalización. Como se ha explicado anteriormente, existe un compromiso entre robustez al ruido y tiempo de adaptación a las variaciones del canal. Si el canal de captura con micrófono cambia muy rápido, una posible solución para mantener un L grande es la de incrementar la tasa de muestreo del audio. Por tanto, el valor óptimo de L depende de la aplicación. En la realización preferente a L se le asigna el valor 20. Por tanto, la duración de la ventana de normalización es de 1,1 segundos, que para aplicaciones típicas de identificación de audio es suficientemente pequeño.

Preferiblemente el postprocesado **216** implementa la función identidad, lo que en la práctica equivale a no realizar ningún postprocesado. El cuantificador **220** usa 4 niveles de cuantificación, en los que la partición y los símbolos se obtienen de acuerdo con los métodos descritos anteriormente (maximización de entropía y centroides de media condicional) aplicados sobre un conjunto de señales de audio de entrenamiento.

Las Figs. 7 y 8 ilustran el rendimiento de un ejemplo preferido en un escenario real, en el que el audio a identificar se consigue capturando un fragmento de audio de dos segundos utilizando el micrófono integrado de un ordenador portátil a 2,5 metros de la fuente de audio en un salón. Como queda reflejado en las Figs. 7 y 8, el rendimiento ha sido comprobado en dos casos diferentes: identificación de fragmentos de música, e identificación de fragmentos de conversaciones. Incluso a pesar de que las gráficas muestren una degradación importante del rendimiento para el caso de música en comparación con el de conversaciones, el valor de P_{MD} sigue siendo menor que 0,2 para P_{FP} por debajo de 10^{-3}, y menor que 0,06 para P_{FP} por debajo de 10^{-2}.

La Fig. 9 muestra el diagrama de bloques general de un ejemplo que utiliza la presente invención para realizar identificación de audio en modo flujo, en tiempo real. Se podría usar el presente ejemplo, por ejemplo, para realizar la identificación continua de una difusión de audio. Este ejemplo de realización de la invención usa una arquitectura cliente-servidor que se explica a continuación. Se mantienen todos los parámetros asignados en el ejemplo preferido descritos anteriormente.

1. El cliente **901** recibe un flujo de audio a través de algún dispositivo de captura **902**, que puede ser por ejemplo un micrófono acoplado a un convertidor A/D. Las muestras de audio recibidas se guardan consecutivamente en un buffer **904** de longitud predeterminada que equivale a la longitud del audio de la solicitud. Cuando el buffer está lleno se leen y procesan las muestras de audio **108** de acuerdo con el método ilustrado en la Fig. **¡Error! No se encuentra el origen de la referencia.** con el fin de calcular el correspondiente hash robusto.

2. El hash robusto, junto con un umbral predefinido por el cliente, es enviado **906** al servidor **911**. El cliente **901** espera entonces por la respuesta del servidor **911**. Cuando se recibe dicha respuesta se muestra **908** al cliente.

3. El servidor se configura para recibir múltiples flujos de audio **910** de múltiples fuentes de audio (en lo sucesivo, "canales"). De forma similar al cliente, las muestras recibidas de cada canal se guardan consecutivamente en un buffer **912**. No obstante, la longitud del buffer en este caso no es la misma que la longitud del audio de la solicitud. Antes bien, el buffer **912** tiene una longitud igual al número de muestras N de una trama de audio. Además, dicho buffer es un buffer circular que se actualiza cada n_o muestras, donde n_o es el número de muestras no solapantes.

4. Cada vez que se reciben n_o muestras nuevas de un canal dado, el servidor calcula **108** el hash robusto de las muestras del canal almacenadas en el correspondiente buffer, que forman una trama completa. Cada nuevo hash se almacena consecutivamente en un buffer **914**, que también se implementa como un buffer circular. Este buffer tiene una longitud predefinida, significativamente más grande que la del hash correspondiente a la solicitud, con el fin de dar cabida a posibles retardos del lado del cliente y a retardos causados por la transmisión a través de las redes de datos.

5. Cuando se recibe un hash del cliente se realiza una comparación **114** (ilustrada en la Fig. **¡Error! No se encuentra el origen de la referencia.**) entre el hash recibido (el hash de la solicitud **110**) y cada uno de los hashes almacenados en los buffer de canal **914**. En primer lugar, se asigna a un puntero **916** el valor 1 con el fin de escoger **918** el primer canal. El resultado **920** de la comparación (correspondencia/ no correspondencia) se almacena en un buffer **922**. Si no quedan más canales por comparar el puntero **916** se incrementa de forma acorde y se realiza una nueva comparación. Una vez se ha comparado el hash recibido con todos los canales se envía **926** el resultado **920** -identificando el canal que coincide, si existe tal coincidencia- al cliente, que finalmente muestra **908** el resultado.

El cliente sigue enviando nuevas solicitudes a intervalos regulares (de duración igual a la del buffer **904** del cliente) y recibiendo las correspondientes respuestas del servidor. De esta forma se actualiza de forma regular la identidad del audio capturado por el cliente.

Como se ha resumido anteriormente, el cliente **901** sólo es responsable de extraer el hash robusto del audio capturado, mientras que el servidor **911** es responsable de extraer los hashes de todos los canales de referencia y de realizar las comparaciones cuando se recibe una solicitud del cliente. Esta distribución de responsabilidades tiene varias ventajas: primero, el coste computacional del cliente es muy bajo, y segundo, la información transferida entre cliente y servidor permite una tasa de transmisión muy baja.

Cuando se usa en modo flujo como se ha descrito aquí, la presente invención puede aprovecharse de la operación de normalización **212** realizada durante la extracción del hash **108**. Más concretamente, el buffer **210** se puede usar para almacenar un número suficiente de coeficientes pasados con el fin de tener siempre L coeficientes para realizar la normalización. Como se ha mostrado anteriormente en las ecuaciones (4) y (5), cuando opera en modo offline (es decir, con una solicitud de audio aislada) la normalización no siempre puede usar L coeficientes pasados porque pueden no estar disponibles. Gracias al uso del buffer **210** se asegura que siempre hay disponibles L coeficientes pasados, mejorando de esta manera el rendimiento global de la identificación. Cuando se usa el buffer **210** el hash calculado para un fragmento de audio dado será dependiente de un cierto número de fragmentos de audio procesados previamente. Esta propiedad hace la invención altamente robusta ante la propagación multitrayecto y los efectos del ruido cuando la longitud L del buffer es suficientemente grande.

El buffer **210** en el instante t contiene un vector (5) por cada fila de la matriz de coeficientes transformados. Para una implementación eficiente el buffer **210** es un buffer circular donde, para cada nueva trama analizada, se añade el elemento X(f, t) más reciente y se descarta el elemento X(f, t−L) más antiguo. Si el valor más reciente de $G(\underline{\mathbf{X}}_{f,t})$ se almacena convenientemente, entonces si $G(\underline{\mathbf{X}}_{f,t})$ viene dada por (7), su valor simplemente se actualiza como sigue:

$$G(\underline{\mathbf{X}}_{f,t+1}) = \left(G^2(\underline{\mathbf{X}}_{f,t}) + \frac{1}{L}\left(|X(f,t)|^2 - |X(f,t-L)|^2 \right) \right)^{\frac{1}{2}}. \quad (19)$$

Por tanto, para cada nueva trama analizada, el cálculo del factor de normalización requiere dos operaciones aritméticas simples independientemente de la longitud L del buffer.

Al operar en modo flujo el cliente **901** recibe los resultados de las comparaciones realizadas por el servidor **911**. En caso de tener más de una correspondencia, el cliente elige aquélla con el valor más alto de

correlación normalizada. Suponiendo que el cliente esté escuchando uno de los canales monitorizados por el servidor, pueden darse tres tipos de eventos:

1. El cliente puede mostrar un identificador que se corresponda con el canal cuyo audio está siendo capturado. Decimos que el cliente está “enganchado” en el canal correcto.

2. El cliente puede mostrar un identificador que se corresponda con un canal incorrecto. Decimos que el cliente está “falsamente enganchado”.

3. El cliente puede no mostrar ningún identificador porque el servidor no ha encontrado ninguna coincidencia. Decimos que el cliente está “desenganchado”. Esto ocurre cuando no hay ninguna coincidencia.

Cuando el cliente está escuchando un canal de audio que no es ninguno de los canales monitorizados por el servidor el cliente siempre debería estar desenganchado. En caso contrario, el cliente estaría falsamente enganchado. Al realizar identificación continua de audio de difusión es deseable estar correctamente enganchado el mayor tiempo posible. No obstante, el evento de estar falsamente enganchado es altamente indeseable, por lo que en la práctica su probabilidad debe mantenerse muy baja. La Fig. 10 muestra la probabilidad de ocurrencia de todos los posibles eventos, obtenida empíricamente, en términos del umbral usado para detectar una correspondencia. El experimento fue llevado a cabo en un entorno real donde el dispositivo de captura fue el micrófono integrado de un ordenador portátil. Como puede apreciarse, la probabilidad de estar falsamente enganchado es despreciable para umbrales por encima de 0.3, manteniendo muy alta a su vez la probabilidad de estar correctamente enganchado (por encima 0.9). Se ha hallado que este comportamiento se mantiene bastante estable en experimentos con otros ordenadores portátiles y micrófonos.

REIVINDICACIONES

1. Un método de hashing robusto de audio, incluyendo un paso de extracción de hash robusto en el que se extrae un hash robusto (110) a partir de contenido de audio (102, 106); dicho paso de extracción de hash robusto incluye:

- dividir el contenido de audio (102, 106) en, al menos, una trama;

- aplicar un proceso de transformación (206) en la trama para calcular, en dicha trama, un conjunto de coeficientes transformados (208);

- aplicar un proceso de normalización (212) a los coeficientes transformados (208) para obtener un conjunto de coeficientes normalizados (214), en el que dicha normalización (212) implica el cálculo del producto del signo de cada coeficiente de dichos coeficientes transformados (208) por el cociente de dos funciones homogéneas de cualquier combinación de dichos coeficientes transformados (208), donde ambas funciones homogéneas son del mismo orden;

- aplicar un procedimiento de cuantificación (220) sobre dichos coeficientes normalizados (214) para obtener el hash robusto (110) del contenido de audio (102,106).

2. El método de la reivindicación 1 mediante el que se realiza un paso de comparación en el que el hash robusto (110) se compara con al menos un hash de referencia (302) para obtener una correspondencia.

3. El método de la reivindicación 2, en el que el paso de comparación implica, para cada hash de referencia (302):

extraer del correspondiente hash de referencia (302) al menos un sub-hash (306) de la misma longitud *J* que la longitud del hash robusto (110);

convertir (308) el hash robusto (110) y cada uno de dichos sub-hashes (306) en los correspondientes símbolos de reconstrucción dados por el cuantificador;

calcular una medida de similitud (312) de acuerdo con la correlación normalizada (310) entre el hash robusto (110) y cada uno de dichos sub-hashes (306) de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash a estudiar (110) de longitud *J*, h_r un sub-hash de referencia (306) de la misma longitud *J*, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}};$$

comparar una función de dichas medidas de similitud (312) con un umbral predefinido;

decidir, en base a dicha comparación, si el hash robusto (110) y el hash de referencia (302) representan el mismo contenido de audio.

4. El método de las anteriores reivindicaciones, en el que el proceso de normalización (212) se aplica sobre los coeficientes transformados (208) dispuestos en una matriz de tamaño *F*×*T* para obtener una matriz de coeficientes normalizados (214) de tamaño *F'* × *T'*, con *F'* = *F*, *T'* ≤ *T*, cuyos elementos *Y*(*f'*, *t'*) se calculan de acuerdo con la siguiente regla:

$$Y(f',t') = \frac{\text{sign}(X(f',M(t'))) \times H(\mathbf{X}_{f'})}{G(\mathbf{X}_{f'})},$$

donde $X(f', M(t'))$ son los elementos de la matriz de coeficientes transformados (208), $\mathbf{X}_{f'}$ es la fila f-ésima de la matriz de coeficientes transformados (208), $M()$ es una función que hace corresponder a índices del conjunto {1,…, T'} otros índices de {1,…, T}, y tanto $H()$ como $G()$ son funciones homogéneas del mismo orden.

5. El método de la reivindicación 4, en el que las funciones homogéneas $H()$ y $G()$ son tales que:

$$H(\mathbf{X}_{f'}) = H(\overline{\mathbf{X}}_{f',M(t')}), \;\; G(\mathbf{X}_{f'}) = G(\underline{\mathbf{X}}_{f',M(t')}),$$

con

$$\overline{\mathbf{X}}_{f',M(t')} = [X(f',M(t')), X(f',M(t')+1), \ldots, X(f',k_u)],$$
$$\underline{\mathbf{X}}_{f',M(t')} = [X(f',k_l), \ldots, X(f',M(t')-2), X(f',M(t')-1)],$$

donde k_l es el máximo de $\{M(t')-L_l,1\}$, k_u es el mínimo de $\{M(t')+L_u-1,T\}$, $M(t')>1$, y $L_l>1$, $L_u>0$.

6. El método de la reivindicación 5, en el que $M(t')=t'+1$ y $H(\overline{\mathbf{X}}_{f',M(t')}) = \text{abs}(X(f',t'+1))$, derivando en la siguiente regla de normalización:

$$Y(f',t') = \frac{X(f',t'+1)}{G(\underline{\mathbf{X}}_{f',t'+1})},$$

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

$$G(\underline{\mathbf{X}}_{f',t'+1}) = L^{-\frac{1}{p}} \times (a(1) \times |X(f',t')|^p + a(2) \times |X(f',t'-1)|^p + \ldots + a(L) \times |X(f',t'-L+1)|^p)^{\frac{1}{p}},$$

donde $L_l=L$, $a=[a(1_1, a(2), \ldots, a(L)]$ es un vector de ponderación, y p es un número real positivo.

8. El método de las anteriores reivindicaciones, en el que el proceso de transformación (206) implica una subdivisión espectral por sub-bandas de cada trama (204).

9. El método de las anteriores reivindicaciones, en el que durante el proceso de cuantificación (220) se emplea al menos un cuantificador multinivel.

10. El método de la reivindicación 9, en el que se obtiene al menos un cuantificador multinivel mediante un método de entrenamiento que comprende:

- el cálculo de una partición (608), obteniéndose Q intervalos de cuantificación disjuntos mediante la maximización de una función de coste predefinida que depende de los estadísticos de los coeficientes normalizados calculados a partir de un conjunto de entrenamiento (602) de fragmentos de audio; y

- el cálculo de símbolos (612), asociando un símbolo a cada intervalo calculado.

11. El método de la reivindicación 10, en el que la función de coste es la entropía empírica de los coeficientes cuantificados, calculada de acuerdo con la siguiente fórmula:

$$\text{Ent}(\mathcal{P}_f) = -\sum_{i=1}^{Q} (N_{i,f}/L_c) \log(N_{i,f}/L_c),$$

donde $N_{i,f}$ es el número de coeficientes de la fila f-ésima de la matriz de coeficientes postprocesados asignados al intervalo i-ésimo de la partición, y L_c es la longitud de cada fila.

12. Un método para decidir si dos hashes robustos calculados de acuerdo con el método de hashing de audio robusto de cualquiera de las anteriores reivindicaciones representan el mismo contenido de audio, caracterizado por que dicho método comprende:

- la extracción del hash más largo (302) de al menos un sub-hash (306) de la misma longitud *J* que la longitud del hash más corto (110);

- la conversión (308) del hash más corto (110) y de dichos sub-hashes (306) en los correspondientes símbolos de reconstrucción dados por el cuantificador;

- el cálculo de una medida de similitud (312) de acuerdo con la correlación normalizada (310) entre el hash más corto (110) y cada uno de dichos sub-hashes (306) de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash que está siendo analizado (110) de longitud *J*, h_r un sub-hash de referencia (306) de la misma longitud *J*, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}};$$

la comparación de una función de dichas medidas de similitud (312) con un umbral predefinido;

- la decisión, en base a dicha comparación, de si los dos hashes robustos (110, 302) representan el mismo contenido de audio.

13. Un sistema de hashing robusto de audio, caracterizado por que comprende un módulo de extracción de hash robusto (108) para extraer un hash robusto (110) a partir de contenido de audio (102,106), y el módulo de extracción de hash robusto (108) comprendiendo medios de procesamiento de datos configurados para:

- la división del contenido de audio (102, 106) en al menos una trama;

- la aplicación de un proceso de transformación (206) sobre dichas tramas para calcular, para cada una de ellas, un conjunto de coeficientes transformados (208);

- la aplicación de un proceso de normalización (212) sobre los coeficientes transformados (208) para obtener un conjunto de coeficientes normalizados (214), donde dicho proceso de normalización (212) comprende el cálculo del producto del signo de cada coeficiente transformado (208) por el cociente de dos funciones homogéneas de cualquier combinación de los coeficientes transformados (208), donde ambas funciones homogéneas son del mismo orden;

- la aplicación de un proceso de cuantificación (220) en dichos coeficientes normalizados (214) para obtener un hash robusto (110) del contenido de audio (102, 106).

14. El sistema de la reivindicación 13 que comprende a mayores un módulo de comparación (114) para comparar el hash robusto (110) con al menos un hash de referencia (302) para encontrar una correspondencia.

15. Un sistema para decidir si dos hashes robustos calculados mediante el sistema de hashing robusto de audio de las reivindicaciones 13 y 14 representan el mismo contenido de audio, que caracterizado por que dicho sistema comprende medios de procesamiento de datos configurados para:

- la extracción del hash más largo (302) de al menos un sub-hash (306) con la misma longitud *J* que la longitud del hash más corto (110);

- la conversión (308) del hash más corto (110) y cada uno de dichos sub-hashes (306) en los correspondientes símbolos de reconstrucción dados por el cuantificador;

- el cálculo de una medida de similitud (312) de acuerdo con la correlación normalizada (310) entre el hash más corto (110) y cada uno de dichos sub-hashes (306) de acuerdo con la siguiente regla:

$$C = \frac{\sum_{i=1}^{J} h_q(i) \times h_r(i)}{\mathrm{norm}_2(\mathbf{h}_q) \times \mathrm{norm}_2(\mathbf{h}_r)},$$

donde h_q representa el hash a comparar (110) de longitud J, h_r un sub-hash de referencia (306) de la misma longitud J, y donde

$$\mathrm{norm}_2(\mathbf{h}) = \left(\sum_{i=1}^{J} \mathbf{h}(i)^2 \right)^{\frac{1}{2}};$$

- la comparación de una función de dicha media de similitud (312) con un umbral predefinido;

- la decisión, en base a dicha comparación, de si dos hashes robustos (110, 302) representan el mismo contenido de audio.

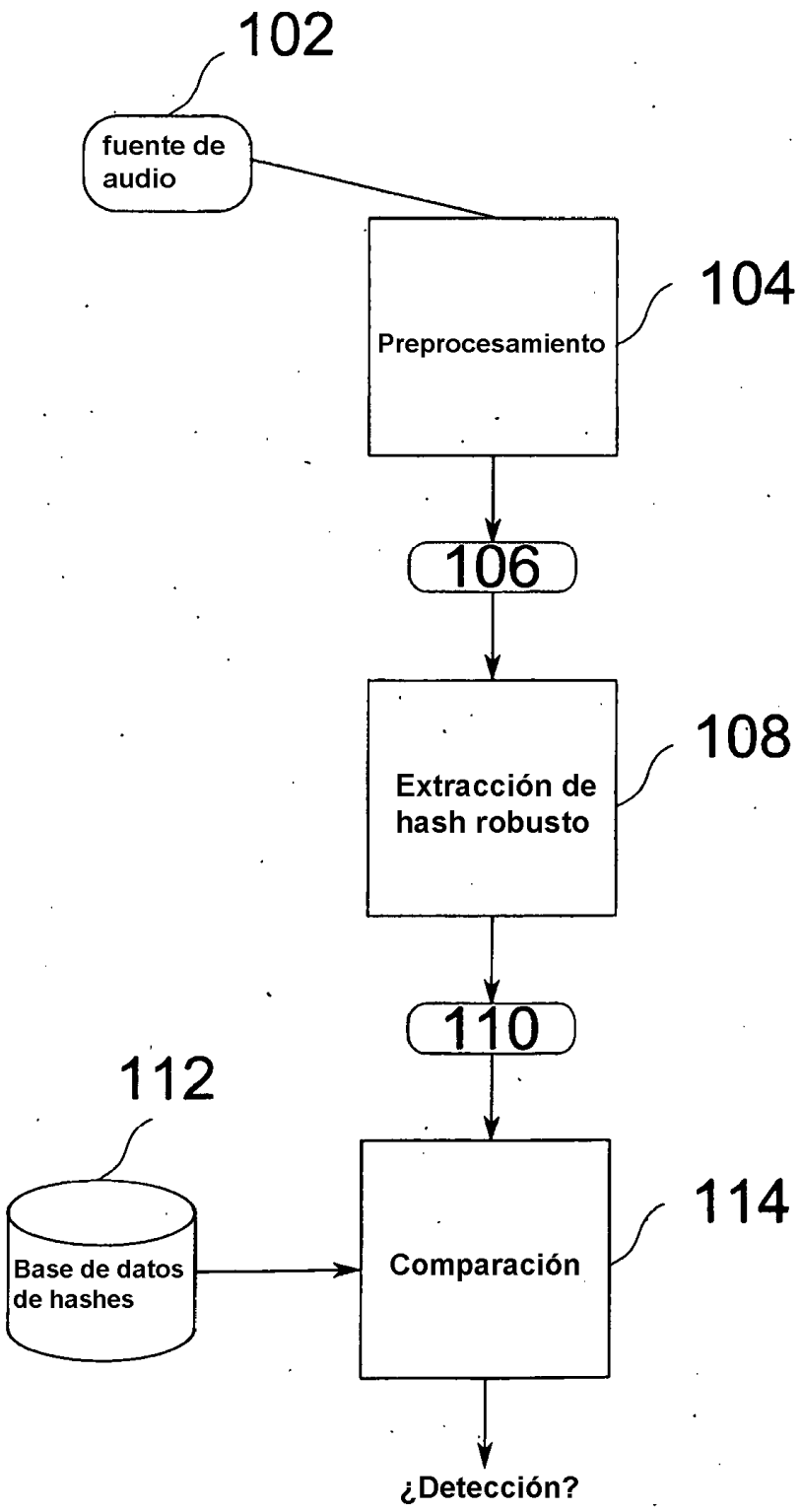
102
fuente de audio
Preprocesamiento
104
106
Extracción de hash robusto
108
110
112
Base de datos de hashes
Comparación
114
¿Detección?

Fig. 1

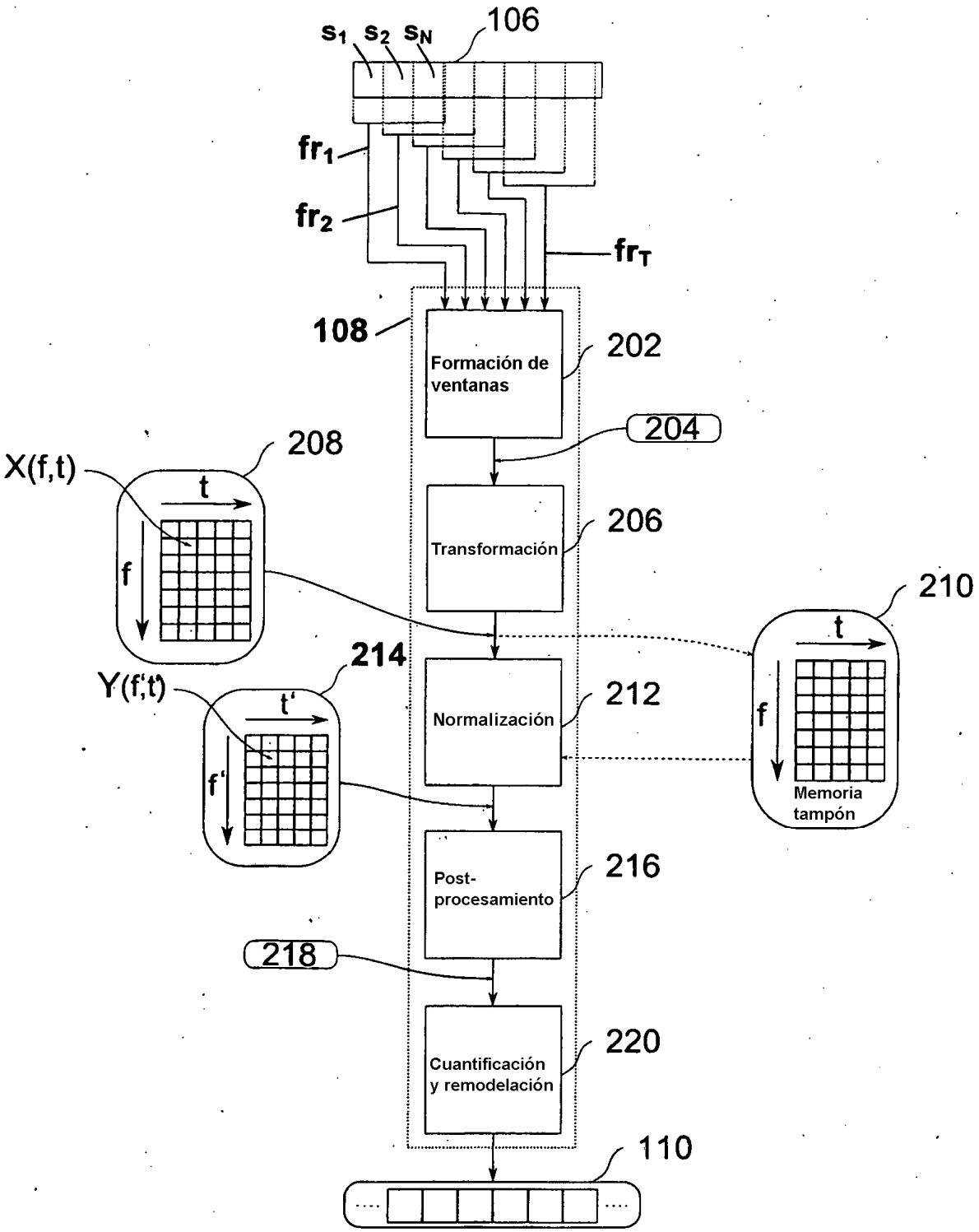
s1 s2 sN
106
fr1
fr2
frT
108
Formación de ventanas
202
204
208
X(f,t)
t
f
Transformación
206
210
214
Y(f',t')
t'
f'
Normalización
212
Memoria tampón
Post-procesamiento
216
218
Cuantificación y remodelación
220
110

Fig. 2

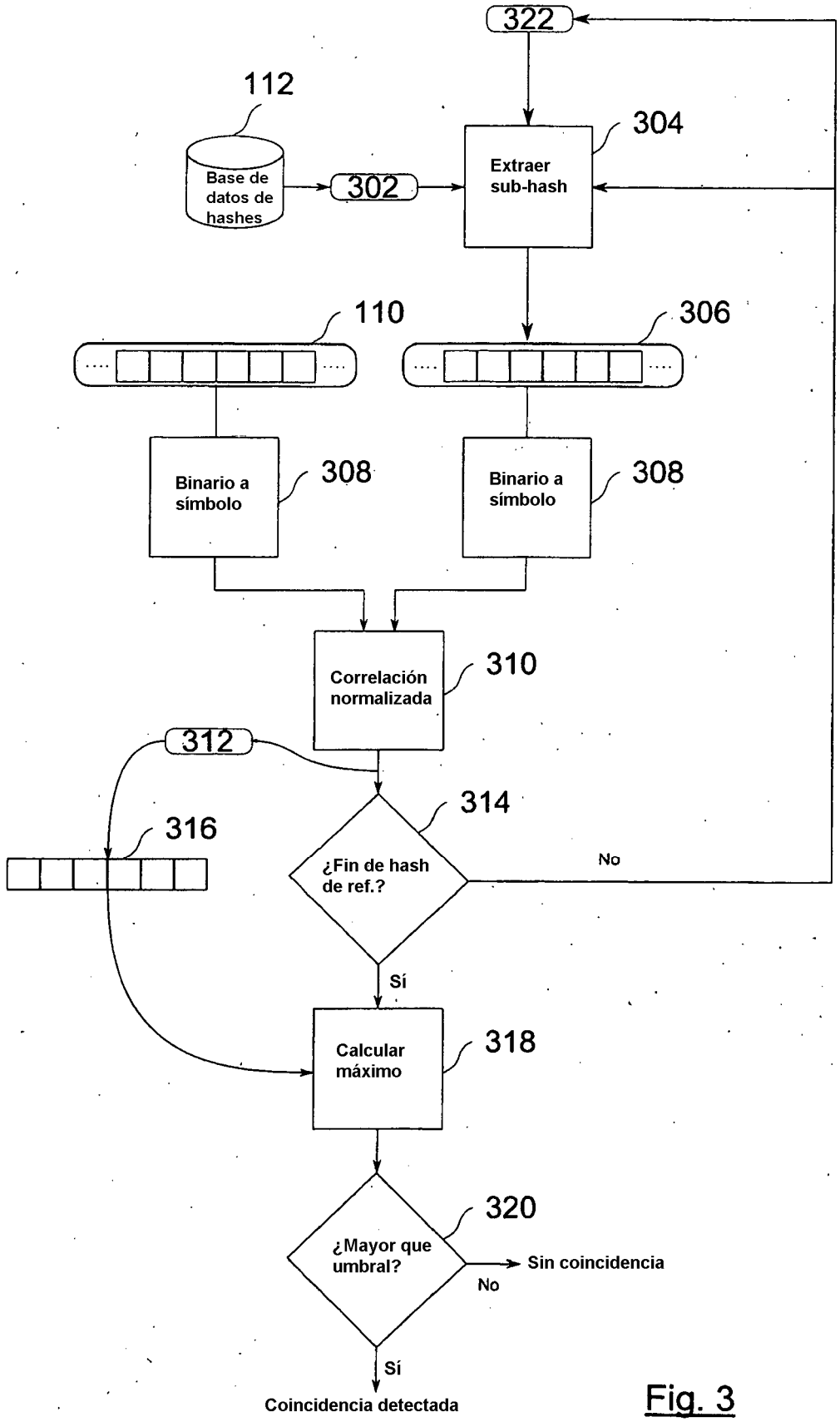
322
112
Base de datos de hashes
302
304
Extraer sub-hash
110
306
Binario a símbolo
308
Binario a símbolo
308
Correlación normalizada
310
312
316
314
¿Fin de hash de ref.?
No
Sí
Calcular máximo
318
¿Mayor que umbral?
320
Sin coincidencia
No
Sí
Coincidencia detectada

Fig. 3

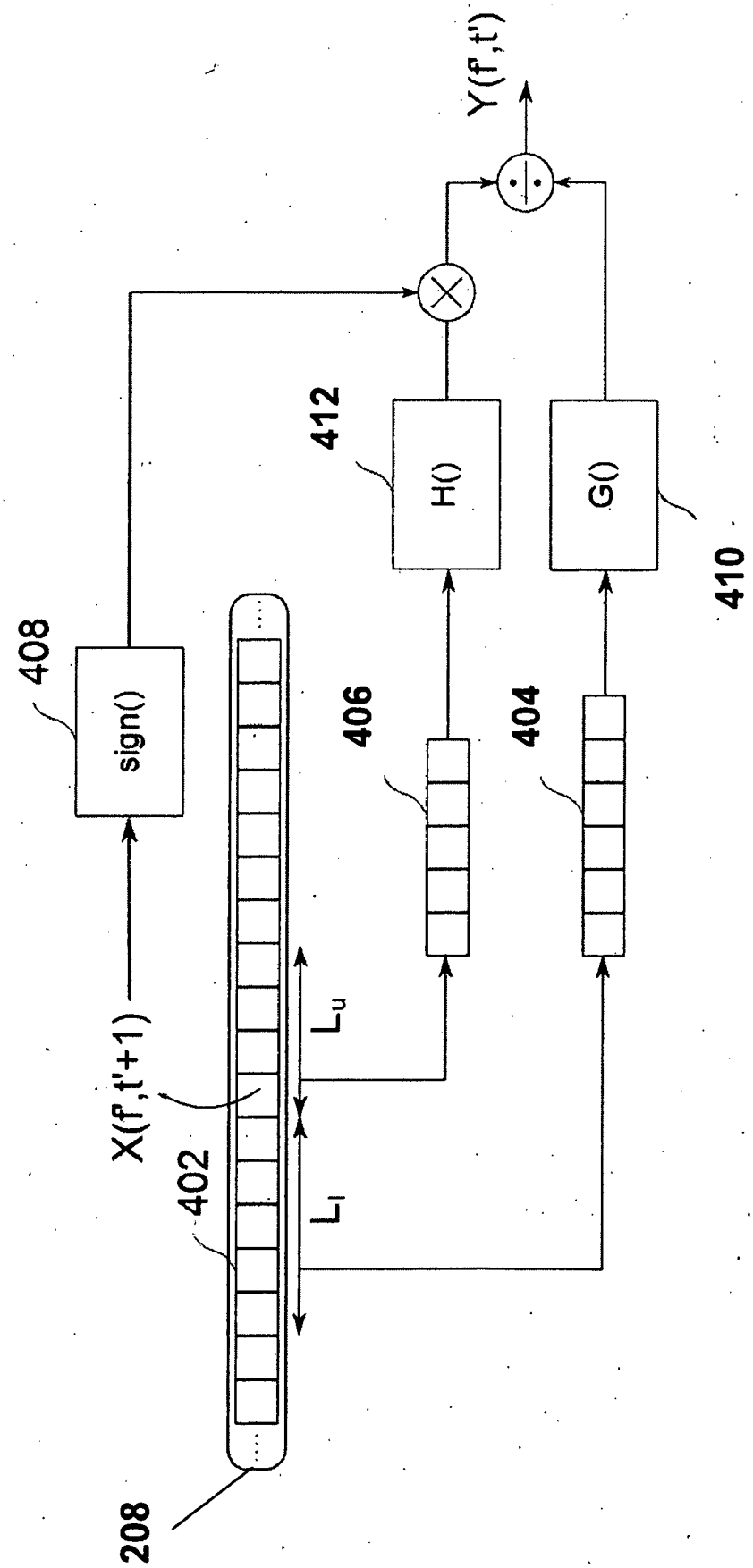

Fig. 4

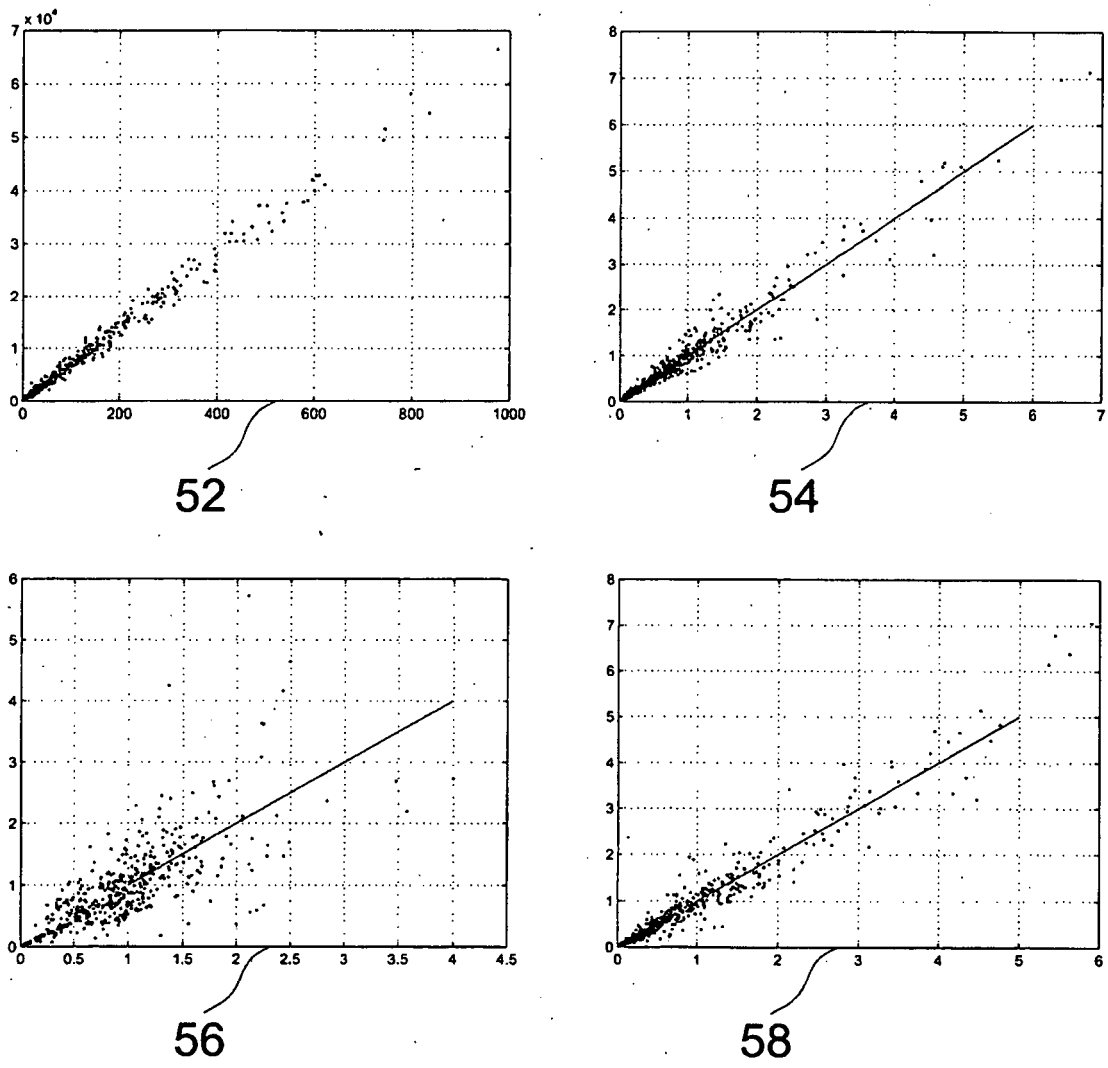

Fig. 5

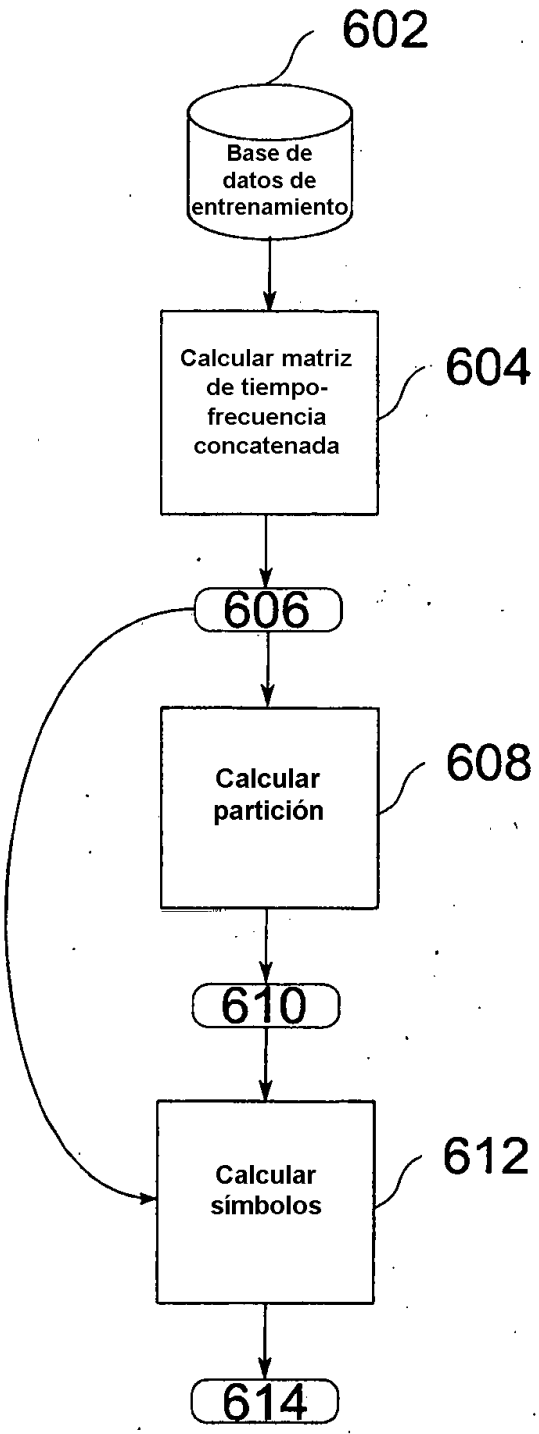
602
Base de datos de entrenamiento
Calcular matriz de tiempo-frecuencia concatenada
604
606
Calcular partición
608
610
Calcular símbolos
612
614

Fig. 6

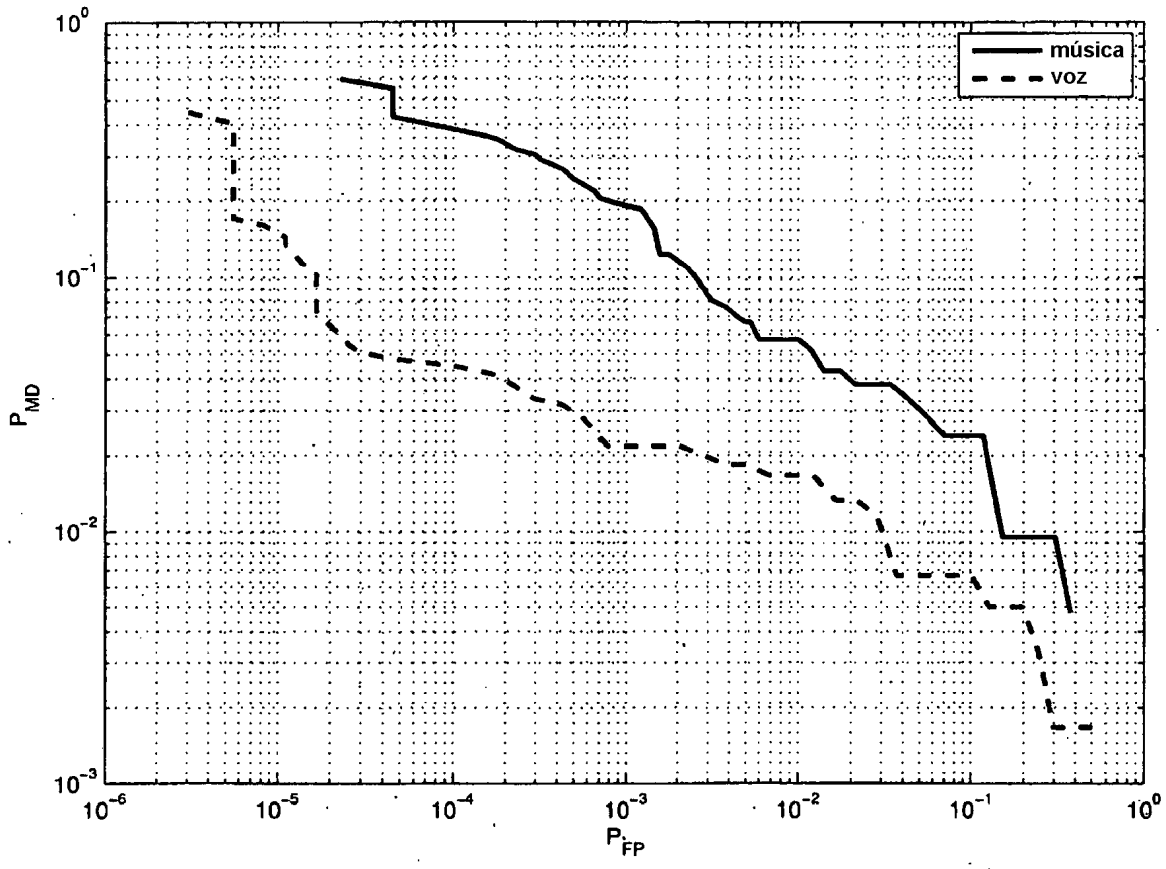

Fig. 7

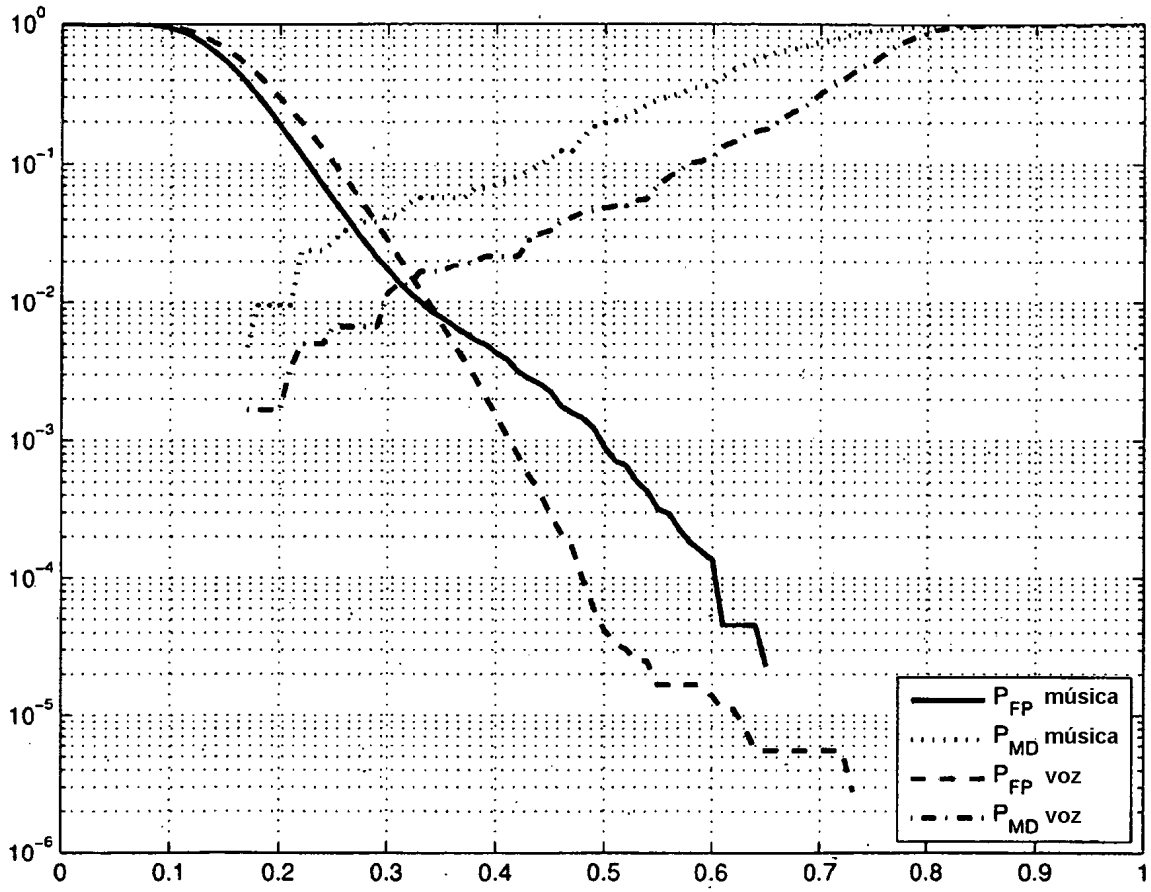
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
P_{FP} música
P_{MD} música
P_{FP} voz
P_{MD} voz

Fig. 8

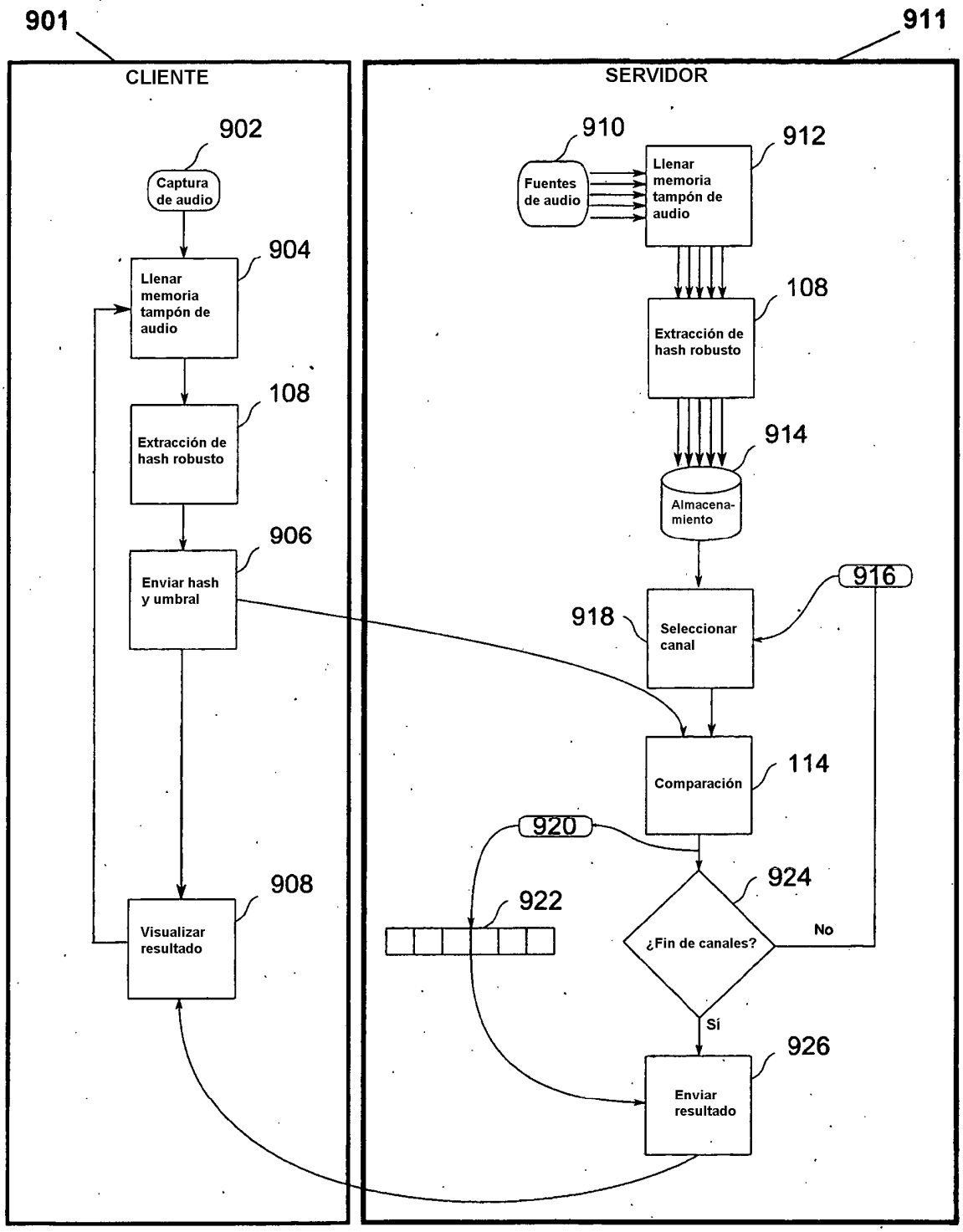
901
CLIENTE
902
Captura de audio
904
Llenar memoria tampón de audio
108
Extracción de hash robusto
906
Enviar hash y umbral
908
Visualizar resultado
911
SERVIDOR
910
Fuentes de audio
912
Llenar memoria tampón de audio
108
Extracción de hash robusto
914
Almacena-miento
916
918
Seleccionar canal
Comparación
114
920
922
924
¿Fin de canales?
No
Sí
926
Enviar resultado

Fig. 9

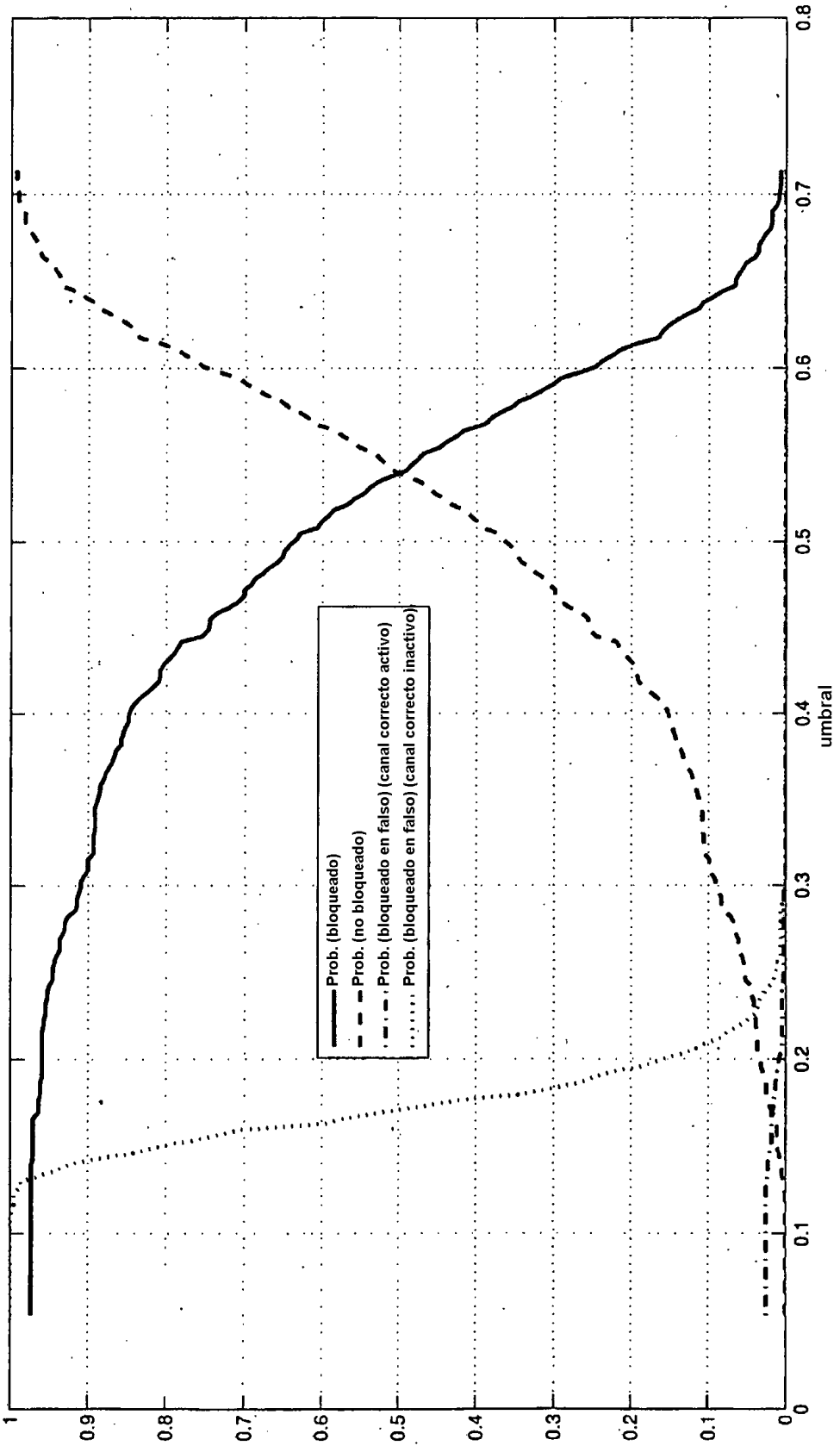
Prob. (bloqueado)
Prob. (no bloqueado)
Prob. (bloqueado en falso) (canal correcto activo)
Prob. (bloqueado en falso) (canal correcto inactivo)
umbral
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fig. 10