# 2. Uso de Audio Fingerprinting para Identificación

De todas las posibles aplicaciones ya nombradas, nos vamos a centrar en el uso de huellas para identificar trozos de audio y, más concretamente, anuncios de radio. Para ello, primero se va a hacer un repaso de los diferentes algoritmos propuestos para realizar esta tarea [9], y posteriormente nos centraremos en el algoritmo de Haitsma y Kalker para Philips [10].

## 2.1 Repaso de los diversos métodos propuestos

A pesar de las distintas lógicas detrás de la tarea de identificación, los métodos comparten ciertos aspectos. Como se puede ver en la figura 3, hay dos procesos fundamentales: la extracción de la huella y el algoritmo de búsqueda de coincidencias.

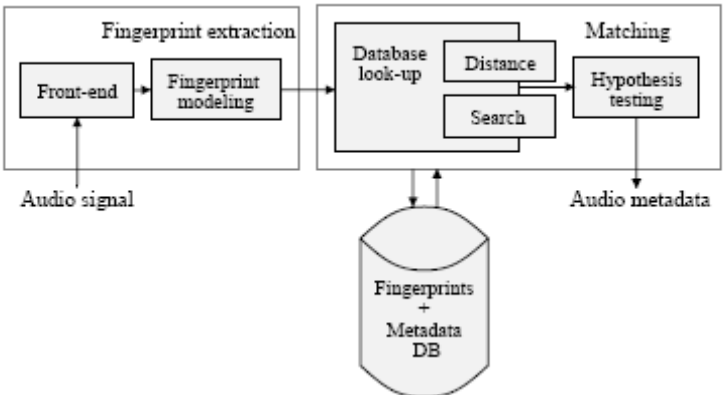


Fig. 3: Estructura general de un sistema de identificación

### 2.1.1 Extracción de Huellas

La extracción de la huella proporciona un conjunto de características perceptuales relevantes de una grabación de una forma concisa y robusta. Los requerimientos de la huella, como ya hemos nombrado otras veces incluyen: capacidad de discriminación sobre un enorme número de otras huellas, invarianza a las distorsiones, compacidad y simplicidad computacional.

Las soluciones propuestas para cumplir todos los requisitos arriba mencionados implican un compromiso entre reducción de la dimensionalidad y pérdida de información. La extracción de la huella consiste en un "front-end" y un bloque de modelado de huellas (ver figura 4). El "front-end" computa una serie de medidas tomadas de la señal, que explicaremos más adelante. El bloque de modelado de huellas define la representación final de la huella, por ejemplo, un vector, una traza de vectores, una secuencia de índices a clases de HMM (modelos ocultos de

Markov), una secuencia de palabras correctoras de errores o atributos musicalmente significativos de alto nivel.

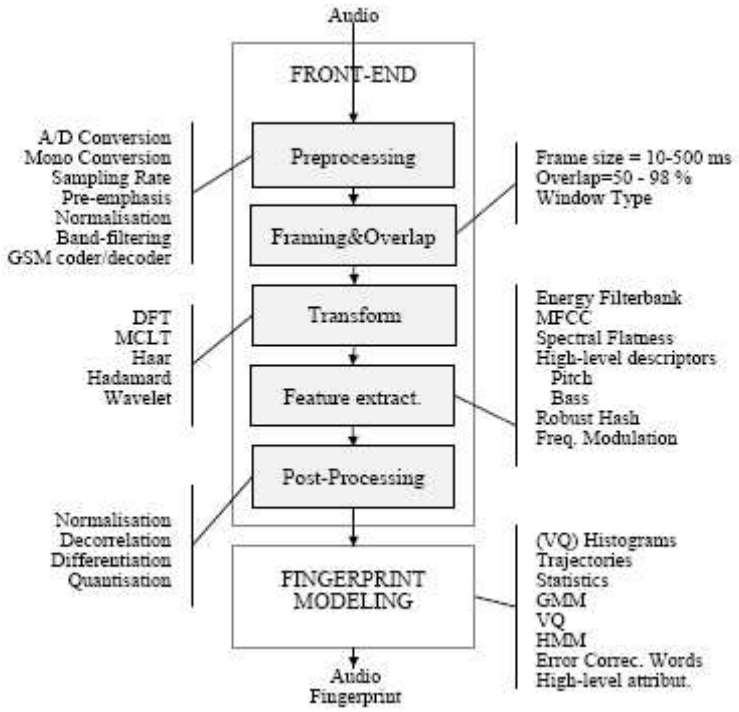


Fig.4: Estructura general del proceso de extracción, con front-end (arriba) y modelado (abajo)

Dada una huella derivada de una grabación, el algoritmo de búsqueda busca en una base de datos de huellas para encontrar la mejor coincidencia. Se necesita, por tanto, una manera de comparar huellas, como puede ser la distancia. Puesto que el número de comparaciones es alto y la distancia puede ser costosa de computar, requerimos métodos que aceleren la búsqueda. Es habitual ver métodos que usan una distancia más simple para rápidamente descartar candidatos y una más correcta pero costosa distancia para un reducido conjunto de candidatos. Hay también métodos para computar distancias off-line (sin conexión) y construir una estructura de datos que permita reducir el número de cómputos a realizar on-line. Unos buenos métodos de búsqueda deberían ser:

- Rápidos: El escaneo secuencial y el cálculo de la distancia pueden ser demasiado lentos para bases de datos enormes.

- Correctos: Deberían devolver los objetos calificados con una nula (o, al menos, baja) Tasa de Falso Rechazo (FRR).

- Uso de memoria eficiente: Deberían requerir poco espacio en memoria.

- Fácilmente actualizable: Deberían permitir fácilmente insertar, borrar y actualizar objetos.

El último bloque del sistema – comprobación de hipótesis (ver fig.3) – computa una medida fiable indicando como de seguro está el sistema sobre una identificación realizada.

2.1.1.1 Front-End

El "front-end" convierte una señal de audio en una secuencia de características relevantes para "alimentar" al bloque de modelado de huellas. En el diseño del "front-end" hay que tener en cuenta varias cosas fundamentales:

- Reducción de la dimensionalidad
- Parámetros perceptualmente significativos (similares a los usados por el HAS, sistema auditivo humano)
- Invarianza o robustez (a distorsiones en el canal, ruido de fondo, etc.)
- Correlación temporal (sistemas que capturen la dinámica espectral)

Ahora vamos a ir detallando los distintos bloques que se observan en la fig.4. En algunas aplicaciones, donde el audio a identificar está codificado, por ejemplo en mp3, es posible saltarse algunos de estos bloques y extraer las características directamente de la representación codificada.

A. Preprocesado

En un primer paso, el audio es digitalizado (si es necesario) y convertido a un formato general: Frecuentemente a un formato de datos en bruto (16 bits PCM), en mono promediando los canales izquierdo y derecho, a una determinada frecuencia de muestreo (que puede ir desde los 5 a los 44,1 Khz.). Algunas veces el audio es preprocesado para simular el canal, por ejemplo: filtrado paso-banda en identificación telefónica. Otros tipos de procesado son un codificador/decodificador GSM en el sistema de identificación de un teléfono móvil, pre-énfasis, normalización de amplitud (limitando el rango dinámico entre (-1,1)).

B. Framing & Overlap (Descomposición en tramas y solapamiento)

Una asunción clave en la medida de características es que la señal puede ser considerada estacionaria en el intervalo de unos pocos milisegundos. Por tanto, la señal se divide en tramas de un tamaño comparable a la velocidad de variación de los eventos acústicos subyacentes. El número de tramas computadas por segundo se llama "frame rate". Para minimizar las discontinuidades al principio y al final de cada bloque, se aplica una ventana. Para asegurar la robustez a la variación de los datos (y también cuando los datos de entrada no están bien alineados) es necesario usar solapamiento. Hay otra vez un compromiso al escoger los valores entre la tasa de cambio del espectro y la complejidad del sistema.

C. Transformaciones lineales: Estimaciones espectrales

La idea detrás de las transformaciones lineales es la transformación del conjunto de medidas en un nuevo conjunto de características. Si la transformada es escogida convenientemente, la redundancia se reduce significativamente. Hay transformaciones óptimas en el sentido de compactación de la información y propiedades de decorrelación, como la Transformada de Karhunen-Loève (KLT) o la Descomposición en Valores Simples (SVD). Estas transformadas, sin embargo, son dependientes del problema y computacionalmente complejas. Por esta razón son habituales transformadas de menor complejidad que usan bases de vectores fijadas. La mayoría de los métodos CBID (Identificación de Información basada en el contenido) por tanto usan transformaciones tiempo-frecuencia estándar para facilitar una compresión eficiente, eliminación de ruido y el subsiguiente procesado. Lourens [11] y Kurth et al. [12] han propuesto, en algunos casos (para secuencias altamente distorsionadas, donde el análisis tiempo-frecuencia también presenta distorsión) el uso de medidas de la potencia de la señal. La potencia también puede ser vista como una distribución tiempo-frecuencia simplificada, con solo una frecuencia.

La transformación más común es la Transformada Rápida de Fourier (FFT). Han sido propuestas otra serie de transformadas, como por ejemplo la Transformada Discreta del Coseno (DCT), la Transformada de Haar o la Transformada de Walsh-Hadamard. Se han realizado estudios [13] que demuestran, por ejemplo que la DFT (Transformada Discreta de Fourier) es menos sensible generalmente al cambio de bits que la de Walsh-Hadamard y que la MCLT (Modulated Complex Transform) presenta propiedades de invarianza a esto [3].

D. Extracción de características

Una vez que tenemos una representación tiempo-frecuencia, se aplican transformaciones adicionales para generar los vectores acústicos finales. En este paso encontramos una enorme diversidad de algoritmos. El objetivo es otra vez reducir la dimensionalidad y, al mismo tiempo, incrementar la invarianza a las distorsiones. Es muy común incluir el conocimiento de las etapas de transducción del sistema auditivo humano para extraer parámetros más significativos desde el punto de vista perceptual. Por tanto, muchos sistemas extraen diversas características realizando un análisis de las bandas críticas del espectro (ver fig.5). Así por ejemplo hay algoritmos que usan los Coeficientes Mel-Cepstrum en Frecuencia (MFCC, que son los coeficientes de la DCT del logaritmo de la energía de la señal de voz en cada banda perceptual, es decir, en cada banda la energía queda ponderada por el correspondiente filtro perceptual del oído). En el sistema de Allamanche et al. [14] la Medida de la Blancura Espectral (SFM, Spectral Flatness Measure), que es una estimación de la calidad de una banda en el espectro. Papaodysseus et al. [15] presentaron los "vectores representativos de las bandas", que son una lista ordenada de los índices de las bandas con tonos prominentes (con picos con amplitud significativa). También se puede usar la energía de cada banda [16]. Por último, en el algoritmo de Philips [10] que presentaremos posteriormente con detalle, ya que es con el que vamos a trabajar, se usa la energía de 33

bandas logarítmicamente escaladas para obtener una cadena de "hash", que es el signo de la diferencia de energía entre las bandas (tanto en el eje del tiempo como en el de la frecuencia).

Las estimaciones espectrales y las características relacionadas son solo inadecuadas cuando se produce distorsión en el canal de audio. En este caso, para caracterizar el comportamiento variante en el tiempo de las señales de audio se puede utilizar un análisis de modulación en frecuencia. Las características corresponderían a la media geométrica de la estimación modulada en frecuencia de la energía de 19 filtros paso de banda separados logarítmicamente [17].

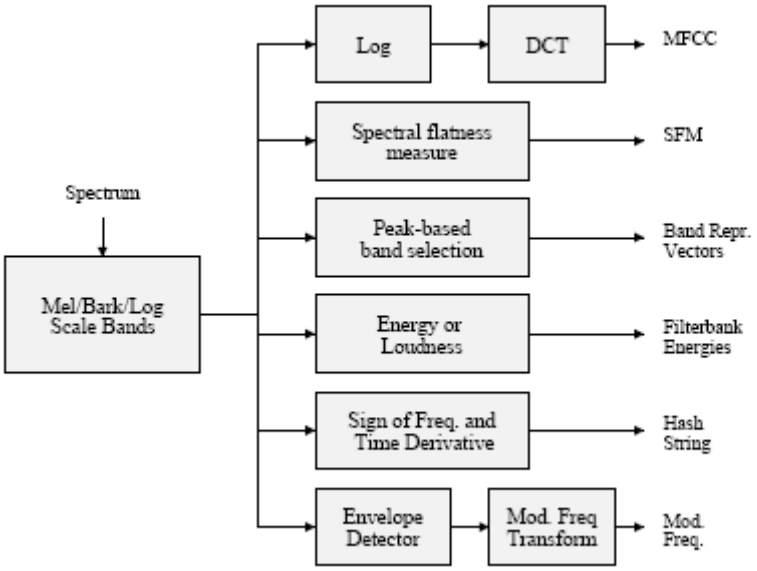


Fig.5 Ejemplos de extracción de características

Los enfoques que se usan en los sistemas de Recuperación de Información de Música incluyen características que han demostrado ser válidos para comparar sonidos: armonicidad, ancho de banda, volumen, ZCR, etc.

Las características usadas más habitualmente son heurísticas, y, por tanto pueden no ser óptimas [18]. Por esa razón, se puede usar una Transformada de Karhunen-Loève Modificada, la Descomposición en Componentes Principales Orientada (OPCA), para encontrar las características óptimas de una forma "no supervisada". Si la PCA (KLT) encuentra un conjunto de direcciones ortogonales que maximizan la varianza de la señal, la OPCA obtiene un conjunto de direcciones posiblemente no ortogonales que tienen en cuenta una serie de distorsiones predefinidas.

E. Post-Procesado

La mayoría de las características descritas hasta ahora son medidas absolutas. Con la intención de caracterizar mejor las variaciones temporales

de la señal, se añaden al modelo de la señal derivadas de más alto orden del tiempo. Por ejemplo, en un sistema propuesto por Cano y Batlle [19], el vector de características es la concatenación de MFCCs, su derivada (delta) y la aceleración (delta-delta), así como ambas derivadas de la energía.

Otros sistemas solo usan la derivada de las características y no las características absolutas [12,14]. Usar las derivadas de las medidas de la señal tiende a amplificar ruido, pero, al mismo tiempo filtra las distorsiones producidas en canales lineales invariantes o lentamente variantes con el tiempo (como una ecualización). La Normalización Media Cepstrum (CNM) también se usa para reducir distorsiones en canales lineales lentamente variantes. Si se usa la distancia euclídea como método de búsqueda, son aconsejables la sustracción de la media y la normalización de la varianza teniendo en cuenta los componentes.

Es relativamente habitual aplicar una cuantización de muy baja resolución a las características: ternaria [13] o binaria [10]. El objetivo de la cuantización es ganar robustez contra las distorsiones, normalizar, facilitar las implementaciones hardware, reducir los requerimientos de memoria y por conveniencia en partes subsiguientes del sistema. Las secuencias binarias hacen falta para extraer palabras correctoras de error en algunos sistemas que las utilizan, como el propuesto por Mihak y Venkatesan [3]. En este, la discretización se diseña para incrementar la aleatoriedad con la intención de minimizar la probabilidad de colisión de huellas.

#### 2.1.1.2 Modelado de Huellas

El bloque de modelado de huellas normalmente recibe una secuencia de vectores de características calculados teniendo en cuenta todas las tramas una por una. Explotar redundancias entre tramas vecinas en el tiempo, dentro de una grabación y a lo largo de toda la base de datos es útil para posteriormente reducir el tamaño de la huella. El tipo de modelo escogido condiciona la métrica de la distancia y también el diseño de algoritmos para una recuperación rápida de información.

Una forma muy concisa de huella se consigue resumiendo las secuencias de vectores multidimensionales de una canción completa (o de una parte de ella) en un vector simple. Así por ejemplo, **Etantrum** calcula el vector a partir de las medias y varianzas de las 16 energías filtradas correspondientes a 30 segundos de audio, dando como resultado una firma de 512 bits. La firma, junto con la información en el formato original de audio es mandada a un servidor para su identificación. La firma TRM de **MusicBrainz** incluye en un vector: la tasa media de cruces por cero, la tasa estimada de "beats" por minuto (BPM), un espectro promediado y algunas características más para representar una pieza de audio (correspondiente a 26 segundos). Estos dos ejemplos aquí nombrados son computacionalmente eficientes y producen una huella muy compacta. Han sido diseñados para aplicaciones como asociar archivos mp3 a metadatos (título, artista, etc.) y pretenden conseguir sobre todo una baja complejidad (tanto en el lado del cliente como en el del servidor) más que una gran robustez.

Las huellas también pueden ser secuencias (trazas, trayectorias) de características. Así, encontramos sistemas que representan la huella como secuencias de vectores binarios. La huella en el sistema de Papaodysseus [15], que consiste en una secuencia de "vectores representativos de las bandas", es codificada en binario por cuestiones de eficiencia de memoria.

Algunos sistemas incluyen atributos de alto nivel musicalmente significativos, tales como el ritmo (BPM) o el tono predominante.

Siguiendo el razonamiento que expusimos antes de la posible sub-optimalidad de las características heurísticas [18], se usan varias capas de OPCA para disminuir las redundancias estadísticas locales de los vectores de características respecto al tiempo. Además de reducir la dimensionalidad, se tienen en cuenta en esta transformación los requisitos extra de robustez ante el intercambio de bits.

En el sistema de Allamanche et al. [14] se explotan las redundancias globales dentro de una canción. Si asumimos que las características de un elemento de audio dado son similares entre ellas, se puede generar una representación compacta agrupando los vectores de características. La secuencia de vectores es, pues, aproximada por un número mucho más bajo de vectores de código representativos, un libro de código. La evolución temporal del audio se pierde completamente con esta aproximación. Además, en este sistema se recogen estadísticas de cortos periodos de tiempo en distintas regiones temporales. Esto da como resultado tanto un mejor reconocimiento, ya que las dependencias temporales son tenidas en cuenta, como una búsqueda de coincidencias más rápida, ya que la longitud de cada secuencia también se reduce.

El sistema de Cano y Batlle [19] usa un modelo que explota más la redundancia global. La base lógica está muy inspirada por la investigación del habla. En el habla, un alfabeto de clases de sonido, es decir los fonemas, pueden usarse para segmentar una colección de datos hablados en bruto en texto, logrando una gran reducción de la redundancia sin mucha pérdida de información. Análogamente, podemos ver un trozo de música como secuencias construidas concatenando clases de sonidos de un alfabeto finito. En un gran número de canciones pop aparecen sonidos de batería "perceptualmente equivalentes". Esta aproximación nos conduce a una huella que consiste en secuencias de índices a un conjunto de clases de sonidos representativo de una colección de elementos de audio. Las clases de sonido son estimadas vía agrupamiento sin supervisión y modeladas con Modelos Ocultos de Markov (HMMs). El modelado estadístico del transcurso de la señal en tiempo permite una reducción de la redundancia local. La representación de la huella como secuencias de índices a clases de sonidos conserva la información de la evolución del audio a través del tiempo.

En [3] las secuencias discretas son mapeadas a un diccionario de palabras correctoras de errores. En [12], el método de indexado está basado en los códigos correctores de errores.

### 2.1.2 Distancias y métodos de búsqueda

2.1.2.1. Distancias

Las métricas de distancia están altamente relacionadas con el tipo de modelo escogido. Cuando se comparan secuencias de vectores es habitual usar una correlación. La distancia euclídea, o versiones ligeramente modificadas de la misma que tratan con secuencias de diferente longitud, se usa por ejemplo en [20]. En [17], la clasificación es el Vecino Más Cercano usando una estimación de la entropía cruzada. En los sistemas donde las secuencias de vectores de características están cuantizadas, se usa una distancia Manhattan (o Hamming cuando la cuantización es binaria). Mihak [3] sugiere que otra métrica de error, que llaman "Pseudo norma exponencial" (EPN), podría ser más apropiada para distinguir mejor entre valores cercanos y distantes con un énfasis más fuerte que el lineal.

Hasta ahora hemos presentado una estructura de trabajo para la identificación que sigue un mismo paradigma para la búsqueda de coincidencias: tanto los patrones de referencia –las huellas almacenadas en la base de datos- como el patrón de prueba –la huella extraída a partir del audio desconocido- están en el mismo formato y son comparados según alguna métrica de distancia, por Ej.: distancia Hamming, una correlación, etc. En algunos sistemas, sólo los elementos de referencia son realmente "huellas" –modeladas compactamente como un libro de códigos o una secuencia de índices a HMMs. En estos casos, las distancias son computadas directamente entre la secuencia de características extraídas a partir del audio desconocido y las huellas del audio de referencia almacenadas en la base. En [14], la secuencia del vector de características es comparada con los distintos libros de código usando una métrica de distancia. Para cada libro, se acumulan los errores. El elemento desconocido es asignado a la clase que dé el menor número de errores acumulados. En [21], la secuencia de características es comparada con las huellas (una concatenación de índices apuntando a clases de sonidos HMM) usando el algoritmo de Viterbi. Se selecciona el recorrido más probable en la base de datos.

2.1.2.2- Métodos de Búsqueda

Más allá de la definición de una métrica de distancia para la comparación de huellas, un asunto fundamental para la usabilidad de un sistema es cómo de eficientemente realiza las comparaciones entre el audio desconocido y posiblemente, millones de huellas. Un enfoque de fuerza bruta, que compute las similaridades entre la huella de la grabación desconocida y las que están almacenadas en la base de datos puede ser inviable. El tiempo para encontrar la mejor coincidencia en este método lineal o secuencial es proporcional a N*c(d())+E, donde N es el número de huellas almacenadas y c(d()) el tiempo que se necesita para encontrar una sola similaridad y E tiene en cuenta algún tiempo extra de CPU.

En general los métodos dependen de la representación de la huella, pero vamos a hacer una clasificación más o menos general de los enfoques propuestos en la literatura.

- *Pre-computar distancias offline*: Uno no puede calcular similaridades offline con la huella candidata, puesto que ésta no ha sido presentada previamente al sistema. Sin embargo uno pude computar distancias entre las huellas ya almacenadas y construir una estructura de datos para reducir el número de evaluaciones de similaridad una vez que se presenta la huella. Es posible construir offline conjuntos de clases de equivalencia, calcular algunas similaridades online para descartar algunas clases y buscar exhaustivamente entre el resto. Si la medida de similaridad es una métrica, por ejemplo, la medida es una función que cumple las siguientes propiedades: positividad, simetría, reflexividad y la desigualdad triangular, hay métodos para reducir el número de evaluaciones y garantizar que no hay falsos rechazos. Los espacios vectoriales permiten el uso de eficientes métodos de acceso espacial ya existentes.

- *Filtrado de candidatos improbables con una medida de similaridad simple:* Otra posibilidad es usar una medida de similaridad más simple para eliminar rápidamente muchos candidatos y la más precisa y compleja en el resto. Como se demuestra en [22], para garantizar que no se produzcan falsos rechazos, la medida simple utilizada para descartar hipótesis poco prometedoras debe limitar por debajo a la medida más cara (fina).

- *Indexado de archivos inverso:* Un método de búsqueda muy eficiente es el uso de indexado de archivos inverso. Haitsma et al. propusieron un índice de posibles trozos de una huella que apuntan a posiciones en las canciones. Dado que un trozo de la huella candidata está libre de errores (coincidencia exacta), se puede obtener eficientemente una lista de canciones candidatas para buscar exhaustivamente en ella. En [19], se usan indexados y heurísticas similares a las usadas en biología computacional para la comparación del ADN para acelerar la búsqueda en un sistema donde las huellas son secuencias de símbolos. Kurth et al. [12] presentan un índice que usan palabras de código extraídas de secuencias binarias que representan el audio. A veces estos enfoques, aunque son muy rápidos hacen suposiciones sobre los errores permitidos en las palabras usadas para construir el índice, lo que podría resultar en falsos rechazos.

- *Reducción de candidatos:* Una optimización simple para acelerar la búsqueda es mantener el mejor resultado obtenido hasta el momento. Podemos abandonar el cálculo de una medida de similaridad si llegados a un cierto punto sabemos que ya no vamos a mejorar el mejor resultado obtenido hasta el momento. Algunas medidas pueden aprovecharse de algunas estructuras como árboles de sufijos para evitar cálculos duplicados [23]. Millar et al. [24] proponen un árbol para evitar redundancias en el cálculo de la mejor coincidencia en una estructura de trabajo construida con la representación de huellas de [10]. Combinando la estructura de árbol con una heurística del "mejor hasta ahora" se evita no sólo la computación de similaridad de la huella actual sino que también la de todas las huellas que tengan un inicio común.

- *Otros enfoques:* En [25], el almacén de las huellas se separa en dos bases de datos. La primera y más pequeña guarda las huellas con mayor probabilidad de aparición, por ejemplo las canciones más populares del momento y la otra guarda el resto. Las huellas candidatas son confrontadas

primero con la más pequeña y más probable y sólo cuando no se encuentra ninguna coincidencia el sistema examina la segunda base de datos. Los sistemas de producción de hecho usan varios de los métodos de aceleración descritos más arriba. El de Wang y Smith [25], por ejemplo, además de buscar primero en el almacén de canciones más populares, usa un indexado de archivos inverso para acceder rápidamente a las huellas junto con una heurística para filtrar candidatos poco prometedores antes de buscar exhaustivamente con la medida de similaridad más precisa.

### 2.1.3 Comprobación de Hipótesis

Este último paso pretende responder si el elemento en cuestión está o no en el almacén de datos a identificar. Durante la comparación de la huella extraída con la base de datos de huellas, se obtienen resultados (a partir de las distancias). Para poder decidir si hay una identificación correcta, el resultado debe estar por encima de un determinado umbral. No es fácil de escoger dicho umbral ya que depende de: el modelo usado para la huella, la información del elemento, la similaridad de las huellas de la base y el tamaño de la misma. Mientras más grande sea la base, mayor es la probabilidad de indicar una coincidencia erróneamente, lo que es un falso positivo. La tasa de falso positivo se llama también tasa de falsa aceptación (FAR) o probabilidad de falsa alarma. La tasa de falso negativo también es llamada tasa de falso rechazo (FRR). La nomenclatura está relacionada con las medidas de evaluación del comportamiento del sistema de Recuperación de Información: Precisión y Memoria.