(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 556 276**

(21) Número de solicitud: 201300721

(51) Int. Cl.:

G06K 9/00 (2006.01)

(12) SOLICITUD DE PATENTE A1

(22) Fecha de presentación:

31.07.2013

(43) Fecha de publicación de la solicitud:

14.01.2016

(56) Se remite a la solicitud internacional:

PCT/ES2014/000131

(71) Solicitantes:

UNIVERSIDAD DE SEVILLA (85.0%)
Paseo de las Delicias s/n - Pabellón de Brasil
41013 Sevilla ES y
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTIFICAS (15.0%)

(72) Inventor/es:

ARJONA LÓPEZ, María Rosario y
BATURONE CASTILLO, María Iluminada

(74) Agente/Representante:

GONZÁLEZ CARVAJAL, Ramón

(54) Título: **Método de identificación de huellas dactilares y dispositivo que hace uso del mismo**

(57) Resumen:

Método de identificación de huellas dactilares y dispositivo que hace uso del mismo.

Se describen en este documento un método y un dispositivo que permiten generar un vector de características de una huella dactilar humana mediante una serie de procesos realizados a partir de la imagen de dicha huella. Con esos vectores se puede realizar una clasificación, indexación o determinación de identidad de huellas (y, por tanto, de individuos), así como proteger secretos o generar claves criptográficas a partir de huellas.

La mayoría de los métodos de identificación reportados emplean vectores de características y técnicas para extraerlos y compararlos que no son adecuadas para dispositivos electrónicos con recursos de cómputo y almacenamiento reducidos, como una FPGA o un circuito integrado de aplicaciones específicas. Por el contrario, el método propuesto en esta invención si es adecuado, ofreciendo buenas prestaciones en cuanto a tiempos de extracción de vectores (situándolos por debajo del milisegundo para tamaños de huellas estándares), tiempos de emparejamiento y ordenación (valores despreciables de pocos nanosegundos por usuario) y requerimientos de memoria (poco más de cien bytes por huella). Los usos principales de la invención se sitúan en sistemas automáticos de identificación de huellas pequeños, portables, baratos y/o seguros donde el usuario está presente y quiere identificarse.

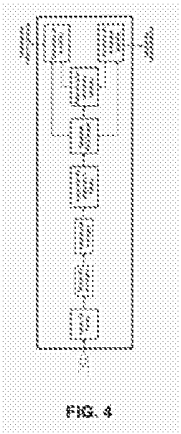

FIG. 4

ES 2 556 276 A1

DESCRIPCIÓN

MÉTODO DE IDENTIFICACIÓN DE HUELLAS DACTILARES Y DISPOSITIVO QUE HACE USO DEL MISMO

D E S C R I P C I Ó N

OBJETO DE LA INVENCIÓN

La presente invención se enmarca en el campo de los sistemas y métodos de identificación biométrica.

El objeto de la invención consiste en un método y un dispositivo que permiten generar un vector de características de una huella dactilar humana mediante una serie de procesos realizados a partir de la imagen de dicha huella. Con esos vectores se puede realizar una clasificación, indexación o determinación de identidad de huellas (y, por tanto, de individuos), así como proteger secretos o generar claves criptográficas a partir de huellas.

ANTECEDENTES DE LA INVENCIÓN

El uso de huellas dactilares como característica biométrica está ampliamente extendido para aplicaciones de identificación de individuos, control de acceso, etc., por su alta discriminación y porque los usuarios aceptan con normalidad el hecho de introducir la huella en un dispositivo de captura (proporciona facilidad de uso al ser una técnica no intrusiva). Es una de las características biométricas que con mayor éxito se han aplicado en la actividad forense y policial y, más recientemente, en sistemas de control de acceso. Los sistemas automáticos de identificación de huellas (AFIS, "Automatic Fingerprint Identification Systems") requieren comparar una huella de entrada con las huellas almacenadas en la base del sistema. En estas aplicaciones, los individuos se registran previamente en una base mediante las características de sus huellas dactilares. Posteriormente, cuando se quiere identificar un individuo, se vuelven a extraer las características de sus huellas y se comparan con las características almacenadas en la base de huellas.

Actualmente es un desafío la realización eficiente de sistemas de identificación que

utilicen bases con un número elevado de huellas y proporcionen tiempos de respuesta inmediatos, a la vez que exactitud en la identificación. El tiempo necesario para la identificación de un individuo (*tidentificación*) se puede expresar como:

$$tidentificación = textracción + temparejamiento*N + tdecisión$$

donde *textracción* es el tiempo invertido en la extracción de características de la huella, *temparejamiento* es el tiempo empleado para comparar las características extraídas de la huella de entrada con cada una de las N características almacenadas en la base, y *tdecisión* es el tiempo empleado para decidir cuál de los N individuos registrados es el candidato elegido como poseedor de la huella de entrada, en el caso de una aplicación de identificación, o el tiempo empleado en generar una lista reducida con los M individuos candidatos a poseerla (con M mucho menor que N), en el caso de una aplicación de indexación.

El valor de *textracción* es mucho mayor que el de *temparejamiento* porque el proceso de extracción de características es mucho más complejo que el del emparejamiento. Por ejemplo, el algoritmo de extracción MINDTCT desarrollado por el NIST ("National Institute of Standards and Technology") es un orden de magnitud más lento en una misma plataforma PC con un procesador Intel Core i7 que el algoritmo de emparejamiento BOZORTH98 de NIST, por ejemplo, más de 200 ms de media para el primero y menos de 20 ms para el segundo (tanto MINDTCT como BOZORTH98 disponibles en NIST Biometric Image Software (NBIS), http://www.nist.gov/itl/iad/ig/nbis.cfm). Sin embargo, aunque el *temparejamiento* sea menor, al estar multiplicado por N, la búsqueda en la base puede ser demasiado lenta para aplicaciones en tiempo real.

Para reducir el número de comparaciones de la huella de entrada con respecto a las huellas almacenadas en la base del sistema, se utilizan métodos denominados de "clasificación exclusiva" en los que las huellas se distribuyen en grupos disjuntos pre-establecidos, de forma que la huella de entrada se clasifica en uno de esos grupos y sólo se compara con las huellas registradas de ese grupo. El esquema comúnmente utilizado sigue las propuestas de Galton y Henry (E.R. Henry, "Classification and Uses of Finger Prints", London: Routledge, 1900) que distinguen cinco grupos de huellas ("arch", "whorl", "tended arch", "left loop", y "right loop"). El problema es que la mayor parte de las huellas pertenecen sólo a tres grupos ("right loop", "left loop", y "whorl"), con lo cual no se produce una gran reducción del número de comparaciones en una

base de huellas extensa (R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint classification by directional image partitioning", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, pages 402-421, 1999). Otro problema de los métodos de clasificación exclusiva es que determinar a què clase pertenece una huella se trata en muchos casos de una operación ambigua. Estos inconvenientes han propiciado la aparición de los métodos denominados de "clasificación continua", que asignan un vector de características numéricas a cada huella. De esta forma, en la fase de indexación ("indexing"), se crea una tabla o base indexada de huellas. En la fase de recuperación ("retrieving"), ante una huella de entrada con un vector de características representativo, se calcula la similitud entre el vector de entrada y los almacenados mediante una medida de distancia, de forma que puede obtenerse una lista reducida con los M candidatos màs similares (con M mucho menor que N). A continuación se pueden aplicar otros métodos de identificación para determinar el individuo correcto de entre los seleccionados.

Una huella dactilar es una imagen donde se distinguen crestas y valles en escala de grises. La comparación directa de las huellas en escala de grises no ofrece buenos resultados, además de ser una técnica costosa en cuanto a cómputo y almacenamiento. Es mejor procesar la imagen para obtener características distintivas y compactas. Se suele hablar de características de 3 niveles, según el nivel de detalle con el que se analice la huella. Las características de nivel 1 se obtienen de un análisis global de la huella. Un ejemplo es la imagen (campo o mapa) direccional (también llamado imagen, campo o mapa de orientaciones), que contiene las orientaciones locales de las crestas respecto a un eje de referencia. Otro ejemplo son los puntos singulares, que son puntos de la huella donde convergen (núcleos o "cores") o divergen ("deltas") las orientaciones y en torno a los cuales se encuentra la mayor parte de la información distintiva de una huella. Las características de nivel 2 se obtienen de un análisis local y en más detalle de la huella. Un ejemplo son las tradicionales minucias, que pueden ser terminaciones (lugares donde las crestas acaban) y bifurcaciones (lugares donde las crestas se separan en otras dos). Las características de nivel 3 (como poros o crestas incipientes) se obtienen tras un análisis muy detallado de la huella que requiere una adquisición de la misma con muy buena calidad. La extracción de características es tanto más distintiva y a la vez más costosa cuanto mayor sea el nivel.

Hasta el momento, apenas se han reportado técnicas que empleen características de nivel 3. Sí se han reportado varias técnicas que emplean características de nivel 2. El

paso previo a todas estas técnicas es extraer las minucias. Los procesos de extracción de minucias son complejos debido a que la imagen de la huella tiene que ser preparada para localizar los mínimos detalles (las minucias son poco robustas ante el posible ruido que la imagen de una huella puede presentar). Por ejemplo, la técnica de extracción de minucias más común requiere que la imagen de la huella sea mejorada, segmentada, binarizada, y adelgazada. Un ejemplo ampliamente conocido de algoritmo detector de minucias es el MINDTCT comentado anteriormente. La comparación de huellas basada en minucias es relativamente lenta y, por lo tanto, no es adecuada para indexación (por ejemplo, usando algoritmos como BOZORTH98 comentado anteriormente). Las técnicas de indexación que usan características de nivel 2 aplican un post-procesado sobre las minucias extraídas. Así por ejemplo, en la Patente "Method for searching fingerprint database based on quantum algorithm", CN102495886 (A), se elige un conjunto de minucias y se expresan en coordenadas polares respecto a unos puntos de referencia que también se eligen convenientemente. En la Patente "Fast fingerprint identification and verification by minutiae pair indexing", WO2008135521 (A2), se emplea indexación mediante parejas de minucias. En el documento "Methods and systems for automated fingerprint recognition", WO2008098357 (A1), se asocian patrones de minucias a las huellas. El método y sistema propuesto en el documento "Progressive fingerprint matching system and method", US2004062426 (A1) se basa en emparejamiento de huellas por minucias. En el documento "Fingerprint verification", US2005058325 (A1), se muestrean las minucias y se ordenan en subconjuntos. En el documento "System and method for matching (fingerprint) images an aligned string-based representation", US6185318 (B1), se emplean las minucias como puntos de referencia. En el documento "Methods and related apparatus for fingerprint indexing and searching", US6181807 (B1), se extraen minucias de las huellas y se comparan en el proceso de búsqueda. En el documento "Vector based topological fingerprint matching", WO9532482 (A1), se emplean las posiciones de las minucias y se les asigna un número de índice. Otras técnicas conocidas son las que emplean tripletas de minucias. En el documento "Fingerprint identification method based on triangulation and LOD technology", CN101620677 (A), se emplea una tecnología de triangulación para extraer vectores de características globales y locales. Anteriormente se ha propuesto usar todas las tripletas posibles que pueden formarse con cada minucia. Otros autores proponen aplicar la triangulación de Delaunay de orden 1 a las coordenadas de las minucias para asignar una estructura topológica única a cada huella. En otras técnicas conocidas en el arte se usan triángulos de Delaunay de orden superior a 1 para extraer más información geométrica. Otra técnica que usa características de nivel 2 es

la representación MCC ("Minutia Cylinder-Code"), que asigna a cada minucia una estructura local que codifica la probabilidad de encontrar minucias a su alrededor, con una diferencia de orientación similar a un valor dado.

Las técnicas que usan características de nivel 1 ofrecen prestaciones competitivas con mucho menor coste computacional. De hecho, la extracción de la imagen direccional es un paso necesario en la mayoría de los algoritmos de extracción de minucias y de comparación de huellas, por lo que podría decirse que su coste es cero. Estas técnicas se diferencian unas de otras en cómo extraen características representativas y compactas de la imagen direccional. Por ejemplo, en técnicas conocidas se emplea un modelo de orientación de huellas basado en expansiones de Fourier bidimensionales para adaptarse a la periodicidad intrínseca de las orientaciones. En otras soluciones se emplean un conjunto de momentos complejos polares (PCMs) para extraer características de la imagen direccional invariantes a rotaciones de la huella.

En el documento "New fingerprint database retrieval method", CN102368242 (A), se emplean puntos singulares, información sobre la relación entre puntos singulares y momentos invariantes ortogonales. En el documento "Method for rapidly calculating fingerprint similarity", CN101996318 (A), se buscan unidades topológicas similares entre cada par de huellas a comparar, se expanden para obtener unidades topológicas similares mayores y se agrupan para obtener una medida de similitud global.

La mayoría de las soluciones para identificación e indexación de huellas son implementaciones software que involucran algoritmos de un coste computacional elevado en términos de tiempo y recursos. El coste es elevado, no sólo para la extracción de las características, sino incluso para el algoritmo de emparejamiento que calcula la similitud entre las características de la huella de entrada y las almacenadas.

Las soluciones para indexación normalmente se evalúan mostrando, para una razón de penetración dada (un porcentaje promedio de la base de huellas que se va a analizar), la razón de error (porcentaje de huellas de entrada cuyo registro no se recupera de entre la lista con similitudes más altas con esa penetración). Ésta es la razón de penetración definida como el porcentaje de candidatos que se consideran para ver si entre ellos está el poseedor verdadero de la huella de entrada (M/N). Otra medida normalmente analizada para evaluar la bondad de una técnica de indexado es la tasa promedio en un escenario de búsqueda incremental ("incremental search

scenario"), que se calcula como la tasa promedio que hay que llevar a cabo cuando no se quieren cometer errores en la recuperación del poseedor de la huella de entrada. Los tiempos promedios que se invierten en la búsqueda no suelen reportarse y tampoco los requerimientos de memoria. En los trabajos que sí se reportan, los tiempos que se muestran son de realizaciones sobre PCs: 67 ms en un Intel Pentium 4 a 2.26 GHz y 1.6 ms, 14 ms o 16 ms (según la técnica) en un Intel Core 2 Quad a 2.66 GHz sobre 2000 huellas de la base NIST DB4.

En las soluciones para identificación, una vez generado el vector de características y comparado con los vectores almacenados, no se genera una lista ordenada de huellas sino que se establece un umbral de emparejamiento para aceptar o rechazar si un individuo es quién dice ser. En este caso, lo que se miden son razones de falso rechazo (FRR, "False Rejection Rate") y de falsa aceptación (FAR, "False Acceptance Rate"), en lugar de tasas de penetración, como en el caso de indexación. También denominadas FNMR ("False Non-Match Rate") o FMR ("False Match Rate"), respectivamente.

El contexto de aplicación de estas soluciones suele ser el forense y policial, en el que las huellas que se tienen de un individuo se han adquirido sin su cooperación (por ejemplo, porque se trate de identificar a un fallecido o a un delincuente). Se denomina un contexto "fuera de línea", por lo que las capturas pueden ser de mala calidad y los algoritmos pueden realizarse sobre PCs sin requerimientos especialmente restrictivos de velocidad y consumo de memoria. Existen bases de huellas, como las de la "Fingerprint Verification Competition" (FVC), que están construidas con muchas capturas de mala calidad y mal adquiridas para probar la bondad de técnicas complejas de identificación e indexado.

Un contexto de aplicación diferente es el que se denomina "en línea", en el que el usuario de un sistema de reconocimiento coopera con el sistema porque quiere autenticarse (por ejemplo, en un sistema de control de acceso). En este caso, las capturas son de mucha mejor calidad e, incluso, se puede interactuar con el usuario para que introduzca bien su huella. En esta línea, se conoce una solución para la estimación de la calidad de una huella basada en un algoritmo para la extracción de puntos singulares que satisface las restricciones en términos de recursos, tiempo de respuesta y resultados de reconocimiento impuestos para una aplicación de adquisición inteligente en un dispositivo hardware empotrado. En el documento "Method for authenticating an individual by use of fingerprint data", US7136514 (B1),

se tiene en cuenta que el individuo que introduce su huella mediante un sensor de barrido puede barrer el dedo en diferentes direcciones respecto al eje del sensor. En el documento "Fingerprint matching", GB2320352 (A), se emplean índices de calidad en la extracción del vector de características para luego emplearlos en el cálculo del emparejamiento entre huellas.

Los requerimientos de velocidad sí son restrictivos en un contexto de aplicación en línea, porque la operación debe ser en tiempo real. La facilidad de uso del sistema y su precio también pueden ser requerimientos importantes en este contexto. El usuario puede emplear cómodamente un dispositivo electrónico pequeño, ligero y barato, por ejemplo, una tarjeta o token con recursos reducidos. Los recursos de los que dispone una tarjeta inteligente o un DSP para dispositivos empotrados son mucho menores que los de un PC: CPUs de 50 ó 100 MHz y memorias disponibles (ROM, EEPROM y RAM) de unas cuantas decenas de KBytes, en el mejor de los casos.

Los tiempos aumentan mucho si la plataforma donde se implementan los algoritmos tiene pocos recursos. Por ejemplo, el algoritmo MINDTCT adaptado y ejecutado sobre un procesador LEON2 empotrado invierte en su ejecución casi tres órdenes de magnitud más que en la plataforma PC (unos 100 s, según se ha reportado). Por este motivo, la extracción de características se suele hacer sobre una plataforma tipo PC y las soluciones que emplean tarjetas o DSPs para reconocimiento en línea por huella dactilar sólo implementan el algoritmo de emparejamiento entre las características almacenadas y las que le llegan del exterior. Además, las características almacenadas suelen ser las de un individuo sólo (emparejamiento 1 frente a 1 en vez de 1 frente a N). A esta solución se le suele denominar "match on card". Aun así, se han propuesto soluciones en las que se rediseña el software del algoritmo de emparejamiento, se emplea aritmética de punto fijo y se extiende el conjunto de instrucciones del procesador empotrado para acelerar la ejecución. Los algoritmos de "match on card" se han estudiado recientemente en la campaña MINEX II organizada por el NIST. Los resultados obtenidos demuestran que son menos precisos que los que se ejecutan sobre una plataforma PC. Otra opción para la implementación en sistemas empotrados es emplear FPGAs ("Field Programmable Gate Arrays"). En las FPGAs se pueden implementar coprocesadores hardware que aceleren la ejecución de los algoritmos. Así, por ejemplo, existe una solución que propone la correlación directa de imágenes en escala de grises, usando una FPGA Virtex 4. Para aplicaciones de indexación de huellas, existe una técnica que crea una base de huellas cuyos índices se basan en la extracción de minucias, en la que la base de huellas y la búsqueda sobre la base se

implementan en placas PCI basadas en FPGAs mientras que la extracción de minucias se realiza en el PC al que se conectan las placas.

En términos de seguridad, es muy interesante que todo el proceso de extracción de características, su almacenamiento y emparejamiento se pueda realizar en el mismo dispositivo, lo que se denomina "authentication on card", porque así las características distintivas de los individuos se circunscriben dentro de un perímetro mucho más pequeño, que, por tanto, es más fácil de controlar y defender. En esta línea, existen soluciones donde se implementa un algoritmo de extracción de minucias en una FPGA Spartan 3 y soluciones donde se implementa un sistema de reconocimiento basado en la localización de puntos singulares sobre una placa RC203E de Celoxica equipada con una FPGA Virtex II. En vez de simplificar los algoritmos a implementar, la solución analizada en una segunda opción es emplear FPGAs que se reconfiguran según la tarea a realizar (extracción de la imagen direccional, mejora y segmentación de la imagen de la huella, binarización, suavizado, adelgazamiento, detección de minucias, alineamiento y emparejamiento). Esta idea de "authentication on card" aparece en varias patentes. Entre ellas, podemos citar "Biometric identity verification system and method", US 20080223925 A1 y, entre las más recientes, "Smart card system with ergonomic fingerprint sensor and method of using", US 8276816 B2.

Otro gran problema de los sistemas de identificación basados en huella dactilar, ya no relacionado con la implementación sino con su propia naturaleza, es el de la falta de diversidad en la obtención de características distintivas. Por ejemplo, un usuario dispone como mucho de 10 dedos en sus manos. Si se descubre que un impostor se apodera de las características de uno de sus dedos, el individuo ya ha perdido el 10% de sus posibles características. Si el impostor se apodera de las características de los 10 dedos, el individuo ya no puede registrarse en ningún sistema. Este problema también se denomina como la escasa revocabilidad del sistema, es decir, es difícil generar nuevas características cuando otras han sido descubiertas o comprometidas.

Para evitar que las características puedan ser comprometidas, se han propuesto sistemas que las transforman mediante funciones no invertibles, como las funciones hash, de forma que recuperar las características originales a partir de las transformadas sea prácticamente inviable desde el punto de vista computacional. En estos sistemas, la medida de similitud o emparejamiento entre las características de entrada y las previamente registradas se realiza en el espacio transformado. Por eso hay que analizar muy bien cómo afecta la transformación a las prestaciones del

sistema resultante (por ejemplo, en cuanto a tasas de falsa aceptación y falso rechazo). Esta solución no solventa el problema de la diversidad porque la transformación de características no incrementa el número de posibles características. Para ello, se puede emplear un número aleatorio (conocido como "salt" en las técnicas criptográficas) que se combine con las características originales, de forma que su transformación sea diferente aun cuando emplee la misma información biométrica. El "salt" actúa como una palabra de paso que el individuo debe introducir en el sistema, además de su huella dactilar. La desventaja de esta solución es que la palabra de paso y las características transformadas no deben hacerse públicas para incrementar la seguridad del sistema.

Otro esquema que posee la ventaja de no requerir almacenamiento seguro de información es el de los denominados sistemas de cifrado biométricos ("biometric criptosystems"). Se basan en combinar las características biométricas orginales (sin ninguna transfomación) con información adicional de tal manera que los datos resultantes, conocidos como "helper data" (datos de ayuda), puedan ser públicos. Las dos técnicas más empleadas en sistemas de cifrado han sido las denominadas *Fuzzy Commitment* y *Fuzzy Vault*. Fuzzy Commitment es un esquema más básico y más simple que Fuzzy Vault. Como contrapartida, Fuzzy Commitment requiere que el vector de características sea un vector binario, ordenado y de longitud fija. Fuzzy Commitment se implementa en las dos fases siguientes:

- Fase de registro: la característica biométrica se combina (usualmente mediante una operación XOR, en el caso de características binarias) con una palabra código generada mediante la aplicación de un código corrector de errores a un número aleatorio, clave o palabra de paso (contraseña). El resultado es una información de ayuda que no necesita ser almacenada de forma segura.

- Fase de verificación o recuperación de secreto: la nueva característica biométrica, ligeramente diferente a la que se obtuvo durante el registro (lo cual es habitual), se combina con la información de ayuda, que es pública, para recuperar la palabra código (aplicando un código corrector de errores). A partir de la información recuperada en esta fase, también se puede generar una clave criptográfica.

Hoy en día, la mayoría de los métodos de identificación por huella dactilar reportados (así como los sistemas de cifrado basados en ellos) emplean vectores de características y técnicas para extraerlos y compararlos que no son adecuados para dispositivos electrónicos con recursos de cómputo y almacenamiento reducidos. Por eso son necesarias soluciones de identificación por huella dactilar válidas para

sistemas de cifrado que, manteniendo buenos resultados de reconocimiento, sean adecuadas para dispositivos electrónicos de bajo consumo de potencia, con capacidad de cálculo limitada y que no requieran un "hardware" potente y/o voluminoso que emplee grandes recursos.

DESCRIPCIÓN DE LA INVENCIÓN

Se propone un método y un dispositivo para implementar el método que permiten, mediante una serie de procesos y a partir de una imagen capturada de una huella dactilar, generar un vector de características basado en características de nivel 1, en concreto en la segmentación de la imagen direccional en regiones homogéneas. El método permite obtener una cadena de bits de longitud fija a partir de la huella capturada preferentemente en línea mediante un sensor de huella de los que se emplean en los sistemas automáticos de identificación (óptico, capacitivo, etc.).

En una posible realización el método aquí descrito puede ser adaptado y emplearse en aplicaciones de clasificación en las que las huellas de los individuos se distribuyen en grupos más o menos disjuntos, según el algoritmo de agrupamiento que se aplique sobre los vectores de características. El método también puede emplearse en aplicaciones de indexado e identificación/autenticación, en las que se registran los individuos mediante los vectores de características que se generan en una fase de indexado o registro. En la fase de recuperación o verificación, dada una huella de entrada, se genera una lista ordenada de individuos candidatos a poseer esa huella (en el indexado) o se identifica el mejor candidato (en aplicaciones de identificación). En el caso de autenticación, se registra un solo individuo y en la fase de verificación se mide la similitud entre el vector generado y el almacenado. Si supera un umbral, el individuo se autentica. No se autentica en otro caso.

El método aquí descrito se puede emplear en multi-biometría. Dadas varias muestras de huellas capturadas de dedos diferentes de un mismo individuo, los vectores obtenidos de cada dedo se concatenan para obtener un vector de identificación digital del individuo. Y también, dadas varias muestras de huellas capturadas de un mismo dedo, los vectores obtenidos de cada muestra se concatenan para obtener un vector de identificación de la huella.

El método puede emplearse en aplicaciones de identificación (y autenticación) por doble factor porque el vector que genera puede combinarse fácilmente con vectores derivados de claves o contraseñas.

El método puede emplearse en los denominados esquemas de protección de plantilla ("template"). En particular, es muy adecuado para sistemas de cifrado basados en la técnica Fuzzy Commitment, porque el vector generado es binario, ordenado y de longitud fija. En estos esquemas, el método de la invención ofrece las ventajas de no-reversibilidad de los vectores transformados y diversificación de los vectores generados, manteniendo la precisión en la identificación (y autenticación).

El método propuesto en esta invención puede implementarse en un dispositivo electrónico de bajo coste (con recursos de cómputo y memoria reducidos), como por ejemplo, una FPGA o un circuito integrado de aplicaciones específicas, ofreciendo buenas prestaciones en cuanto a tiempos de extracción de características (situándolos por debajo del milisegundo para tamaños de huellas estándares), tiempos de emparejamiento y ordenación de candidatos (valores despreciables de pocos nanosegundos por candidato) así como requerimientos de memoria (poco más de 100 bytes por huella). Así se puede conseguir una solución muy segura porque toda la información biométrica de los individuos puede estar confinada dentro del mismo dispositivo electrónico y no salir de él.

El objeto de la invención se basa en un contexto de aplicación en línea, es decir, el usuario del dispositivo colabora para identificarse, a diferencia de otros contextos de aplicación de identificación por huella dactilar, como el forense o el policial, en los que el usuario no colabora (porque está fallecido o no desea que lo identifiquen). En un contexto de aplicación en el que el individuo quiere registrarse e identificarse, las características se extraen con calidad. En cualquier caso, el dispositivo que implementa la invención permite evaluar la calidad del proceso y la interacción en línea con el usuario para evitar capturas defectuosas de huellas.

El método de identificación se basa en generar un vector de características de la huella dactilar para su identificación a partir de una primera imagen de la misma en escala de grises que contiene crestas y valles de la huella, para ello se llevan a cabo los siguientes pasos:

a) determinar para cada píxel de la primera imagen, p_{ij} (donde ij hacen referencia a la fila y columna del píxel en la imagen), el gradiente de la

intensidad de la imagen (de los niveles de grises) en ese píxel,

b) determinar la dirección del gradiente mediante un ángulo α_{ij} con respecto a un eje de referencia,

c) dividir el intervalo de valores posibles de ángulos, α_{ij}, en G sub-intervalos $(g_1, \ldots, g_G)$ que no se solapan y cuya unión da lugar al intervalo completo de posibles valores, englobando cada sub-intervalo g_k ángulos desde un valor α_{k-1} hasta α_k,

d) etiquetar cada g_k sub-intervalo con una etiqueta, c_k,

e) asociar, para cada píxel p_{ij} de la primera imagen, la etiqueta correspondiente al sub-intervalo al que pertenece el ángulo α_{ij} correspondiente a ese píxel,

f) generar una segunda imagen a partir de la primera imagen, donde en dicha segunda imagen cada píxel lleva asociado una etiqueta,

g) realizar un proceso de suavizado a la segunda imagen para obtener zonas que comprenden píxeles con las mismas etiquetas,

h) localizar al menos un punto núcleo convexo en la segunda imagen suavizada,

i) definir una ventana centrada en el punto núcleo convexo,

j) realizar un muestreo de píxeles comprendidos en la ventana, y

k) generar el vector a partir de las etiquetas de los píxeles muestreados en el paso anterior, de forma ordenada.

La determinación del sub-intervalo al que pertenece el ángulo α del gradiente en un píxel se determina a partir del cálculo del gradiente horizontal (G_x) y del gradiente vertical (G_y) de la intensidad de la imagen (de los niveles de grises) en ese píxel.

La determinación del sub-intervalo $g_k = [\alpha_{k-1}, \alpha_k)$ que se le asocia al píxel p_{ij} comprende:

- determinar el signo de G_x
- determinar el signo de G_y
- determinar que :
- α pertenece al primer cuadrante de ángulos comprendido entre 0º y 90º, cuando G_x y G_y tienen el mismo signo, y, dentro de este primer cuadrante se distinguen dos situaciones según el rango de ángulos que abarca cada sub-intervalo a evaluar:

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el

primer cuadrante, porque tanto α_{k-1} como α_k son menores o iguales que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y < G_x \cdot \tan(\alpha_k)$$

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está parcialmente incluido en el primer cuadrante, porque α_{k-1} es menor que 90º pero α_k es mayor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y$$

* α pertenece al segundo cuadrante comprendido entre 90º y 180º, cuando G_x y G_y tienen signos distintos, y, dentro de este segundo cuadrante se distinguen dos situaciones según el rango de ángulos que abarca cada sub-intervalo a evaluar:

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el segundo cuadrante, porque tanto α_{k-1} como α_k son mayores o iguales que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_x \cdot \tan(\alpha_{k-1})| \geq |G_y| > |G_x \cdot \tan(\alpha_k)|$$

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está parcialmente incluido en el segundo cuadrante, porque α_k es mayor que 90º pero α_{k-1} es menor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_y| > |G_x \cdot \tan(\alpha_k)|$$

El proceso de suavizado calcula para cada pixel p_i de la segunda imagen preferentemente cuál de las etiquetas es la que más veces aparece en una ventana de tamaño S x S píxeles de la segunda imagen, ventana centrada en el píxel a suavizar, donde S se puede factorizar como S= s1 x s2 x ... x sn, proceso que comprende:

* comenzar con ventanas de tamaño s1 x s1 píxeles y aplicarles el suavizado a sus s1 x s1 etiquetas,
* continuar con ventanas de (s1 x s2) x (s1 x s2) pixeles y aplicar el suavizado sobre s2 x s2 etiquetas suavizadas previamente en el paso anterior,
* proceder así hasta llegar al tamaño de ventana (s1 x s2 x ... x sn) x (s1 x s2 x ... x sn) pixeles y aplicar el suavizado sobre sn x sn etiquetas

suavizadas previamente en el paso anterior.

La determinación del núcleo convexo puede llevar a cabo los siguientes pasos:

- hacer otra división del intervalo de valores posibles de ángulos, α_{ij}, en 4 sub-intervalos ($g'_1, \ldots, g'_4$) que no se solapan y cuya unión da lugar al intervalo completo de posibles valores,
- etiquetar cada g'_k sub-intervalo con una etiqueta, c'_k,
- convertir la segunda imagen suavizada, en la que a cada uno de los píxeles se le asocia una de entre G etiquetas ($c_1, \ldots, c_G$), donde-preferiblemente $G > 4$, en una imagen tetra-direccional suavizada, en la que a cada uno de los píxeles se le asocia una de entre cuatro etiquetas ($c'_1, \ldots, c'_4$) y la conversión comprende a su vez:
  - cambiar cada etiqueta c_k asociada al sub-intervalo de ángulos g_k por aquella etiqueta c'_k asociada al sub-intervalo de ángulos g'_k que verifique que la intersección $g_k \cap g'_k$ sea la mayor, y
  - determinar el núcleo convexo como el punto donde se tocan tres de cuatro regiones homogéneas de la imagen tetra-direccional suavizada, que son regiones que engloban la mayoría de las crestas con curvatura convexa.

Asimismo la invención aquí descrita también está dirigida como otro objeto de la misma a un dispositivo para generar un vector de características de una huella dactilar a partir de una imagen de la misma, dispositivo que se encuentra asociado a unos medios de captura de imagen de la huella y caracterizado porque comprende:

- un bloque de asignación de etiquetas destinado a asignar a cada píxel de la imagen una de entre G etiquetas posibles, que permite generar la segunda imagen,
- un bloque de suavizado destinado a realizar un proceso de suavizado a la segunda imagen para obtener zonas que comprenden píxeles con las mismas etiquetas,
- un bloque de determinación del núcleo convexo en la huella, destinado a localizar al menos un punto núcleo convexo en la segunda imagen suavizada,
- un bloque de ventana destinado a definir una ventana centrada en el punto núcleo convexo, realizar un muestreo de píxeles comprendidos en la ventana, obtener la etiqueta de cada píxel muestreado y generar el vector a partir de las etiquetas obtenidas de forma ordenada.

El bloque de asignación de etiquetas comprende:

- un filtro preferiblemente de Sobel 3x3 con máscaras de convolución con valores enteros y potencias de 2 para calcular los gradientes horizontales (G_x) y los gradientes verticales (G_y) de la intensidad de la imagen (de los niveles de grises) en los píxeles, y
- operadores lógicos tipo OR y AND, operadores relacionales y operaciones de valor absoluto y multiplicación por valores constantes.

El bloque de suavizado se encuentra adaptado para procesar la segunda imagen barriendo sus píxeles de uno en uno y proporcionar los píxeles de la imagen suavizada también de uno en uno, donde el bloque de suavizado define una ventana de tamaño SxS, donde S se puede factorizar como S= s1 x s2 x ... x sn, y donde el bloque de suavizado comprende una serie de registros y n sub-bloques con una arquitectura híbrida serie-paralelo de los que:

- un primer sub-bloque con tamaño de ventana s1 x s1 está adaptado para aplicar una función de suavizado en paralelo sobre s1 x s1 etiquetas de píxeles que se han ido almacenando en los correspondientes registros, sub-bloque cuya etiqueta resultante se va almacenando una tras otra en una serie de registros;
- un segundo sub-bloque con tamaño de ventana (s1 x s2) x (s1 x s2) está adaptado para aplicar una función de suavizado en paralelo sobre s2 x s2 etiquetas suavizadas previamente por el primer sub-bloque y disponibles en los correspondientes registros que almacenan la salida del primer sub-bloque, sub-bloque cuya etiqueta resultante se va almacenando una tras otra en una serie de registros;
- así hasta un sub-bloque enésimo con tamaño de ventana (s1 x s2 x ... x sn) x (s1 x s2 x ... x sn), que de nuevo aplica la función de suavizado en paralelo sobre sn x sn etiquetas suavizadas previamente por el sub-bloque anterior y disponibles en los correspondientes registros que almacenan la salida del sub-bloque anterior, sub-bloque cuya salida proporciona la etiqueta del píxel en la imagen suavizada.

El bloque de determinación del núcleo convexo comprende:

- un sub-bloque del bloque de determinación del núcleo convexo adaptado para convertir la segunda imagen suavizada en una imagen tetra-direccional,

preferentemente truncando los $\log_2 G$ bits de cada píxel a 2 bits que codifican las etiquetas de la imagen tetra-direccional, y

- un sub-bloque del bloque de determinación del núcleo convexo adaptado para localizar al menos un punto núcleo convexo.

De manera adicional y en diversas realizaciones se puede disponer de:

- Un bloque de memoria destinado a almacenar la imagen capturada de la huella.
- Un bloque de mejora de la imagen destinado a procesar la misma mejorando su calidad.
- Un bloque de fusión de información adaptado para: adquirir una clave o contraseña, aplicar una función no invertible (hash) a dicha clave o contraseña y combinar el resultado del paso anterior con el vector de características de la huella.

Dado que el dedo, la huella, no siempre se encuentran orientados de la misma manera respecto al sensor, se contempla la opción de incluir un bloque de orientación de la imagen destinado a girar o rotar la misma hasta una posición determinada en el caso de que la huella captada por los medios de captura de imagen de la huella no se encuentre en una orientación determinada, bloque que preferentemente gira por ángulos fijos para aplicar transformaciones lineales entre píxeles originales (x_i, y_i) y píxeles de las imágenes rotadas (x_f, y_f) con los parámetros de la transformación lineal fijos para cada giro, y bloque que en una realización posible puede ser programable en el número de rotaciones así como los parámetros asociados a las rotaciones.

En el caso de protección de plantilla ("template"), el dispositivo que implementa el método adicionalmente puede comprender:

- Un bloque de adquisición adaptado para adquirir un número aleatorio, clave o contraseña y aplicar un codificador de un código corrector de errores para generar un secreto,
- Un bloque con operadores XOR adaptado para calcular y almacenar unos datos de ayuda públicos a partir del vector de características de la huella y el secreto y
- Un bloque decodificador de un código corrector de errores adaptado para

recuperar un secreto a partir de una extracción del vector de características y de los datos de ayuda almacenados de la huella asociada al secreto.

Todos los bloques descritos anteriormente pueden incluirse en un dispositivo electrónico de bajo coste que, además, permite interactuar con el usuario en base a la evaluación de la calidad con unos indicadores extraídos fundamentalmente de la operación de suavizado. El dispositivo puede contar con LEDs para comunicar al usuario con un sencillo código de colores que la huella ha sido adquirida con buena calidad y/o el dedo ha sido bien colocado sobre el sensor (por ejemplo, un LED iluminado en verde indica proceso correcto y en rojo indica error). El dispositivo puede comunicar al usuario información más extensa sobre la captura (de forma visual mediante un pequeño panel LCD o de forma audible mediante un sintetizador de voz sencillo), como por ejemplo "el dedo se ha colocado demasiado abajo en el sensor". Puesto que el contexto de aplicación es en línea y el usuario está presente, esta información puede traducirse en que el usuario introduzca otra vez su dedo en el sensor (más arriba o abajo, con más o menos presión, etc., según la información que reciba del dispositivo). En tal caso, si el dispositivo incluye un panel LCD y/o un sintetizador de voz para la interacción en línea con el usuario a la hora de capturar la huella, éstos también se pueden aprovechar para comunicar hacia el exterior el/los candidato/s seleccionado/s en el proceso de reconocimiento.

La interacción en línea con el usuario proporciona muestras biométricas de calidad, que, por tanto, reducen las tasas de error que se obtienen para unas tasas bajas de penetración en la base de huellas y la tasa promedio de penetración en un escenario de búsqueda incremental ("incremental search scenario") en una aplicación de indexado. Como consecuencia, el promedio de candidatos entre los que siempre aparece el poseedor de la huella de entrada es un porcentaje pequeño de toda la base. También mediante la interacción con el usuario se mejoran las razones de falso rechazo (FRR) y falsa aceptación (FAR) para una aplicación de identificación.

DESCRIPCIÓN DE LOS DIBUJOS

Para complementar la descripción que se está realizando y con objeto de ayudar a una mejor comprensión de las características de la invención, de acuerdo con un ejemplo preferente de realización práctica de la misma, se acompaña como parte integrante de

dicha descripción, un juego de dibujos en donde con carácter ilustrativo y no limitativo, se ha representado lo siguiente:

Figura 1.- Muestra un gráfica donde se aprecia la tasa de error frente a la tasa de penetración obtenida con el método de la invención aplicada a tres bases de huellas: la FVC2000 DB2a y la FVC2002 DB1a (de las bases de huellas de "Fingerprint Verification Competition") y una base de huellas generada a partir de usuarios de un sistema experimental de identificación en línea.

Figuras 2a-2d.- Muestran unas gráficas donde se representa la razón de falsa aceptación (FAR) y de falso rechazo (FRR) frente a un umbral que mide el porcentaje de etiquetas diferentes entre los vectores de características obtenidos con el método de la invención aplicado a la base de huellas en línea: (a) Sin aplicar fusión multi-biométrica. (b) Con fusión (operador mínimo) de 3 muestras por huella en la fase de registro. (c) Con fusión (operador suma) de 2 dedos y (operador mínimo) de 3 muestras por dedo en la fase de registro. (d) Con fusión de una muestra de un dedo y contraseña.

Figuras 3a-3d - Muestran unas imágenes donde se aprecian posibles resultados de los pasos básicos del método para generar un vector de características de una huella dactilar: (a) Primera imagen de la que parte el método, una imagen en escala de grises captada por un sensor óptico de huella dactilar (tomada de la FVC2002 DB1). (b) Segunda imagen suavizada, con regiones homogéneas de etiquetas, donde cada pixel lleva asociado una de entre ocho etiquetas (cada una de las ocho etiquetas se representa por una trama diferente). (c) Ventana con información distintiva centrada en el punto núcleo convexo. (d) Píxeles muestreados de la ventana para obtener el vector de características.

Figura 4.- Muestra el diagrama de bloques funcionales de un dispositivo que extrae un vector de características distintivo de una huella dactilar. Los bloques dibujados con línea discontinua se usan o no según la aplicación.

Figura 5.- Muestra una representación del bloque de suavizado 27 x 27 que emplea tres sub-bloques: un primer sub-bloque de tamaño de ventana 3 x 3, que suaviza 3 x 3 píxeles; un segundo sub-bloque de tamaño de ventana 9 x 9, que aplica suavizado sobre 3 x 3 resultados del sub-bloque anterior; y, por último un sub-bloque 27 x 27, que aplica suavizado sobre 3 x 3 resultados del sub-bloque anterior.

Figura 6.- Muestra el diagrama de bloques funcionales de un dispositivo que implementa un método de indexación e identificación/autenticación por huella y posible clave: (1A) Fase de registro sin clave (2A) Fase de verificación sin clave (1B) Fase de registro con clave (2B) Fase de verificación con clave. Los caminos que no se marcan se usan tanto en la fase de registro como en la de verificación. Los bloques y caminos dibujados con línea discontinua se usan o no según la aplicación.

Figura 7.- Muestra el diagrama de bloques funcionales de un dispositivo que implementa un sistema de cifrado biométrico de identificación/autenticación por huella: (1) Fase de registro (2) Fase de verificación. Los caminos que no se marcan se usan tanto en la fase de registro como en la de verificación. Los bloques y caminos dibujados con línea discontinua se usan o no según la aplicación.

REALIZACIÓN PREFERENTE DE LA INVENCIÓN

A la vista de las figuras se describe a continuación un modo de realización del objeto de la invención aquí descrita

El método de la invención se ha implementado en un dispositivo electrónico también objeto de esta invención; para una realización particular del dispositivo se selecciona una implementación en una FPGA de Xilinx, una Virtex 6 XC6VLX240T-3FFG1156, que contiene 37680 slices y 416 bloques RAM de 36 Kbits. El método de la invención también se podría implementar en un circuito integrado de aplicación específica (ASIC); en tal caso, el dispositivo electrónico sería aún más pequeño, consumiría menos potencia y podría integrarse con sensores de huellas (por ejemplo, de tipo capacitivo) que emplean tecnologías CMOS.

En una realización preferida del método de identificación de huellas dactilares a partir de una extracción de vectores de características de la invención se ha implementado de la siguiente manera. Un bloque asigna a cada píxel de una imagen, correspondiente a una huella dactilar, una de entre ocho etiquetas posibles, empleando 3 bits para codificar las etiquetas, 8 bits para codificar intensidades (luminancias) de la imagen de la huella en escala de grises y 14 bits para los gradientes (obtenidos mediante filtros de Sobel 3x3), generándose una segunda

imagen.

Un bloque que aplica suavizado sobre la segunda imagen emplea un bloque de suavizado 3x3 conectado en cascada con otro bloque 9x9, conectado en cascada con otro bloque 27x27. Un bloque detecta el núcleo convexo de la huella como el punto donde intersectan tres de las cuatro regiones de la imagen tetra-direccional suavizada.

La segunda imagen suavizada se procesa para seleccionar una ventana distintiva centrada en el núcleo convexo. En este caso la ventana tiene unas dimensiones de 129x129 píxeles, muestreada de 8 en 8 píxeles, es decir, que se genera una cadena de 867 bits por huella capturada. La implementación incluye, además, bloques para calcular indicadores de calidad, una memoria para almacenar la imagen de la huella en escala de grises y un bloque que aplica una rotación sobre la imagen en escala de grises almacenada en la memoria. Todo ello ocupa el 18.31% de los slices y el 15.87% de los bloques RAM, pudiéndose alcanzar una frecuencia máxima de operación de 257.7 MHz y considerando una huella con 374 filas x 388 columnas (como las de la FVC 2002 DB1). Aplicando un procesado pixel a pixel de la huella, esto significa que el tiempo en obtener los 867 (17x17x3) bits del vector de características de una captura (sin rotaciones) puede ser de 0.56 ms (374 x 388 / 257.7 µs).

Si se tienen en cuenta 3 rotaciones de la huella para registrar a un usuario, se almacenan vectores de 2601 (3 x 17 x 17 x 3) bits por usuario. En la FPGA Virtex 6 considerada, pueden almacenarse los vectores de casi 5900 usuarios en los 416 bloques RAM de 36 Kbits.

El bloque que ordena los niveles de similitud entre el vector de entrada y los almacenados, que aplica un método de inserción y genera una lista con 50 candidatos, ocupa el 11.48% del total de slices de la FPGA Virtex 6 que estamos considerando y permite una frecuencia máxima de 207.5 MHz. Esto significa que el tiempo invertido en la fase de recuperación es bastante bajo (varias décimas de milisegundo para ordenar 5900 usuarios).

El mismo dispositivo, en este caso la misma FPGA, puede incluir todos los bloques requeridos por las fases de indexado y recuperación del método de la invención. En el caso de la Virtex 6 XC6VLX240T-3FFG1156 de Xilinx como dispositivo único, 66 bloques RAM de 36 Kbits se emplean para almacenar la huella y la ventana distintiva (considerando una huella con 374 filas x 388 columnas como las de la FVC 2002 DB1)

y se dispone de 350 bloques RAM de 36 Kbits, que permiten registrar más de 4950 usuarios (considerando 2601 bits por usuario).

El método de la invención, implementado en este ejemplo de realización FPGA, se ha evaluado con dos de las bases de huellas de "Fingerprint Verification Competition": la FVC2000 DB2a y la FVC2002 DB1a, con 800 capturas cada una. Hay que tener en cuenta que bases como las FVC están construidas con muchas capturas de mala calidad y mal adquiridas para probar la bondad de técnicas complejas de identificación e indexado. Además, también se ha considerado una base de huellas con 560 capturas, generada a partir de usuarios de un sistema experimental de identificación en línea.

La colocación del dedo es importante para extraer correctamente el vector de características. En el contexto de aplicación en línea en el que el usuario quiere identificarse, el dedo se suele colocar adecuadamente. Por ejemplo, en el experimento de registro de usuarios en línea en el que se capturaron 560 huellas, 23 capturas no permitieron extraer la ventana distintiva (4.11%). En las bases de huellas FVC2000 DB2 y FVC2002 DB1, con 800 capturas, como el contexto de aplicación es distinto, el número de capturas de las que no puede extraerse correctamente el vector de características es bastante superior: 149 en la FVC2000 DB2 (18.6%) y 104 en la FVC2002 DB1 (13%).

La calidad de la imagen capturada también es importante evaluarla, pues en una captura de 560 huellas, 10 capturas (1.79%) fueron de muy mala calidad (porque las huellas realmente estaban deterioradas). En las bases de huellas FVC2000 DB2 y FVC2002 DB1, 16 (2%) y 24 (3%) capturas son también de muy mala calidad (debido a huellas deterioradas o capturas no bien adquiridas).

La Figura 1 representa la tasa de error frente a la tasa de penetración obtenida con la técnica de la invención implementada en este ejemplo de realización y aplicada a las tres bases de huellas consideradas. Para obtener esta figura, en todas las bases se han eliminado las huellas de las que no se puede extraer su vector de características correctamente y que son de muy mala calidad (los porcentajes comentados anteriormente), puesto que con el dispositivo de la invención, que interactúa con el usuario, estos porcentajes se hubieran reducido a cero. Se ha aplicado una mejora sobre las imágenes en escala de grises (aplicando filtros complejos) y reforzado la técnica de detección del núcleo convexo. Como vectores de características registrados

en la base se han tomado los de la primera captura de cada individuo (con 5 rotaciones en la FVC2002 DB1, 3 rotaciones en la FVC 2000 DB2, y ninguna en la tercera de las bases). Como vectores de entrada se han tomado todos los del resto de capturas, sin ninguna rotación. Para calcular el nivel de similitud entre el vector de entrada y los almacenados previamente rotados (en el caso de las FVC2002 DB1 y FVC 2000 DB2) se ha seleccionado el máximo de los niveles de similitud con cada uno de los almacenados.

La tasa promedio de penetración que hay que llevar a cabo cuando no se quieren cometer errores en la recuperación del poseedor de la huella de entrada ("incremental search scenario"), en las mismas condiciones que los resultados de la Figura 1, ha sido 3.16% en la FVC2000 DB2, 2.88% en la FVC2002 DB1 y 1.62% en la tercera de las bases analizadas.

El mismo dispositivo, en este caso la misma FPGA, puede incluir una implementación de la función hash ganadora de la última competición SHA-3 del NIST, Keccak, para permitir la identificación/autenticación por el doble factor de "quién eres" y "lo que sabes". Esta función para generar 512 bits ocupa 1188 slices (3.15% del total) permitiendo una frecuencia máxima de 435.3 MHz.

La Figura 2 representa la razón de falsa aceptación (FAR) y de falso rechazo (FRR) frente a un umbral que mide el nivel de disimilitud (porcentaje de etiquetas diferentes entre los vectores de características). Los resultados corresponden a la base de huellas con capturas en línea. La Figura 2a ilustra los resultados de una identificación sin fusión biométrica. El valor donde las razones se intersectan (EER) es 5.4%. La Figura 2b ilustra los resultados de una identificación con fusión de 3 muestras capturadas por cada huella del individuo en la fase de registro y una muestra capturada en la fase de verificación. El valor donde las razones se intersectan (EER) es 2.5%. La Figura 2c ilustra los resultados de una identificación con fusión de 2 dedos por individuo, con 3 muestras capturadas por cada dedo en la fase de registro y una muestra capturada por cada dedo en la fase de verificación. El valor donde las razones se intersectan (EER) es 0.9%. La Figura 2d ilustra los resultados de una identificación con fusión de los vectores de características con una función hash que devuelve 512 bits aplicada sobre una palabra de paso o contraseña para cada individuo. El valor donde las razones se intersectan (EER) es 0%.

El mismo dispositivo, en este caso la misma FPGA, puede incluir todos los bloques

requeridos por la técnica de protección del método de la invención. En el caso de la Virtex 6 XC6VLX240T-3FFG1156 de Xilinx como dispositivo único, el bloque codificador de Reed-Solomon para n=511 y k=383 ocupa 473 slices (1.26% de los slices) y permite operar a una frecuencia máxima de 415 MHz. El bloque decodificador de Reed-Solomon para n=511 y k=383 ocupa 24.763 slices (el 65% del total de slices) trabajando con una frecuencia máxima de 78.5 MHz.

De manera más detallada el método para generar un vector de características de una huella dactilar genera el vector que es una cadena de bits de longitud fija que representa de forma compacta una huella dactilar. Para obtener ese vector, y tal y como se ha detallado anteriormente, se parte de una primera imagen como por ejemplo la captura de la huella como imagen en escala de grises (Figura 3a), se determina para cada píxel el gradiente de la intensidad de la imagen (de los niveles de grises) en ese píxel, y se determina la dirección del gradiente mediante un ángulo con respecto a un eje de referencia. El intervalo de valores posibles de ángulos se divide en G sub-intervalos (g_1,..., g_G) que no se solapan y cuya unión da lugar al intervalo completo de posibles valores, cada sub-intervalo g_k englobando ángulos desde α_{k-1} hasta α_k. A cada sub-intervalo, g_k, se le asocia una etiqueta, c_k. A cada píxel de la imagen de la huella se le asocia la etiqueta correspondiente al sub-intervalo al que pertenece el ángulo de la dirección del gradiente en ese píxel. Como resultado, se genera una segunda imagen a partir de la primera imagen de la huella, donde en dicha segunda imagen cada píxel lleva asociado una etiqueta. A continuación, se realiza un proceso de suavizado a la segunda imagen para obtener zonas que comprenden píxeles con las mismas etiquetas (Figura 3b). Se localiza al menos un punto núcleo convexo en la segunda imagen suavizada y se define una ventana centrada en el punto núcleo convexo (Figura 3c). Se realiza un muestreo de píxeles comprendidos en la ventana (Figura 3d). Las etiquetas de los píxeles muestreados, ordenadas, generan el vector de características de la huella. Si las etiquetas se codifican con bits, el vector es una cadena de bits ordenada y de longitud fija.

Si el número de etiquetas, G, considerado es pequeño (por ejemplo, cuatro etiquetas), el vector que se va a generar es poco distintivo de la huella, es decir, puede haber muchas huellas con un vector parecido, lo que se traduce en una tasa elevada de falsa aceptación, en el caso de una aplicación de identificación/autenticación. Sí puede emplearse un número como cuatro etiquetas en aplicaciones de clasificación, en las que las huellas se distribuyen en grupos pre-establecidos de acuerdo a la similitud de sus vectores de características, de forma que la huella de entrada se clasifica en uno

de esos grupos o se le asignan grados de pertenencia a varios de esos grupos.

Por el contrario, si se contempla un número de etiquetas alto (por ejemplo, dieciséis etiquetas), el vector que se va a generar es muy distintivo, pero cambia demasiado para diferentes capturas de una misma huella, lo que se traduce en una tasa elevada de falso rechazo, en el caso de una aplicación de identificación/autenticación. En una realización preferente del método de la invención para aplicaciones de identificación/autenticación, se han elegido ocho etiquetas, que es el caso que se ilustra en la Figura 3.

Los sub-intervalos deben cubrir todo el rango de ángulos que pueden tener las direcciones de los gradientes (entre 0º y 180º) de una forma más o menos espaciada. En una realización preferente de la invención para aplicaciones de identificación/autenticación con G=8, se han elegido los siguientes: g_1 = [0º, 22.5º), g_2 = [22.5º, 45º), g_3 = [45º, 67.5º), g_4 = [67.5º, 90º), g_5 = [90º, 112.5º), g_6 = [112.5º, 135º), g_7 = [135º, 157.5º) y g_8 = [157.5º, 180º), eligiendo como eje de referencia el eje longitudinal de la huella.

El tamaño de la ventana centrada en el núcleo convexo depende del sensor empleado. Por ejemplo, para las huellas de las bases FVC2002 DB1 (imágenes de 388x374 píxeles capturadas mediante un sensor óptico), las de la FVC2000 DB2 (imágenes de 256x364 pixeles capturadas mediante un sensor capacitivo de bajo coste) y las de una base experimental (imágenes de 440x300 píxeles capturadas mediante un sensor óptico), se ha probado que una ventana adecuada es de 129x129 píxeles (Figura 3c). Como representación distintiva y compacta de la huella no son necesarios todos los píxeles de la ventana, sino que se aplica un muestreo 1/n ("down-sampling"), que significa emplear, preferentemente, 1 de entre n píxeles consecutivos en cada fila de la ventana. Por ejemplo, para las huellas de las bases citadas anteriormente, se aplica un muestreo 1/8 sobre la ventana de 129x129 píxeles, que significa emplear la información de 17x17 píxeles (Figura 3d). En este ejemplo, si las ocho etiquetas se codifican con 3 bits, el vector obtenido para cada huella es una cadena de 17x17x3 = 867 bits = 108.4 Bytes. Estos vectores pueden cifrarse, por motivos de seguridad, y/o comprimirse (por ejemplo aplicando "Run-Length Encoding"), para consumir menos memoria y/o transmitirse más fácilmente.

La técnica para extraer los vectores de características de las huellas se implementa mediante el empleo de los siguientes bloques básicos (Figura 4).

- Un sensor de huella, que proporciona una imagen de la huella en escala de grises. Si el sensor no aplica mejoras sobre la imagen adquirida, se incluye un bloque de mejora de la imagen.
- Si la posición del dedo sobre el sensor de huella puede rotar, también se incluye un bloque que aplica rotación a la imagen de entrada en escala de grises.
- Un bloque que asigna a cada pixel de la imagen una de entre las G etiquetas posibles y genera una segunda imagen.
- Un bloque que aplica suavizado a la segunda imagen.
- Un bloque para detectar el núcleo convexo (o varios puntos candidatos a ser núcleo convexo) en la huella.
- Un bloque para determinar la ventana distintiva, muestrear sus pixeles y almacenar los valores de las etiquetas de esos píxeles en una cadena de bits.
- Adicionalmente, se puede incluir un bloque que evalúa la calidad de todo el proceso y permite la interacción en línea con el usuario.

El bloque que asigna a cada pixel de la imagen una de entre las G etiquetas posibles puede implementarse mediante un sencillo circuito digital. El primer paso que lleva a cabo este bloque es calcular los gradientes horizontales (G_x) y verticales (G_y) de la intensidad de la imagen (de los niveles de grises) con algún filtro adecuado para su implementación hardware (por ejemplo, mediante filtros de Sobel 3 x 3 que emplean máscaras de convolución con valores enteros y potencias de 2). Este paso es habitual en cualquier técnica de extracción de características. A continuación, en vez de calcular en cada pixel la dirección del gradiente de una forma más o menos exacta mediante una función trigonométrica (en hardware dedicado, se emplea usualmente un procesador CORDIC, "COordinate Rotation Digital Computer", para calcular el ángulo α cuya tangente es (G_y/G_x)) y luego calcular el sub-intervalo de entre los G posibles al que pertenece α, la técnica de esta invención compara entre sí los valores de los gradientes G_x y G_y y aplica operadores lógicos (OR y AND), operadores relacionales y operaciones de valor absoluto y multiplicación por valores constantes, que es mucho más eficiente desde un punto de vista hardware. En primer lugar, el bloque determina que α pertenece a un primer cuadrante comprendido entre 0º y 90º, cuando G_x y G_y tienen el mismo signo. En segundo lugar, dentro de este primer cuadrante, el bloque determina que:

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el primer cuadrante, porque tanto α_{k-1} como α_k son menores o iguales que

90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y < G_x \cdot \tan(\alpha_k)$$

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$ está parcialmente incluido en el primer cuadrante, porque α_{k-1} es menor que 90º pero α_k es mayor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y$$

El bloque determina que α pertenece a un segundo cuadrante comprendido entre 90º y 180º, cuando G_x y G_y tienen signos distintos. En tal caso, dentro de este segundo cuadrante, el bloque determina que:

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el segundo cuadrante, porque tanto α_{k-1} como α_k son mayores o iguales que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_x \cdot \tan(\alpha_{k-1})| \geq |G_y| > |G_x \cdot \tan(\alpha_k)|$$

- si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está parcialmente incluido en el segundo cuadrante, porque α_k es mayor que 90º pero α_{k-1} es menor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_y| > |G_x \cdot \tan(\alpha_k)|$$

Donde $\tan(\alpha_k)$ y $\tan(\alpha_{k-1})$ son valores constantes previamente conocidos una vez se han fijado los sub-intervalos a considerar, $g_k = [\alpha_{k-1}, \alpha_k)$.

El circuito digital que implementa estas operaciones puede emplear aritmética de punto fijo, y palabras de $\log_2 G$ bits para codificar las G etiquetas posibles correspondientes a los G sub-intervalos g_k.

El bloque de suavizado aplica un tamaño de ventana S x S, donde S depende, en general, del tipo de sensor de huella empleado. Como suavizar en paralelo S x S píxeles (para conseguir elevada velocidad de procesado) puede ser muy costoso, se puede optar por conectar en cascada varios sub-bloques de suavizado uno tras otro. Si el valor de S se puede factorizar como S= s1 x s2 x ... x sn, primero se puede usar un sub-bloque con tamaño de ventana s1 x s1, que aplica la función de suavizado

sobre s1 x s1 etiquetas de píxeles; el segundo sub-bloque con tamaño de ventana (s1 x s2) x (s1 x s2), que aplica la función de suavizado sobre s2 x s2 etiquetas suavizadas previamente por el sub-bloque anterior, y así hasta el último sub-bloque con tamaño de ventana (s1 x s2 x ... x sn) x (s1 x s2 x ... x sn), que de nuevo aplica la función de suavizado sobre sn x sn etiquetas suavizadas previamente por el sub-bloque anterior. Por ejemplo, para sensores ópticos que captan imágenes de 388x374 o 440x300 píxeles o sensores capacitivos que captan imágenes de 256x364 píxeles, se ha probado que es adecuado un bloque de suavizado 27 x 27 que se puede realizar con tres sub-bloques de suavizado conectados en cascada: un primer sub-bloque de tamaño de ventana 3 x 3, que suaviza 3 x 3 píxeles; un segundo sub-bloque de tamaño de ventana 9 x 9, que aplica suavizado sobre 3 x 3 resultados del sub-bloque anterior; y, por último un sub-bloque 27 x 27, que aplica suavizado sobre 3 x 3 resultados del sub-bloque anterior (Figura 5). En una realización preferente, la función de suavizado sj x sj considera una ventana de píxeles en torno al píxel analizado y asigna a éste el valor de la etiqueta que más veces aparece en toda la ventana. La técnica de conectar *n* sub-bloques en cascada permite reducir el hardware requerido y la latencia del proceso de suavizado porque procesar S x S valores en paralelo es mucho más costoso que procesar en paralelo sj x sj píxeles. Por ejemplo, si los píxeles de la imagen se van procesando uno a uno, el suavizado de toda la imagen se puede realizar con estos sub-bloques en cascada (y los bancos de registros necesarios) invirtiendo tantos ciclos de reloj como píxeles tenga la imagen.

El bloque que calcula el núcleo convexo (o los puntos candidatos a serlo) puede emplear técnicas ampliamente conocidas, como las basadas en el cálculo del índice de Poincaré. Una ventaja del método de la invención es que permite reforzar la detección de este punto sin apenas coste computacional, como se describe a continuación. Partiendo de la segunda imagen suavizada, se obtiene directamente una imagen tetra-direccional suavizada. Por ejemplo, si G=8 y los ocho sub-intervalos son g_1 = [0º, 22.5º), g_2 = [22.5º, 45º), g_3 = [45º, 67.5º), g_4 = [67.5º, 90º), g_5 = [90º, 112.5º), g_6 = [112.5º, 135º), g_7 = [135º, 157.5º) y g_8 = [157.5º, 180º), se pueden obtener directamente los siguientes cuatro sub-intervalos $g'_1 = g_1 \cup g_8$, $g'_2 = g_2 \cup g_3$, $g'_3 = g_4 \cup g_5$ y $g'_4 = g_6 \cup g_7$. Puesto que cada píxel de la segunda imagen suavizada se representa por 3 bits, la obtención de la imagen tetra-direccional suavizada es tan simple como truncar de 3 a 2 los bits de cada píxel, si las etiquetas se codifican adecuadamente. El núcleo convexo se puede determinar como el punto donde intersectan tres de cuatro regiones homogéneas de la imagen tetra-direccional suavizada, que son las tres regiones que engloban la mayoría de las crestas con curvatura convexa. Como la

detección correcta del núcleo convexo es importante para extraer correctamente el vector de características, se pueden considerar varios puntos como candidatos y extraer los vectores de características asociados a ellos.

La técnica de la invención permite contemplar huellas adquiridas con el dedo rotado respecto al eje longitudinal del sensor. La ventana con información representativa de la huella descrita anteriormente está caracterizada por su invariancia ante traslaciones del dedo sobre el sensor puesto que el punto central de la ventana es el punto núcleo convexo. Sin embargo, la ventana no es invariante a rotaciones. Para asegurar que diferentes capturas de la misma huella adquiridas con posibles rotaciones presenten un nivel de similitud alto con su vector de características correspondiente almacenado en la base, una solución con bajo coste en hardware es incluir un bloque que permite rotar la imagen de la huella en escala de grises. Pueden tenerse en cuenta R rotaciones previas a la obtención de la segunda imagen (por ejemplo, con R=5: -22.5º, -11.25º, 0º, 11.25º y 22.5º). Si la imagen capturada de la huella, cuyos píxeles tienen por coordenadas cartesianas (x_i, y_i), se rota un ángulo β respecto al píxel de coordenadas (x_c, y_c), las coordenadas de los píxeles pasan a ser ahora (x_f, y_f). Esta operación se puede expresar matemáticamente como sigue:

$$\begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & -sen(\beta) & 0 \\ sen(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \\ 1 \end{bmatrix}$$

Por ejemplo, si el punto de giro se toma como el punto central de la imagen de la huella (para una imagen de 374 filas y 388 columnas, los valores para x_c e y_c son 187 y 194, respectivamente), y el ángulo de rotación se elige como 11.25º, la expresión anterior se puede reducir a la siguiente:

$$\begin{bmatrix} 0.9808 & -0.1951 & 41.4407 \\ 0.1951 & 0.9808 & -32.7542 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix}$$

Por ejemplo, si cada píxel de la imagen de entrada se direcciona por sus coordenadas (x_i, y_i) en la memoria donde se almacena la captura, el bloque que aplica una rotación de 11.25º, como la anterior, direcciona ahora a ese píxel por sus coordenadas (x_f, y_f). Como las rotaciones que se contemplan son fijas, este bloque implementa una transformación lineal entre (x_i, y_i) y (x_f, y_f) con parámetros constantes, por lo que no requiere el empleo de multiplicadores. En una realización posible de este bloque, el número de rotaciones puede ser programable así como los parámetros asociados a

las rotaciones.

Para hacer la técnica de clasificación, indexado o identificación robusta a rotaciones, se contemplan más o menos ángulos según el nivel de rotación que se quiera soportar. Las rotaciones se pueden contemplar en la fase de indexado o registro y/o en la fase de recuperación o verificación y no tienen por qué coincidir en número. Así, por ejemplo, en la fase de indexado o registro, si se consideran P candidatos de núcleo convexo y V píxeles por ventana distintiva, se puede extraer una cadena de P x V x 3 bits por cada rotación, según lo comentado anteriormente. Si se contemplan R rotaciones, el índice total que se emplea para representar una captura de huella concatena las R cadenas de bits, resultando un vector característico de una longitud de R x P x V x 3 bits.

El método genera un número de identificación digital (un vector de R x P x V x 3 bits) que se le asocia al individuo poseedor de la huella, de forma que puede generarse una base con los N números asociados a los N individuos registrados (los vectores pueden cifrarse, por motivos de seguridad, y/o comprimirse, para consumir menos memoria y/o transmitirse más fácilmente). En aplicaciones multi-biométricas que empleen D dedos por individuo, se genera para cada individuo un vector de D x R x P x V x 3 bits concatenando los D números de identificación que se obtienen a partir de la huella de cada dedo. En aplicaciones multi-biométricas que empleen Z muestras del mismo dedo del individuo, se genera para cada individuo un vector de Z x R x P x V x 3 bits concatenando los Z números de identificación que se obtienen a partir de cada muestra.

En la fase de recuperación se genera una lista ordenada de individuos registrados en la base, calculando un nivel de disimilitud (o similitud) entre el vector de características de entrada y cada vector almacenado. Si los vectores han sido cifrados y/o comprimidos deben ser descifrados y/o descomprimidos para calcular el nivel de disimilitud. El nivel de disimilitud se calcula como el porcentaje de etiquetas que son diferentes entre el vector de acceso y cada vector almacenado. En el caso de multi-biometría con D dedos, la disimilitud global se obtiene a partir de la fusión (por ejemplo con el operador suma) de las disimilitudes de cada dedo. En el caso de multi-biometría con Z muestras de un mismo dedo, la disimilitud global se obtiene como la fusión (por ejemplo, con el operador mínimo) de las disimilitudes con cada muestra. La lista de individuos registrados se ordena de menor a mayor nivel de disimilitud, pudiéndose truncar la lista en un número dado de individuos o un porcentaje máximo de disimilitud.

En una aplicación de identificación, se selecciona el candidato de la lista que posea menor disimilitud (o, equivalentemente, mayor similitud). En una aplicación de autenticación, el nivel de disimilitud se compara con un umbral.

Para generar números de identificación digital diferentes para una misma huella dactilar, el número obtenido según el método de la invención puede combinarse con el resultado de una función no invertible (hash) de una clave o palabra de paso, siendo la combinación: (a) una simple concatenación o (b) una intercalación determinada de los bits o (c) una operación XOR entre los dos (para ello deben tener la misma longitud de bits), combinación que permite indexación e identificación (o autenticación) por el doble factor de "quien eres" (la huella) y "lo que sabes" (la clave o palabra de paso) y que permite revocar números de identificación comprometidos, generando otros nuevos con una nueva clave o palabra de paso.

La técnica para llevar a cabo la fase de recuperación o verificación se implementa mediante los siguientes bloques básicos (Figura 6): (a) Una memoria para almacenar los números de identificación digital o vectores de características, Bi (i=1, ..., N), de N individuos y así permitir su registro. (b) Un bloque para calcular la similitud entre el número de identificación obtenido de la huella a identificar, B', y los N números almacenados, Bi (i=1, ..., N), bloque que calcula la igualdad entre una etiqueta de entrada y otra almacenada (ambas codificadas con 3 bits, en el caso de G=8) preferentemente con 3 operadores XOR cuyas salidas sean las entradas a un operador NOR, a continuación un contador calcula el número de etiquetas iguales entre el vector de entrada y cada vector almacenado (el nivel de disimilitud es el complemento del nivel de similitud). (c) En el caso de verificación, un bloque que compara el resultado anterior con un umbral. En el caso de recuperación, un bloque que ordena de mayor a menor las similitudes con cada vector almacenado, hasta llegar a un número máximo, M, de candidatos. Existen muchos algoritmos de ordenación reportados en la literatura (por ejemplo basados en árboles binarios o n-arios, métodos de inserción, etc.), pudiéndose elegir uno u otro dependiendo de los objetivos de velocidad y recursos a emplear (los algoritmos más rápidos suelen necesitar más recursos que los más lentos y vice versa).

Los números de identificación digital, B, generados pueden protegerse mediante la técnica denominada Fuzzy Commitment, de la siguiente manera:

- en la fase de registro: se asocia a cada usuario una palabra código aleatoria, Ci (i=1, ..., N), de un código corrector de errores (a B se le añaden ceros o unos hasta

que su tamaño sea el mismo que el de las Ci) y se realiza el cálculo y almacenamiento de una función hash de Ci, hash(Ci), y de los resultados Hi=(B XOR Ci), que se denominan datos de ayuda.

- en la fase de verificación, dado un número de entrada, B', se calculan los Ci'=B' XOR Hi (si B' es similar a B, Ci' será similar a Ci), se aplica el código corrector de errores a Ci' y a su resultado se le aplica el hash. Si el resultado coincide con algún hash de los almacenados, se identificará el usuario correspondiente (si N=1, el usuario será autenticado).
- en una posible fase de comunicación, Ci ó B=Ci XOR Hi se pueden usar como secretos de los que generar claves criptográficas para cifrar o autenticar mensajes.

El código corrector de errores es preferiblemente un código Reed-Solomon, que trata las etiquetas codificadas con 3 bits como símbolos.

Para elegir la tasa de error que debe corregir el código Reed-Solomon se puede aplicar: (a) el porcentaje de etiquetas diferentes para las que se obtiene que la razón de falso rechazo coincide con la razón de falsa aceptación, si se quiere un compromiso óptimo entre ambas tasas; (b) el porcentaje de etiquetas diferentes para las que se obtiene una falsa aceptación nula, si se quiere eliminar el intrusismo; o (c) el porcentaje de etiquetas diferentes para las que se obtiene un falso rechazo nulo, si se quiere eliminar la denegación del servicio.

La técnica para llevar a cabo la protección de los vectores de características se implementa mediante los siguientes bloques básicos (Figura 7): (a) Un bloque de adquisición adaptado para adquirir un número aleatorio, clave o contraseña y aplicar un codificador de un código corrector de errores para generar una palabra código. (b) Un bloque adaptado para generar unos datos de ayuda públicos a partir del vector de características de la huella y de la palabra código, para calcular una función hash de la palabra código y almacenar los resultados en una memoria. (c) Un bloque decodificador de un código corrector de errores adaptado para recuperar un secreto a partir de una extracción del vector de características y de los datos de ayuda almacenados de la huella asociada al secreto.

REIVINDICACIONES

1. Método para generar un vector de características de una huella dactilar para su identificación a partir de una primera imagen de la misma en escala de grises que contiene crestas y valles de la huella, método caracterizado porque comprende:

a) determinar para cada píxel de la primera imagen, p_{ij} (donde ij hacen referencia a la fila y columna del píxel en la imagen), el gradiente de la intensidad de la imagen (de los niveles de grises) en ese píxel,

b) determinar la dirección del gradiente mediante un ángulo α_{ij} con respecto a un eje de referencia,

c) dividir el intervalo de valores posibles de ángulos, α_{ij}, en G sub-intervalos (g_1,..., g_G) que no se solapan y cuya unión da lugar al intervalo completo de posibles valores, englobando cada sub-intervalo g_k ángulos desde un valor α_{k-1} hasta α_k,

d) etiquetar cada g_k sub-intervalo con una etiqueta, c_k,

e) asociar, para cada píxel p_{ij} de la primera imagen, la etiqueta correspondiente al sub-intervalo al que pertenece el ángulo α_{ij} correspondiente a ese píxel,

f) generar una segunda imagen a partir de la primera imagen, donde en dicha segunda imagen cada píxel lleva asociado una etiqueta,

g) realizar un proceso de suavizado a la segunda imagen para obtener zonas que comprenden píxeles con las mismas etiquetas,

h) localizar al menos un punto núcleo convexo en la segunda imagen suavizada,

i) definir una ventana centrada en el punto núcleo convexo,

j) realizar un muestreo de píxeles comprendidos en la ventana, y

k) generar el vector a partir de las etiquetas de los píxeles muestreados en el paso anterior, de forma ordenada.

2. Método según reivindicación 1 caracterizado porque cada sub-intervalo g_k comprende ángulos comprendidos entre 0º y 180º.

3. Método según reivindicación 1 caracterizado porque la determinación del sub-intervalo al que pertenece el ángulo α del gradiente en un píxel se determina a partir del cálculo del gradiente horizontal (G_x) y del gradiente vertical (G_y) de la intensidad de la imagen (de los niveles de grises) en ese píxel.

4. Método según reivindicación 3 caracterizado porque la determinación del sub-intervalo $g_k = [\alpha_{k-1}, \alpha_k)$ que se le asocia al píxel p_{ij} comprende:

- determinar el signo de G_x
- determinar el signo de G_y
- determinar que :
- α pertenece al primer cuadrante de ángulos comprendido entre 0º y 90º, cuando G_x y G_y tienen el mismo signo, y, dentro de este primer cuadrante se distinguen dos situaciones según el rango de ángulos que abarca cada sub-intervalo a evaluar:

  - si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el primer cuadrante, porque tanto α_{k-1} como α_k son menores o iguales que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y < G_x \cdot \tan(\alpha_k)$$

  - si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$ está parcialmente incluido en el primer cuadrante, porque α_{k-1} es menor que 90º pero α_k es mayor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } G_x \cdot \tan(\alpha_{k-1}) \leq G_y$$

- α pertenece al segundo cuadrante comprendido entre 90º y 180º, cuando G_x y G_y tienen signos distintos, y, dentro de este segundo cuadrante se distinguen dos situaciones según el rango de ángulos que abarca cada sub-intervalo a evaluar:

  - si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está totalmente incluido en el segundo cuadrante, porque tanto α_{k-1} como α_k son mayores o iguales que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_x \cdot \tan(\alpha_{k-1})| \geq |G_y| > |G_x \cdot \tan(\alpha_k)|$$

  - si el sub-intervalo que se evalúa, $[\alpha_{k-1}, \alpha_k)$, está parcialmente incluido en el segundo cuadrante, porque α_k es mayor que 90º pero α_{k-1} es menor que 90º, entonces

$$\alpha \in [\alpha_{k-1}, \alpha_k) \text{ si } |G_y| > |G_x \cdot \tan(\alpha_k)|$$

5. Método según reivindicación 1 caracterizado porque el proceso de suavizado calcula para cada píxel p_{ij} de la segunda imagen preferentemente cuál de las etiquetas es la que más veces aparece en una ventana de tamaño S x S píxeles de la segunda imagen, ventana centrada en el píxel a suavizar, donde S se puede factorizar como S= s1 x s2 x ... x sn, método que comprende:

- comenzar con ventanas de tamaño s1 x s1 píxeles y aplicarles el suavizado a sus s1 x s1 etiquetas,
- continuar con ventanas de (s1 x s2) x (s1 x s2) píxeles y aplicar el suavizado sobre s2 x s2 etiquetas suavizadas previamente en el paso anterior,
- proceder así hasta llegar al tamaño de ventana (s1 x s2 x ... x sn) x (s1 x s2 x ... x sn) píxeles y aplicar el suavizado sobre sn x sn etiquetas suavizadas previamente en el paso anterior.

6 Método según reivindicación 1 caracterizado porque la determinación del núcleo convexo adicionalmente comprende:

- hacer otra división del intervalo de valores posibles de ángulos, α_{ij}, en 4 sub-intervalos ($g'_1, \ldots, g'_4$) que no se solapan y cuya unión da lugar al intervalo completo de posibles valores,
- etiquetar cada g'_k sub-intervalo con una etiqueta, c'_k,
- convertir la segunda imagen suavizada, en la que a cada uno de los píxeles se le asocia una de entre G etiquetas ($c_1, \ldots, c_G$), donde-preferiblemente G > 4, en una imagen tetra-direccional suavizada, en la que a cada uno de sus píxeles se le asocia una de entre cuatro etiquetas ($c'_1, \ldots, c'_4$) y la conversión comprende a su vez:
- cambiar cada etiqueta c_k asociada al sub-intervalo de ángulos g_k por aquella etiqueta c'_k asociada al sub-intervalo de ángulos g'_k que verifique que la intersección $g_k \cap g'_k$ sea la mayor, y
- determinar el núcleo convexo como el punto donde se tocan tres de cuatro regiones homogéneas de la imagen tetra-direccional suavizada, que son regiones que engloban la mayoría de las crestas con curvatura convexa

7. Dispositivo para generar un vector de características de una huella dactilar a

partir de una imagen de la misma según el método descrito en las reivindicaciones 1 a 6, dispositivo que se encuentra asociado a unos medios de captura de imagen de la huella y caracterizado porque comprende:

- un bloque de asignación de etiquetas destinado a asignar a cada píxel de la imagen una de entre G etiquetas posibles, que permite generar la segunda imagen,
- un bloque de suavizado destinado a realizar un proceso de suavizado a la segunda imagen para obtener zonas que comprenden píxeles con las mismas etiquetas,
- un bloque de determinación del núcleo convexo en la huella, destinado a localizar al menos un punto núcleo convexo en la segunda imagen suavizada,
- un bloque de ventana destinado a definir una ventana centrada en el punto núcleo convexo, realizar un muestreo de píxeles comprendidos en la ventana, obtener la etiqueta de cada píxel muestreado y generar el vector a partir de las etiquetas obtenidas de forma ordenada.

8. Dispositivo según reivindicación 7 caracterizado porque adicionalmente comprende un bloque de memoria destinado a almacenar la imagen capturada de la huella.

9. Dispositivo según reivindicación 7 caracterizado porque comprende un bloque de orientación de la imagen destinado a girar o rotar la misma hasta una posición determinada en el caso de que la huella captada por los medios de captura de imagen de la huella no se encuentre en una orientación determinada, bloque que preferentemente gira por ángulos fijos para aplicar transformaciones lineales entre píxeles originales (x_i, y_i) y píxeles de las imágenes rotadas (x_f, y_f) con los parámetros de la transformación lineal fijos para cada giro, y bloque que en una realización posible puede ser programable en el número de rotaciones así como los parámetros asociados a las rotaciones.

10. Dispositivo según reivindicación 7 caracterizado porque el bloque de asignación de etiquetas comprende:

- un filtro preferiblemente de Sobel 3x3 con máscaras de convolución con valores enteros y potencias de 2 para calcular los gradientes horizontales (G_x) y los gradientes verticales (G_y) de las crestas de la huella, y

- operadores lógicos tipo OR y AND, operadores relacionales y operaciones de valor absoluto y multiplicación por valores constantes.

11. Dispositivo según reivindicación 7 caracterizado porque el bloque de suavizado se encuentra adaptado para procesar la segunda imagen barriendo sus píxeles de uno en uno y proporcionar los píxeles de la imagen suavizada también de uno en uno, donde el bloque de suavizado define una ventana de tamaño SxS, donde S se puede factorizar como S= s1 x s2 x ... x sn, y donde el bloque de suavizado comprende una serie de registros y n sub-bloques con una arquitectura híbrida serie-paralelo de los que:

- un primer sub-bloque con tamaño de ventana s1 x s1 está adaptado para aplicar una función de suavizado en paralelo sobre s1 x s1 etiquetas de píxeles que se han ido almacenando en los correspondientes registros, sub-bloque cuya etiqueta resultante se va almacenando una tras otra en una serie de registros;
- un segundo sub-bloque con tamaño de ventana (s1 x s2) x (s1 x s2) está adaptado para aplicar una función de suavizado en paralelo sobre s2 x s2 etiquetas suavizadas previamente por el primer sub-bloque y disponibles en los correspondientes registros que almacenan la salida del primer sub-bloque, sub-bloque cuya etiqueta resultante se va almacenando una tras otra en una serie de registros;
- así hasta un sub-bloque enésimo con tamaño de ventana (s1 x s2 x ... x sn) x (s1 x s2 x ... x sn), que de nuevo aplica la función de suavizado en paralelo sobre sn x sn etiquetas suavizadas previamente por el sub-bloque anterior y disponibles en los correspondientes registros que almacenan la salida del sub-bloque anterior, sub-bloque cuya salida proporciona la etiqueta del píxel en la imagen suavizada.

12. Dispositivo según reivindicación 7 caracterizado porque adicionalmente comprende un bloque de fusión de información adaptado para:

- adquirir una clave o contraseña,
- aplicar una función no invertible (hash) a dicha clave o contraseña y
- combinar el resultado del paso anterior con el vector de características de la huella.

13. Dispositivo según reivindicación 7 caracterizado porque adicionalmente comprende los siguientes bloques:

- un bloque de adquisición adaptado para adquirir un número aleatorio, clave o contraseña y aplicar un codificador de un código corrector de errores para generar un secreto,
- un bloque con operadores XOR adaptado para calcular y almacenar unos datos de ayuda públicos a partir del vector de características de la huella y el secreto y
- un bloque decodificador de un código corrector de errores adaptado para recuperar un secreto a partir de una extracción del vector de características y de los datos de ayuda almacenados de la huella asociada al secreto.

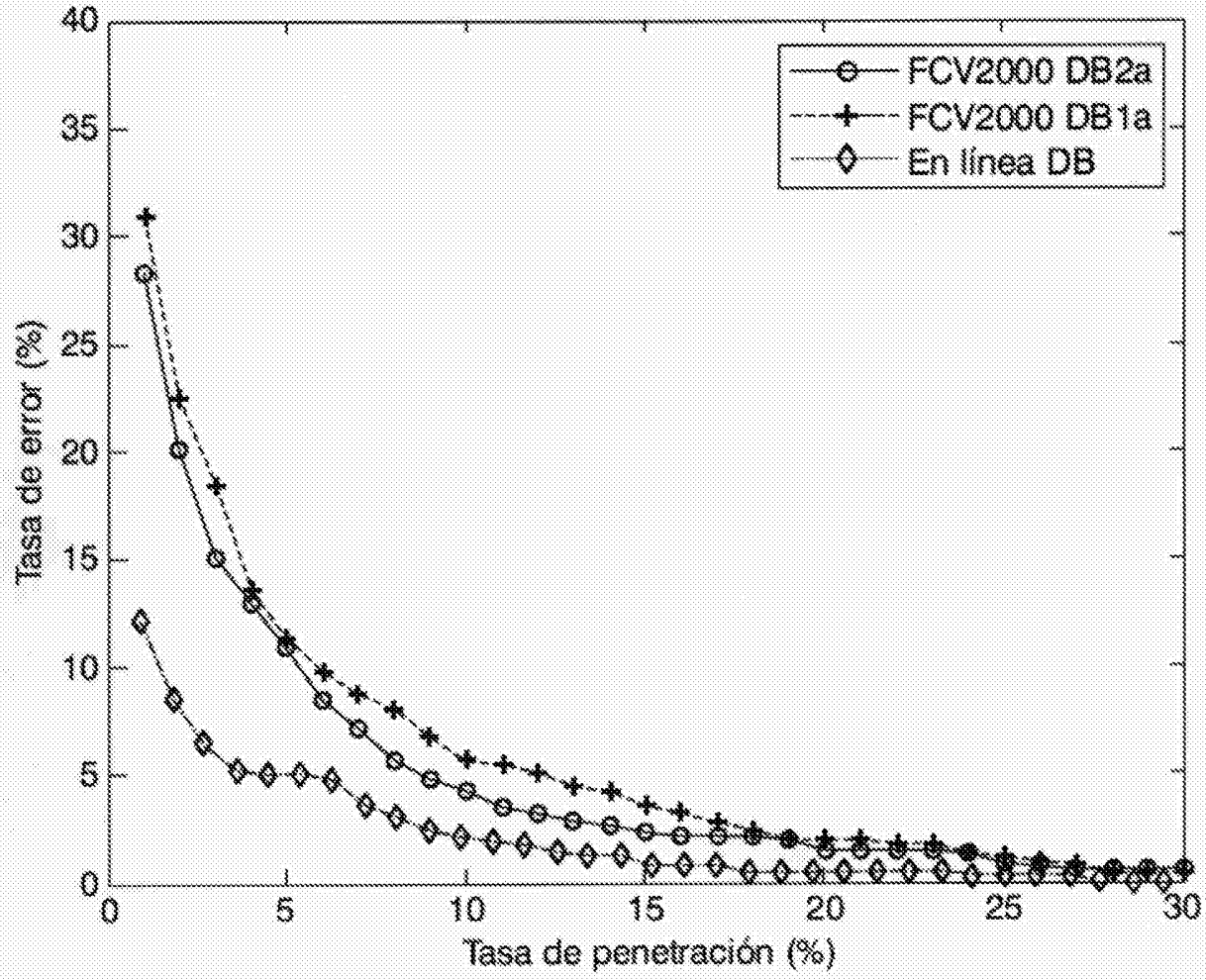
FCV2000 DB2a
FCV2000 DB1a
En línea DB
Tasa de error (%)
Tasa de penetración (%)
0
5
10
15
20
25
30
35
40

FIG. 1

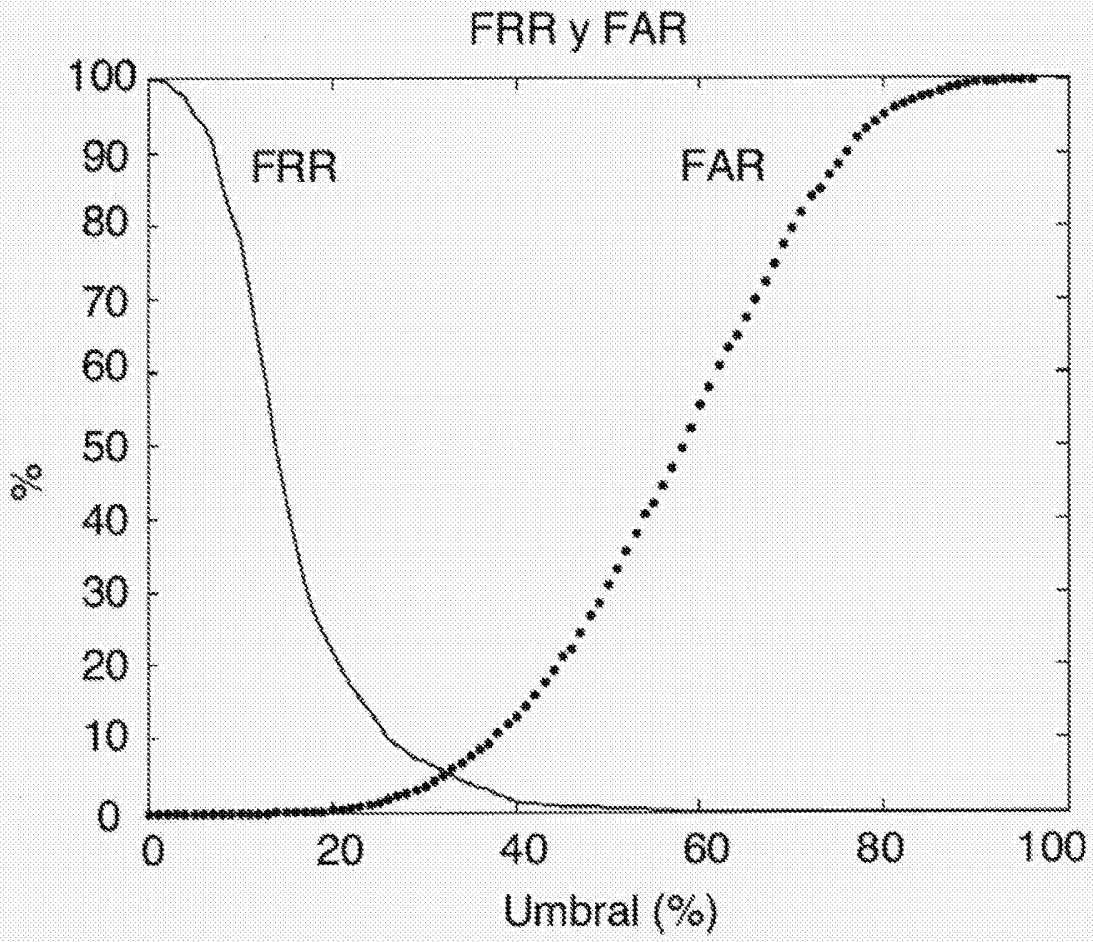
FRR y FAR
100
90
80
70
60
50
40
30
20
10
0
%
FRR
FAR
0
20
40
60
80
100
Umbral (%)

FIG. 2A

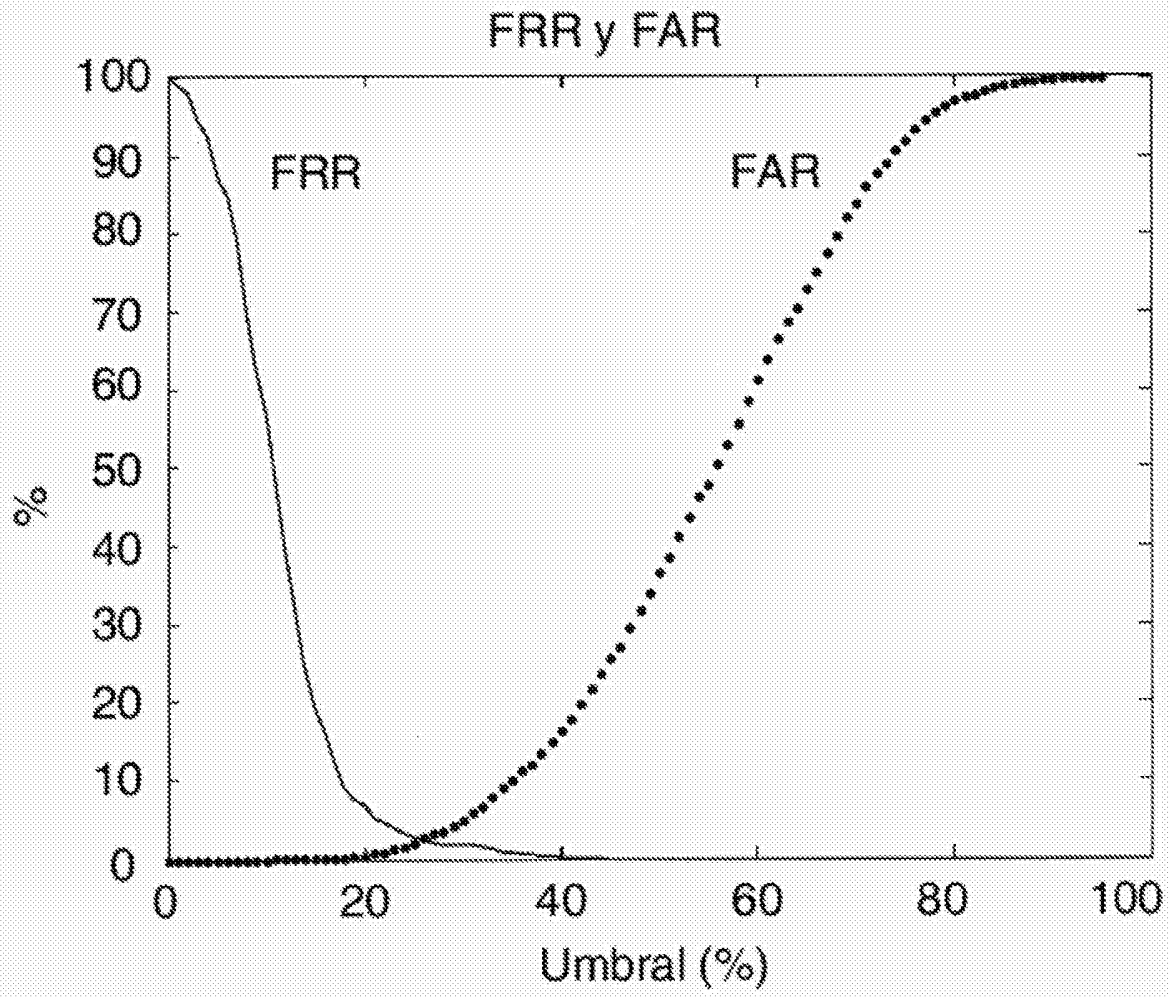
FRR y FAR
100
90
80
70
60
50
40
30
20
10
0
%
FRR
FAR
0
20
40
60
80
100
Umbral (%)

FIG. 2B

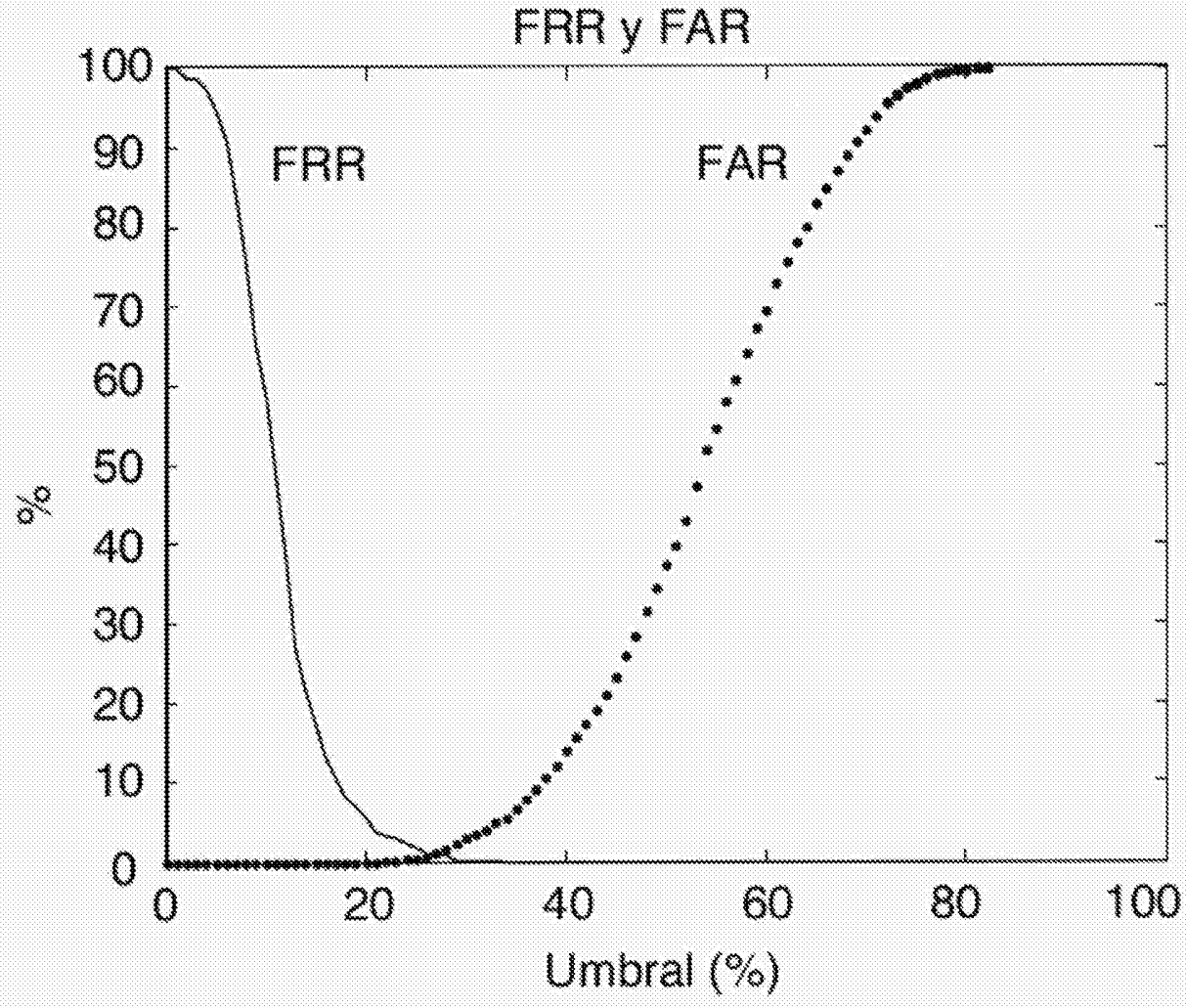
FRR y FAR
100
90
80
70
60
50
40
30
20
10
0
%
FRR
FAR
0
20
40
60
80
100
Umbral (%)

FIG. 2C

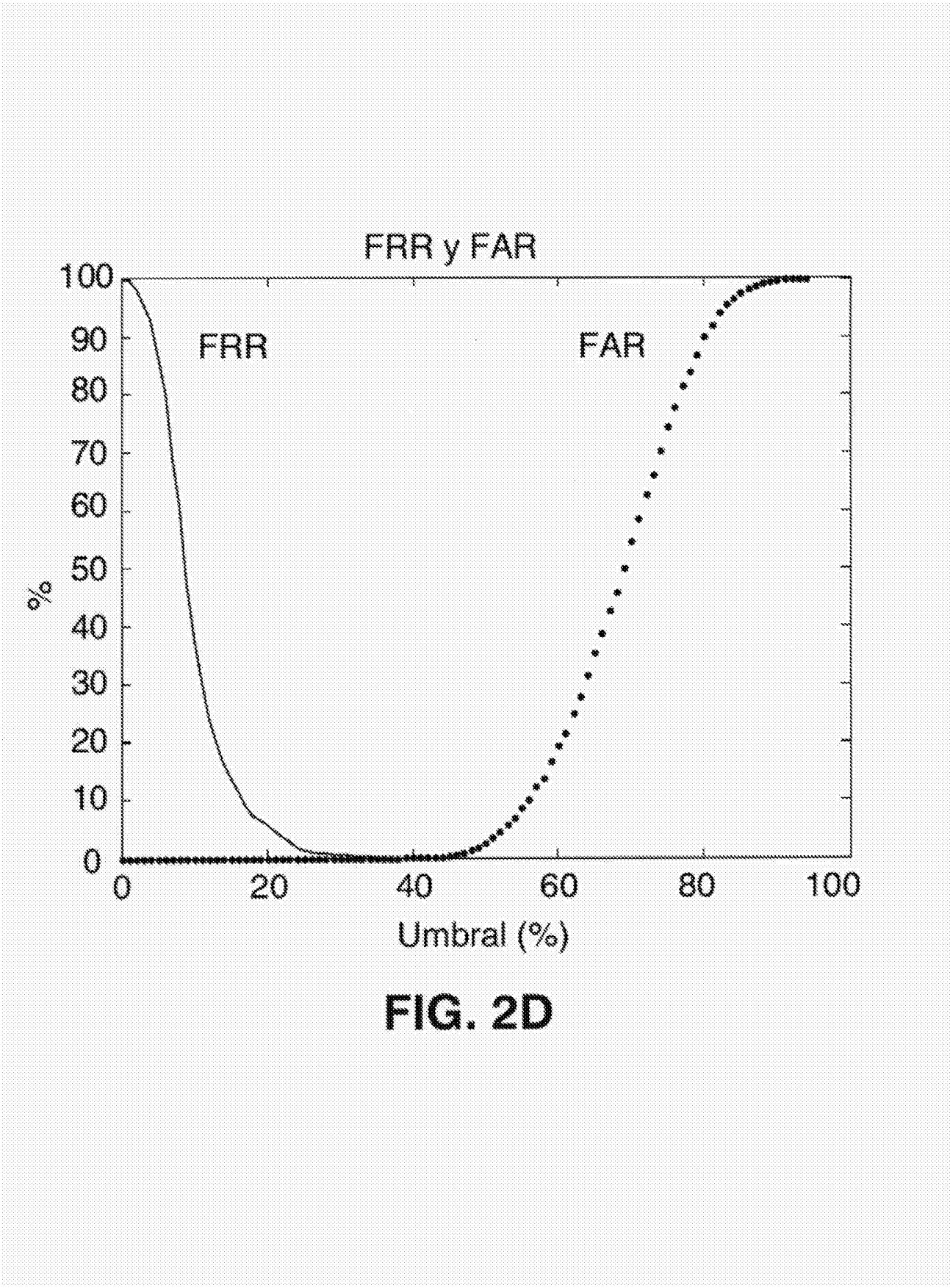
FRR y FAR
100
90
80
70
60
50
40
30
20
10
0
%
FRR
FAR
0
20
40
60
80
100
Umbral (%)

FIG. 2D

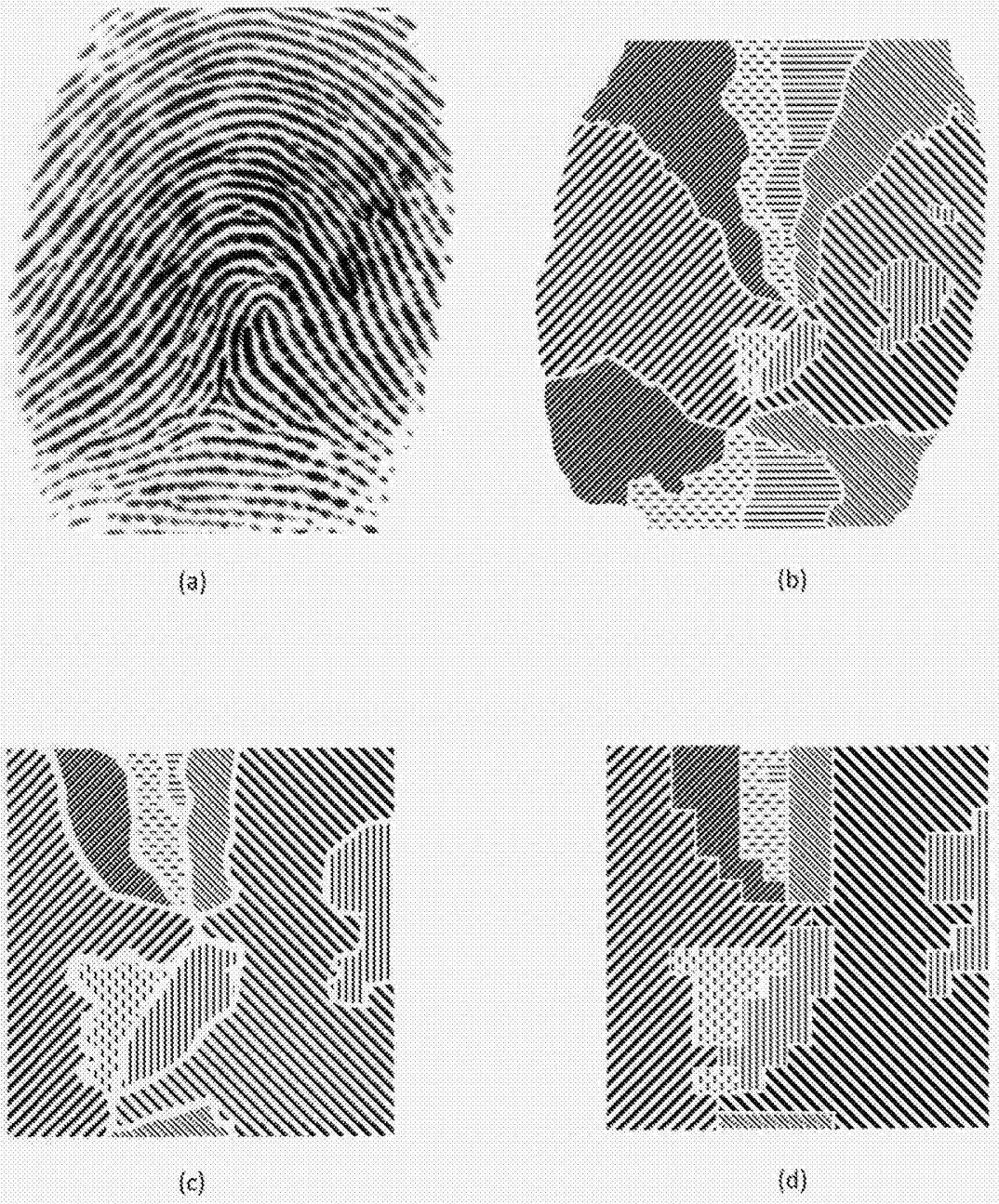

FIG. 3

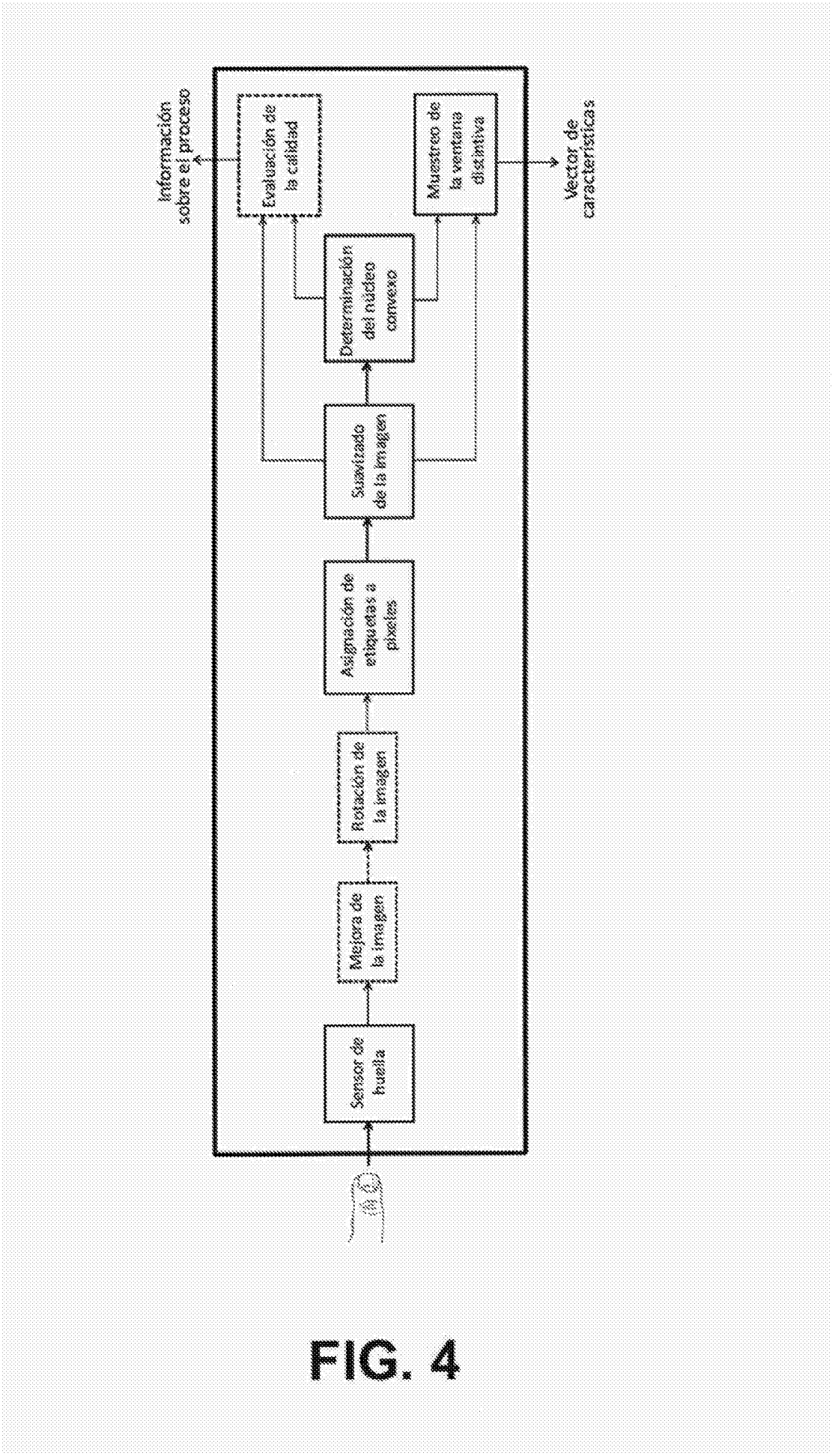
Información sobre el proceso
Evaluación de la calidad
Muestreo de la ventana distintiva
Vector de características
Determinación del núcleo convexo
Suavizado de la imagen
Asignación de etiquetas a píxeles
Rotación de la imagen
Mejora de la imagen
Sensor de huella

FIG. 4

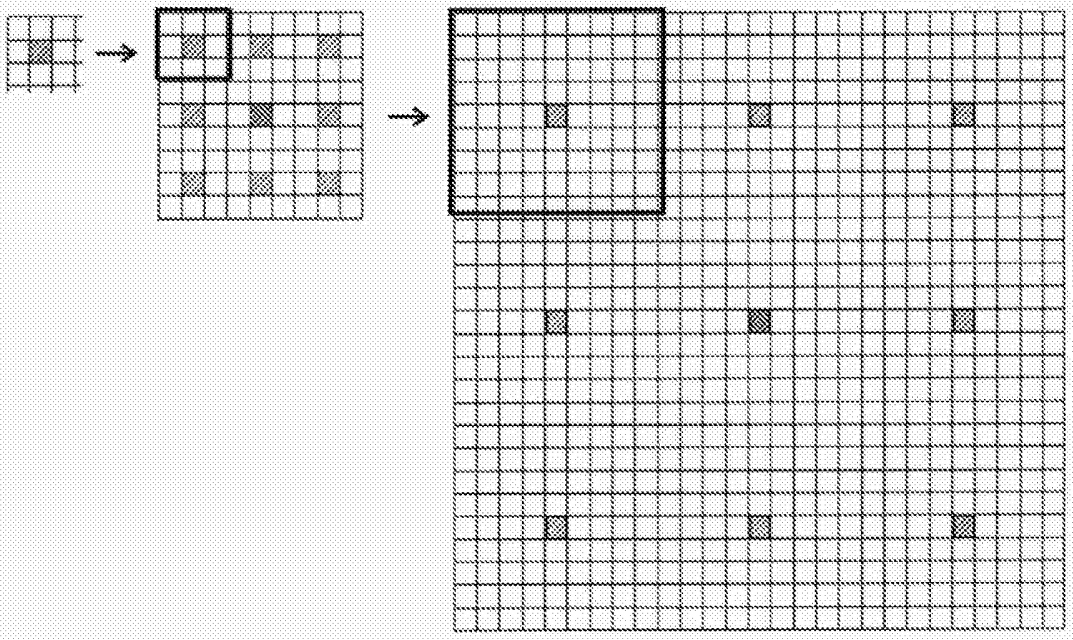

FIG. 5

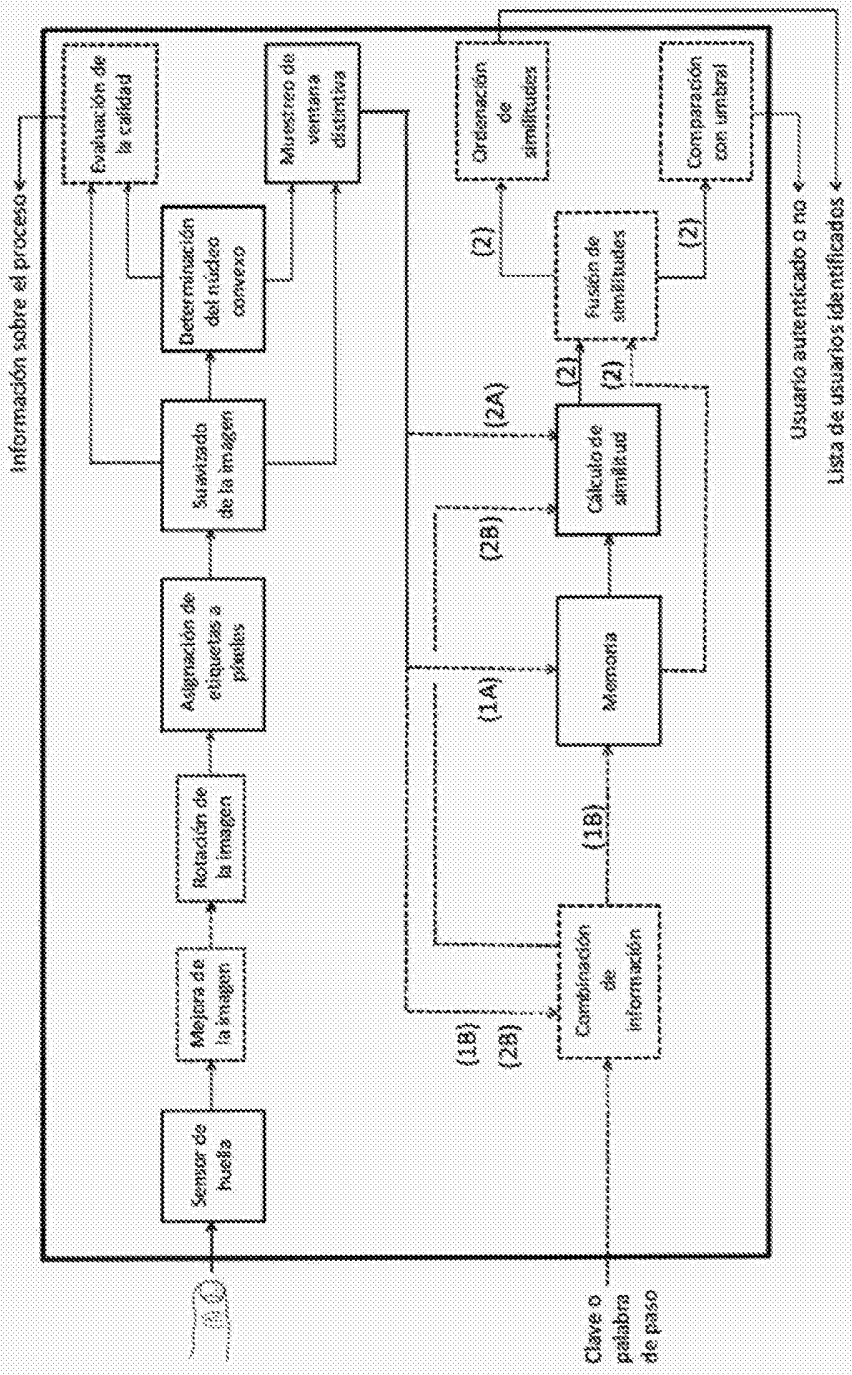
Información sobre el proceso
Evaluación de la calidad
Determinación del núcleo convexo
Suavizado de la imagen
Asignación de etiquetas a píxeles
Rotación de la imagen
Mejora de la imagen
Sensor de huella
Muestreo de ventana distintiva
Ordenación de similitudes
Fusión de similitudes
Cálculo de similitud
Memoria
Combinación de información
Comparación con umbral
Usuario autenticado o no
Lista de usuarios identificados
Clave o palabra de paso
(1A)
(1B)
(2)
(2A)
(2B)

FIG. 6

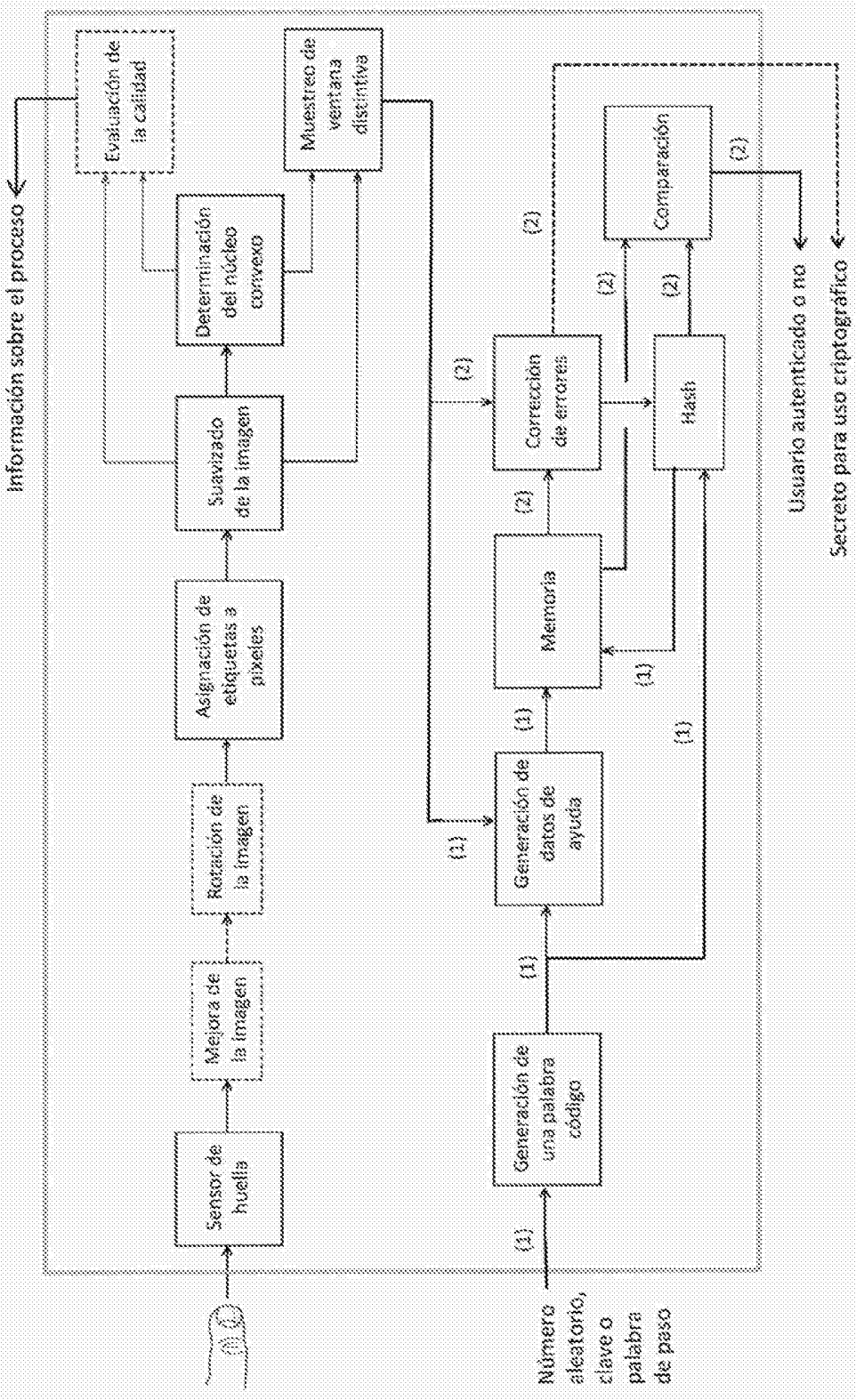
Información sobre el proceso
Evaluación de la calidad
Determinación del núcleo convexo
Muestreo de ventana distintiva
Suavizado de la imagen
Asignación de etiquetas a píxeles
Rotación de la imagen
Mejora de la imagen
Sensor de huella
Corrección de errores
Comparación
Hash
Memoria
Generación de datos de ayuda
Generación de una palabra código
{1}
{2}
Número aleatorio, clave o palabra de paso
Usuario autenticado o no
Secreto para uso criptográfico

FIG. 7