



**INAOE**

# Algoritmos basados en tríos de minucias para la verificación e identificación de huellas dactilares

por

**Miguel Angel Medina Pérez**

Tesis sometida como requerimiento parcial para  
obtener el grado de

**Doctor en Ciencias Computacionales**

en el

**Instituto Nacional de Astrofísica,  
Óptica y Electrónica**

Octubre 2014

Tonantzintla, Puebla

Supervisada por:

**Dr. Leopoldo Altamirano Robles y Dr.  
Milton García Borroto**

©INAOE 2014

El autor(a) otorga al INAOE el permiso de  
reproducir y distribuir copias en su totalidad o  
en partes de esta tesis

*L. Altamirano R*





# Resumen

El reconocimiento automatizado de huellas dactilares y palmares es un área activa de investigación en la actualidad. Las huellas dactilares (palmares) se forman a partir del contacto de los dedos (las palmas) de las manos con las superficies de los objetos.

Dentro de los sistemas automatizados de reconocimientos de huellas, los algoritmos de comparación de huellas constituyen un componente de vital importancia. Un algoritmo de comparación de huellas es una secuencia finita de pasos lógicos, que compara las huellas descritas por rasgos (o abstracciones) y retorna un valor de similitud.

Uno de los enfoques seguidos en la literatura para representar y comparar las huellas es el uso de tríos de minucias. Las minucias son los puntos donde la continuidad de las crestas papilares (glándulas de secreción de sudor en dedos, palmas de la mano y pies) se rompe.

En esta tesis se hace un análisis de los algoritmos basados en tríos de minucias y se identifican las limitaciones que degradan su eficacia. Como resultado de la investigación, se propone una nueva forma de representación y función de comparación de tríos de minucias que supera las limitaciones identificadas en la literatura. Basado en la nueva forma de representación y comparación de tríos de minucias, se construye un algoritmo de comparación para la verificación de huellas dactilares; otro para la verificación de huellas palmares y un tercero para la identificación de huellas latentes (dactilares y palmares). El objetivo de las aplicaciones de verificación de huellas consiste esencialmente en determinar dadas dos impresiones (huellas formadas voluntariamente en condiciones controladas), si estas proceden del mismo dedo. Por otra parte, el objetivo de las aplicaciones de identificación de huellas latentes (huella formadas involuntariamente en condiciones no controladas) consiste básicamente en, dada una huella latente, buscar todas las huellas procedentes del mismo dedo (o palma) presentes en una base de datos.

Los algoritmos propuestos en esta investigación usan los tríos de minucias para hallar múltiples conjuntos de pares de minucias coincidentes. Los conjuntos obtenidos son mezclados de acuerdo a una novedosa heurística para obtener la mayor cantidad de coincidencias entre las huellas que se comparan.

Los algoritmos propuestos se evalúan en bases de datos públicas usadas en la literatura actual así como en bases de datos no públicas de la competencia internacional en curso FVC-onGoing. Los resultados experimentales muestran que los algoritmos alcanzan valores de eficacia altos con respecto al estado del arte.



# Abstract

The automated fingerprint recognition is currently an active research area. Fingerprints (palmprints) are formed from the contact of the fingers (palms) of the hands with the surfaces of objects.

The fingerprint matching algorithms are important components of automated fingerprint recognition systems. A fingerprint (palmprint) matching algorithm is a finite sequence of logical steps, which compares the fingerprints (palmprints) described by features (or abstractions) and returns a similarity value.

An approach followed in the literature for fingerprint matching and representation is the use of minutiae triplets. Minutiae are the points where the continuity of ridges (sweat glands in fingers, hand plants, and soles) break.

In this thesis, the fingerprint matching algorithms based on minutiae triplets are analyzed and the drawbacks that affect their accuracy are identified. As a result of this research, it is proposed a feature representation and a matching function for minutiae triplets that overcome the identified limitations in the literature. Based on the new representation and matching function for minutiae triplets, a matching algorithm is created for fingerprint verification, another matching algorithm is created for palmprint verification, and a third matching algorithm is created for latent fingerprint and palmprint identification. The aim of fingerprint (palmprint) verification applications is essentially determining whether two impressions (prints formed voluntarily in controlled conditions) comes from the same finger (palm) or not. In the other hand, the aim of latent print identification applications is essentially, given a latent print (prints formed involuntarily in uncontrolled conditions), searching all the prints in a database that comes from the same finger (or palm).

The proposed algorithms use minutiae triplets in order to find multiple sets of matching minutiae. These sets are merged according to a new heuristic in order to obtain the highest amount of matching minutiae between the compared prints.

The proposed algorithms are evaluated in public databases which are used in the current literature; they are also tested in the non-publicly databases of the active international competition FVC-onGoing. The experimental results show that the new algorithms achieve high values of accuracy compared with the state of the art results.



# Agradecimientos

Mis más sinceros agradecimientos a mis asesores Dr. Leopoldo Altamirano Robles y Dr. Milton García Borroto por su certera guía y apoyo para la realización de esta investigación. Sus ejemplos han sido motivación para dar lo mejor de mí y esforzarme al máximo en mi trabajo.

Le agradezco al Dr. José Ruíz Shulcloper y a las personas del Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), Habana, Cuba; quienes me motivaron y apoyaron en mis primeros pasos por el Reconocimiento de Huellas.

Un agradecimiento especial a la Universidad de Ciego de Ávila, Cuba; al Centro de Bioplantas y al Laboratorio Provincial de Criminalística de Ciego de Ávila donde siempre recibí apoyo y donde aprendí mucho. Especial gracias a todos y cada uno de los amigos que hice en estos lugares que no menciono porque no cabrían en esta página pero ellos saben quienes son.

Agradezco al comité de revisión por sus útiles comentarios y sugerencias. Gracias al Dr. Luis Villaseñor Pineda, Dr. Manuel Montes y Gómez, Dr. José Enrique Muñoz de Cote Flores Luna, Dr. Hugo Jair Escalante Balderas y Dr. Aythami Morales Moreno.

Gracias a los muchachos dirigidos por el Dr. Miguel Ángel Ferrer Ballester en La Universidad de Las Palmas de Gran Canaria, España; por sumarme a su grupo como uno más de los suyos.

Un agradecimiento muy especial a mis padres Amparo Pérez Mesa y Bruno Rufino Medina Romero quienes nunca renunciaron al empeño de convertirme en un profesional. Gracias a mi hermanita Miviana del Carmen Medina Pérez por apoyarme en las buenas y malas y por sentirse más orgullosa que yo de mis logros.

A mi esposa María Esther Meléndez Abrego por tanto amor, apoyo y por tolerarme sin rendirse.

A mis familias cubanas y mexicanas: los Medina, los Romero, los Abrego, los Gonzáles, los Rodríguez, los Carro, los Toledo y muchos más que siempre llevaré en mi corazón. Gracias también a esos amigos que la vida nos da la oportunidad de hacer por donde quiera que uno pasa y que, por suerte, son muchos.

Finalmente, mis agradecimientos al Instituto Nacional de Astrofísica, Óptica y Electrónica y al Consejo Nacional de Ciencia y Tecnología de México (No. Becario: 256880) por brindar la oportunidad a muchos como yo de convertirnos en doctores.





# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Descripción del problema . . . . .	8
1.2. Objetivo general . . . . .	13
1.3. Objetivos particulares . . . . .	13
1.4. Aportaciones . . . . .	14
1.5. Organización de la tesis . . . . .	15
<b>2. Trabajo relacionado</b>	<b>17</b>
2.1. Clasificación de los algoritmos según el dominio del problema . . . . .	18
2.1.1. Algoritmos para la verificación de huellas . . . . .	18
2.1.2. Algoritmos para la identificación de huellas latentes . . . . .	18
2.2. Clasificación de los algoritmos según los rasgos que usan . . . . .	20
2.2.1. Algoritmos basados en las imágenes . . . . .	21
2.2.2. Algoritmos basados en características globales . . . . .	21
2.2.3. Algoritmos basados en características locales . . . . .	21
2.2.4. Algoritmos basados en micro características . . . . .	31
2.2.5. Algoritmos híbridos . . . . .	32
2.3. Síntesis y conclusiones . . . . .	32
<b>3. Algoritmos de comparación de huellas basados en tríos de minucias</b>	<b>35</b>
3.1. Representación de los rasgos . . . . .	35
3.2. Función de similitud de m-triplets . . . . .	38
3.3. Cálculo de los tríos de minucias . . . . .	43
3.4. Algoritmos de comparación basados en múltiples clusters de coincidencias . .	43
3.5. Síntesis y conclusiones. . . . .	48
<b>4. Resultados experimentales en la verificación de huellas</b>	<b>53</b>
4.1. Protocolos de evaluación . . . . .	53
4.2. Resultados en las competencias FVC2002, FVC2004 y FVC2006 . . . . .	57
4.3. Resultados en la competencia FVC-onGoing . . . . .	62

4.3.1. Resultados en la verificación de huellas dactilares . . . . .	63
4.3.2. Resultados en la verificación de huellas palmares . . . . .	64
4.4. Síntesis y conclusiones . . . . .	64
<b>5. Resultados experimentales en la identificación de huellas latentes</b>	<b>69</b>
5.1. Protocolos de evaluación . . . . .	69
5.2. Resultados en la base de datos NIST SD27 . . . . .	71
5.3. Resultados en la base de datos LPIDB v1.0 . . . . .	75
5.4. Síntesis y conclusiones . . . . .	76
<b>6. Conclusiones y trabajos futuros</b>	<b>77</b>
6.1. Publicaciones . . . . .	78
6.2. Patentes . . . . .	79
<b>Bibliografía</b>	<b>79</b>

# Índice de figuras

1.1.	Ejemplo de huella dactilar. . . . .	5
1.2.	Diagrama de bloques de los sistemas de comparación de huellas. . . . .	6
1.3.	Ejemplo de huellas latentes e impresiones. . . . .	7
1.4.	Ejemplos de los ángulos de las minucias. . . . .	10
1.5.	Ejemplo de minucias coincidentes en huellas de un mismo dedo. . . . .	10
1.6.	Tríos de minucias similares descartados erróneamente por algunos algoritmos.	12
1.7.	Ejemplos de tríos de minucias no coincidentes. . . . .	12
2.1.	La primera forma de representación de tríos de minucias en la literatura. . .	24
2.2.	Falsas coincidencias encontradas por el algoritmo de Jiang y Yau . . . . .	25
3.1.	Los componentes de la nueva forma de representación de tríos de minucias. .	36
3.2.	Ejemplo de múltiples clusters de coincidencias . . . . .	45
3.3.	Esquema general de los algoritmos propuestos. . . . .	46
3.4.	Efectos de incumplir algunos parámetros de calidad . . . . .	49
4.1.	Ejemplos de huellas de las bases de datos para la verificación . . . . .	58
4.2.	Curvas ROC en la competencia internacional FVC2002 . . . . .	59
4.3.	Curvas ROC en la competencia internacional FVC2004 . . . . .	60
4.4.	Curvas ROC en la competencia internacional FVC2006 . . . . .	61
4.5.	Diagramas de Diferencias Críticas . . . . .	63
5.1.	Curvas CMC en NIST SD27 y 1133516 impresiones . . . . .	73
5.2.	Curvas CMC en NIST SD27 por categorías y 1133516 impresiones . . . . .	74
5.3.	Curvas CMC en la base de datos LPIDB v1.0 + THUPALMLAB . . . . .	76



# Índice de cuadros

2.1.	Resultados previos en la competencia FVC-onGoing. . . . .	19
2.2.	Resultados previos en base de datos NIST SD27 . . . . .	20
2.3.	Limitaciones de los algoritmos basados en tríos de minucias . . . . .	33
4.1.	Tecnologías usadas en las bases de datos para verificación . . . . .	57
4.2.	Resultados de M3gl2 en la competencia FVC2002 . . . . .	62
4.3.	Resultados de M3gl2 en las competencias FVC2004 y FVC2006 . . . . .	65
4.4.	Resultados de M3gl2 en FMISO-STD-1.0 de FVC-onGoing. . . . .	66
4.5.	Resultados de M3gl2 en FMISO-HARD-1.0 de FVC-onGoing. . . . .	67
4.6.	Resultados de M3gl2.5 en PV-FULL-1.0 de FVC-onGoing. . . . .	67
4.7.	Resultados de M3gl2.5 en PV-PARTIAL-1.0 de FVC-onGoing. . . . .	68
5.1.	Comparación de eficacia (ranking 1) usando la base de datos NIST SD27 . . .	71



# Glosario de términos

**Algoritmo de comparación de huellas:** Secuencia finita de pasos lógicos, que compara las huellas descritas en término de rasgos (o abstracciones) y retorna un valor de similitud.

**Algoritmo para verificación de huellas:** Algoritmo de comparación de huellas usado para determinar si dos huellas tomadas de manera voluntaria en condiciones controladas proceden del mismo dedo.

**Algoritmo para identificación de huellas latentes:** Algoritmo de comparación de huellas usado para, dada una huella latente, buscar todas las huellas procedentes del mismo dedo presentes en una base de datos.

**Alineación de huellas:** Acción de rotar y trasladar una huella para superponerla sobre otra.

**Comparación falsa coincidente:** Una comparación de huellas se considera falsa coincidente cuando se asume por el sistema, o la aplicación, que las huellas comparadas son del mismo dedo siendo en realidad huellas de dedos diferentes.

**Comparación falsa no coincidente:** Una comparación de huellas se considera falsa no coincidente cuando se asume por el sistema, o la aplicación, que las huellas comparadas son de dedos diferentes siendo en realidad huellas del mismo dedo.

**Cresta papilar:** Glándulas de secreción de sudor en dedos, palmas de la mano y pies.

**CTYZ:** Algoritmo propuesto en (Chen, Tian, Yang y Zhang 2006).

**Curva CMC:** Curva de tasa de huellas latentes identificadas correctamente contra ranking de identificación al ordenar las comparaciones descendientemente según el valor de similitud (Jain y Feng 2011).

**Curva ROC:** Curva de tasa de comparaciones falsas no coincidentes contra tasa de comparaciones falsas coincidentes (Fawcett 2006).

**Delta de huella:** Matemáticamente se puede calcular como el baricentro del triángulo cuyos vértices son los puntos vecinos de crestas diferentes donde la primera derivada de cada cresta (modelada como curvas) cambia de signo.

**Escáner dactilar:** Dispositivo electrónico usado para obtener y digitalizar huellas dactilares.

**FFCS:** Algoritmo propuesto en (Feng et al. 2006).

**Huella dactilar:** Marcas producidas por las crestas papilares y surcos interpapilares al entrar en contacto un dedo de las manos con las superficies de los objetos.

**Huella latente:** Marcas producidas por las crestas papilares y surcos interpapilares al entrar en contacto un dedo, palma de la mano o pie con las superficies de los objetos de manera involuntaria y en condiciones no controladas; típicamente aparece en superficies

irregulares y se requiere del uso de productos químicos para su obtención.

**Huella palmar:** Marcas producidas por las crestas papilares y surcos interpapilares al entrar en contacto una palma de las manos con las superficies de los objetos.

**HK:** Algoritmo propuesto en (Hoyle y Hsiao 2011).

**Impresión:** Marcas producidas por las crestas papilares y surcos interpapilares al entrar en contacto un dedo, palma de la mano o pie con las superficies de los objetos de manera voluntaria en condiciones controladas; típicamente formada sobre un escáner (dactilar o palmar) o un modelo dactilográfico o dígito-palmar.

**JY:** Algoritmo propuesto en (Jiang y Yau 2000).

**JG1:** Algoritmo propuesto en (Jea y Govindaraju 2005).

**JG2:** Algoritmo propuesto en (Jea 2005).

**KV:** Algoritmo propuesto en (Kovács-Vajna 2000).

**Minucia:** Puntos donde la continuidad de una cresta papilar se rompe.

**Model dactilográfico:** Formato oficial de papel, cartón o cartulina usada por entidades policiales para obtener y archivar las huellas dactilares.

**Model dígito-palmar:** Formato oficial de papel, cartón o cartulina usada por entidades policiales para obtener y archivar las huellas dactilares y palmares.

**Núcleo de huella:** Punto más al norte de la cresta más interna (Maltoni et al. 2009).

**PN:** Algoritmo propuesto en (Parziale y Niel 2004).

**Puntos focales:** Núcleos y deltas.

**Puntos vecinos de crestas diferentes:** Dos puntos de diferentes crestas que al unirlos por una recta, dicha recta no corta ni es tangente a una tercera cresta.

**Rasgos de huella:** Abstracción de una huella.

**RUR:** Algoritmo propuesto en (Reisman et al. 2005).

**Sistema embebido en arquitectura ligera de hardware:** hardwares del tipo *System-on-Device* (SoD) y *System-on-Card* (SoC) como lo son las tarjetas inteligentes (Maltoni et al. 2009).

**Surcos interpapilares:** Espacio que separa a dos crestas papilares.

**TB:** Algoritmo propuesto en (Tan y Bhanu 2006).

**TPS:** Thin Plate Spline (Bazen y Gerez 2003).

**XCF:** Algoritmo propuesto en (Xu et al. 2007).

**ZGZ:** Algoritmo propuesto en (Zheng et al. 2009).



# 1

## Introducción

Las huellas dactilares son las marcas producidas por las *crestas papilares* (*crestas* de aquí en adelante) y *surcos interpapilares* (*surcos* de aquí en adelante) de los dedos al entrar en contacto con la superficie de objetos (Figura 1.2). Las crestas y surcos se forman en el sexto mes de vida intrauterina del ser humano y no desaparecen hasta tiempo después de la muerte de la persona (perennes), no existen dos personas con crestas y surcos iguales (diversos), y las crestas y surcos no se modifican de manera natural durante la vida de una persona (inmutables). Estas características hacen que las huellas dactilares sean un elemento importante en la identidad de personas.

Debido a la complejidad y la cantidad de huellas que existen, es necesario crear sistemas

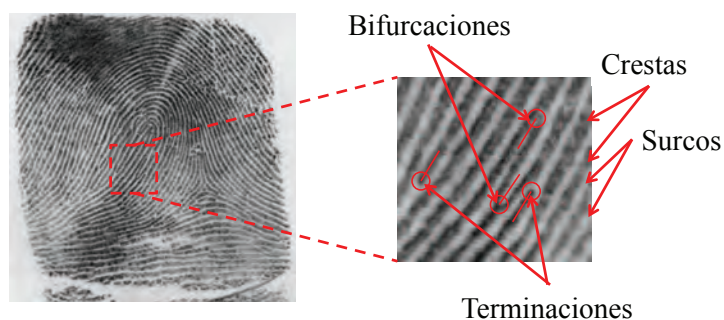


Figura 1.1: Ejemplo de huella dactilar y los dos tipos de minucias que comúnmente se usan en los algoritmos de comparación.

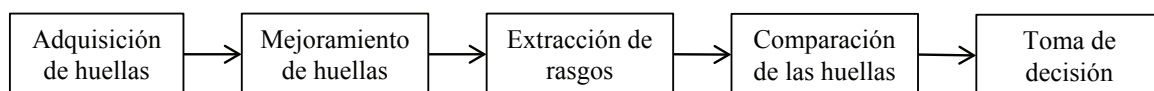


Figura 1.2: Diagrama de bloques que describe los sistemas de comparación de huellas.

informáticos que permitan su procesamiento de manera automática con bajos costos computacionales. Un ejemplo de estos sistemas es el *Integrated Automated Fingerprint Identification System (IAFIS)*, el cual contiene huellas de aproximadamente 73 millones de personas de interés criminal y recibe un promedio de 700 solicitudes de búsquedas por día <sup>1</sup>. Aunque las primeras aplicaciones fueron en las ciencias forenses, la popularidad de la temática se debe principalmente a las aplicaciones en sistemas civiles como el control de acceso físico a instalaciones, el control de acceso lógico a software y el control de votantes en elecciones (Maltoni et al. 2009).

Cada uno de los componentes de los sistemas de comparación de huellas (Figura 1.2) constituye objeto de investigación para la comunidad científica internacional. Esta tesis se enmarca en el desarrollo de algoritmos de comparación de huellas.

En el área de Biometría (Jain et al. 2011), la comparación de huellas (o cotejo de huellas) se realiza a través de una secuencia finita de pasos lógicos que retorna un valor de similitud, lo cual se conoce como algoritmo de comparación de huellas. Según el contexto de la aplicación, en esta investigación se propone clasificar los algoritmos de comparación de huellas como: *algoritmos para verificación de huellas* y *algoritmos para identificación de huellas latentes*. El objetivo de las aplicaciones de verificación de huellas consiste esencialmente en determinar dadas dos huellas tomadas de manera voluntaria en condiciones controladas, si estas proceden del mismo dedo. Por su parte, el objetivo de las aplicaciones de identificación de huellas latentes consiste básicamente en, dada una huella latente (huella formada de manera involuntaria en condiciones no controladas), buscar todas las huellas procedentes del mismo dedo presentes en una base de datos.

Los algoritmos para verificación de huellas son usados comúnmente en aplicaciones de control de acceso y los resultados en las competencias internacionales muestran una elevada eficacia. Solo por citar un ejemplo, en la competencia FVC-onGoing (Dorizzi et al. 2009) los mejores algoritmos alcanzan, para todos los parámetros de eficacia, valores entre 98.0% y 95.0% aproximadamente. Los problemas de verificación de huellas trabajan con huellas tomadas en condiciones controladas (*impresiones* en lo adelante) y con elevada calidad, lo cual justifica los altos valores de eficacia alcanzados. No obstante, se continúa investigando para obtener mejor eficacia con menores costos computacionales. Ejemplos de artículos recientes de verificación de huellas son (Hayfron-Acquah y Sarah Gyimah 2014, Syarif et al. 2014, Asha y Chellappan 2014, Vatsa 2014, Babatunde et al. 2014, Anthonioz y Champod 2014).

<sup>1</sup>Estadística del 2012 (Federal Bureau of Investigation 2012).

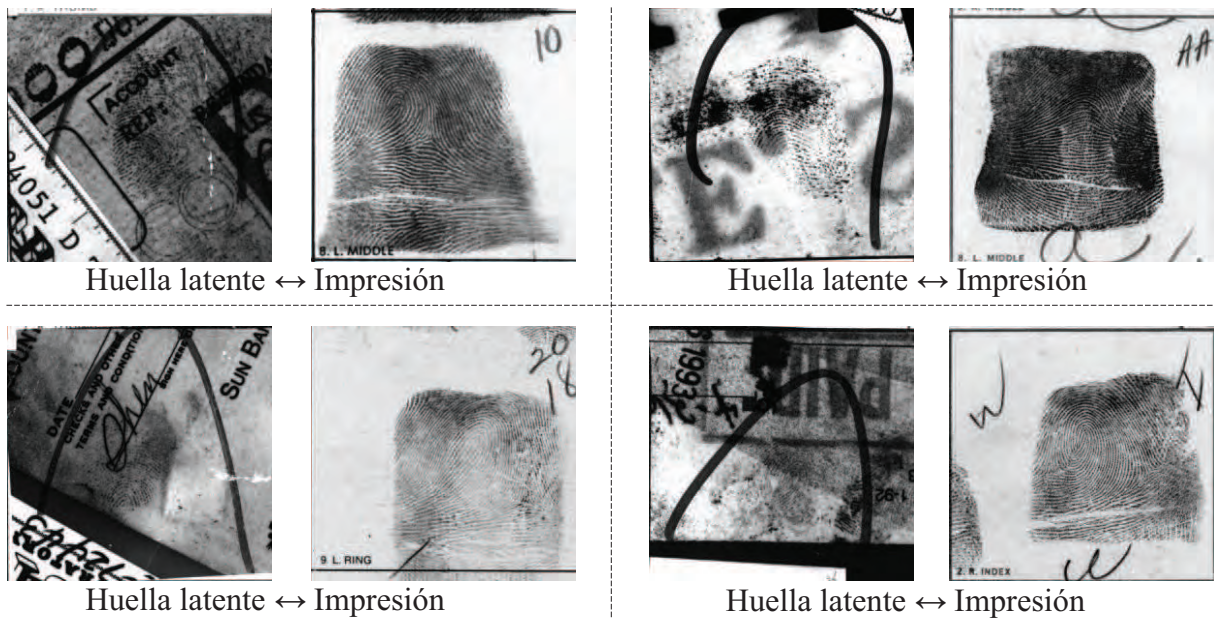


Figura 1.3: Cuatro ejemplos de pares huellas latentes - impresiones pertenecientes a la base de datos NIST SD27 (Garris y McCabe 2000). Nótese que, a diferencia de las impresiones, las huellas latentes contienen información parcial y mucho ruido.

Aunque la literatura sobre verificación de huellas es extensa, son pocos los trabajos sobre identificación de huellas latentes. Las huellas latentes son formadas de manera involuntaria por las personas al manipular objetos y los algoritmos que trabajan con huellas latentes son de baja eficacia; así lo muestra el 53.5% de eficacia alcanzado en (Paulino, Feng y Jain 2013) al comparar 258 huellas latentes contra 31998 impresiones. Varios autores (Cao et al. 2014, Paulino, Feng y Jain 2013, Jain y Feng 2011) atribuyen la dificultad en los problemas de identificación de huellas latentes principalmente a tres factores: baja calidad en las huellas latentes en cuanto a la claridad de las crestas; área pequeña de las huellas latentes comparado con las impresiones; y grandes distorsiones no lineales debido a los mecanismos de formación de las huellas (Figura 1.3). Por su impacto directo en el esclarecimiento de crímenes, es de interés para la comunicada científica internacional el desarrollo de algoritmos de mayor eficacia para la identificación de huellas latentes.

A pesar de los avances alcanzados en la comparación de huellas, aún existen limitaciones que afectan la calidad de los algoritmos en varios contextos de aplicación y que serán tratadas en el marco de esta investigación doctoral.

## 1.1 Descripción del problema

La calidad de los algoritmos de comparación de huellas se evalúa teniendo en cuenta el contexto en que se aplican. Por ejemplo, los algoritmos basados en micro características pueden ser los ideales en un sistema donde las huellas son tomadas con un sensor de 1000 ppp o más de resolución; sin embargo, la calidad de estos algoritmos de comparación es degradada cuando la resolución baja, así que no se recomienda su uso (Gupta y Sutton 2010, Zhao et al. 2010). Los parámetros de calidad más generales que deben cumplir los algoritmos de comparación de huellas son:

- Costo computacional aceptable: El algoritmo cumple las restricciones de memoria y tiempo en el contexto donde se aplica (Cappelli et al. 2010).
- Invarianza a la traslación: El algoritmo retorna alto valor de similitud entre huellas procedentes de un mismo dedo aunque las coordenadas de las huellas estén trasladadas horizontal o verticalmente (Maltoni et al. 2009).
- Invarianza a la rotación: El algoritmo retorna un alto valor de similitud entre huellas procedentes de un mismo dedo aunque las huellas estén rotadas (Maltoni et al. 2009).
- Tolerancia a las distorsiones no lineales de las huellas: El algoritmo es capaz de encontrar similitud entre aquellas áreas de la huella que han sido afectadas por las distorsiones no lineales resultantes de los mecanismos de formación de las huellas (Paulino, Feng y Jain 2013).
- Sensibilidad a la individualidad de las huellas: El algoritmo es capaz de distinguir las huellas de un mismo dedo de aquellas que proceden de dedos diferentes (Jea 2005).
- Insensibilidad a la elección de una sola alineación: Los algoritmos que comparan las huellas basados en alineaciones, rotan y trasladan una huella para superponerla sobre otra y hallar las coincidencias. Los algoritmos insensibles a la elección de una sola alineación asumen acertadamente que no se puede determinar la mejor alineación global a partir de la mejor alineación local. Téngase en cuenta que la mejor alineación local pudo haberse obtenido a partir de una correspondencia entre estructuras locales similares pero falsas. Incluso si la correspondencia entre estructuras locales fuera cierta, un alto valor de similitud local no garantiza que esta sea la mejor opción para alinear globalmente (Maltoni et al. 2009).
- Tolerancia a rasgos ausentes: Debido a las condiciones cambiantes de la piel o a los mecanismos de formación de las huellas, muchas veces hay rasgos ausentes y en estas

condiciones el algoritmo debe ser capaz de retornar un elevado valor de similitud al comparar huellas de un mismo dedo (Maltoni et al. 2009).

- Tolerancia a la baja calidad de las huellas: Debido a las condiciones cambiantes de la piel o los mecanismos de obtención de las huellas, en ocasiones muchos detalles de la huella no aparecen con claridad en la imagen, sin embargo el algoritmo debe retornar un elevado valor de similitud al comparar huellas de un mismo dedo (Jain y Feng 2011).
- Tolerancia a errores del extractor de rasgos: El algoritmo debe retornar un alto valor de similitud entre huellas de un mismo dedo aún cuando el extractor de rasgos no haya extraído todos los rasgos o se haya equivocado en algunos (Jea 2005).
- Determinismo: Dos aplicaciones del mismo algoritmo con los mismos parámetros debe retornar el mismo resultado.

En la actualidad la tecnología impone nuevos retos en el desarrollo de los algoritmos de comparación de huellas, como son los sistemas embebidos en arquitecturas ligeras de hardware y los estándares de interoperabilidad entre sistemas (Cappelli et al. 2010, Maltoni et al. 2009, Jain et al. 2010). En estos contextos los algoritmos basados en minucias muestran una elevada eficacia y eficiencia (Cappelli et al. 2010). Las minucias son los puntos donde la continuidad de las crestas se rompe (Figura 1.2) y gráficamente cada minucia se representa por una circunferencia y un segmento que indica la dirección de la minucia.

En el contexto de esta investigación, las minucias se representan matemáticamente por su posición (coordenadas  $x$  e  $y$ ) y dirección (ángulo  $\theta$  expresado en radianes). El origen del sistema de coordenadas se localiza en la esquina superior izquierda de las imágenes de manera que los valores de  $x$  se incrementan de izquierda a derecha, los valores de  $y$  se incrementan de arriba hacia abajo y los valores de  $\theta$  se incrementan de acuerdo al sentido de las manecillas del reloj. La dirección de las minucias de tipo *terminaciones* (Figura 1.4) se miden por el ángulo entre el eje horizontal y la línea que comienza al final de la cresta y atraviesa dicha cresta por el medio. La dirección de las minucias de tipo *bifurcaciones* (Figura 1.4) se miden por el ángulo entre el eje horizontal y la línea que comienza en el punto donde ocurre la bifurcación y se extiende de manera paralela a las dos crestas que se forman en la bifurcación.

Los algoritmos de comparación de huellas basados en minucias esencialmente buscan las minucias coincidentes entre las huellas que comparan; esto es, dada una huella representada por el conjunto de minucias  $A = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots, \mathbf{q}_n\}$  y otra huella representada por el conjunto  $B = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_m\}$  los algoritmos buscan un conjunto  $M \subset \{(\mathbf{p}_i, \mathbf{q}_j) : \mathbf{p}_i \in A \wedge \mathbf{q}_j \in B\}$  (Figura 1.5). Al comparar dos huellas se desconoce de antemano la rotación y traslación entre ellas, por tanto la correspondencia entre minucias es ambigua y cada minucia de una huella puede corresponder con cualquier minucia de la otra (Feng 2008).

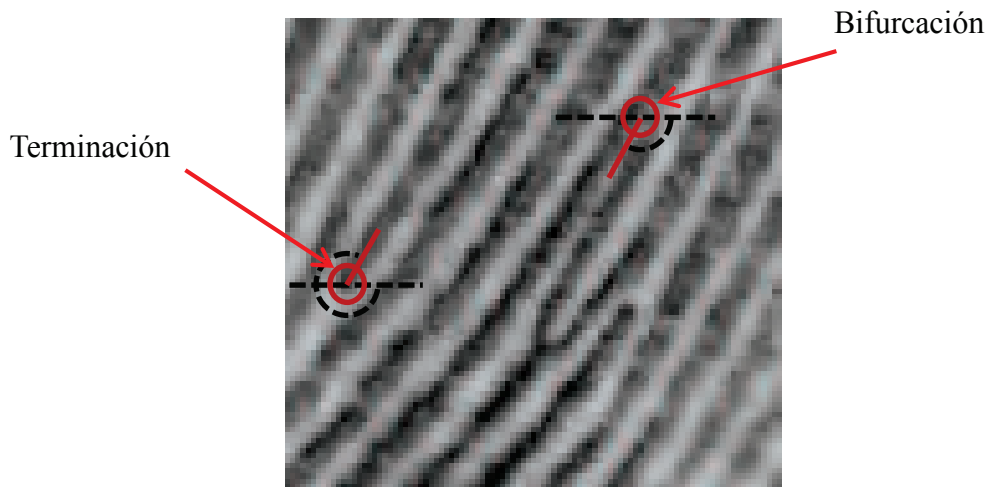


Figura 1.4: Ejemplo con los dos tipos de minucias usadas en esta investigación así como los ángulos que caracterizan las minucias.



Figura 1.5: Ejemplo de minucias coincidentes en huellas de un mismo dedo. Las líneas discontinuas unen las minucias coincidentes entre las huellas. Nótese que las líneas discontinuas no son paralelas, esto se debe a las deformaciones provocadas en los dedos durante el proceso de captura de huellas en escáneres de contacto.

De aquí que la tendencia actual es utilizar estructuras locales, relativas a las minucias, invariantes a la rotación y traslación (*descriptores de minucias*), que permitan establecer de manera rápida las correspondencias entre las minucias (Jain et al. 2010).

Uno de los descriptores de minucias más populares es el que utiliza tríos de minucias (Maltoni et al. 2009). Estos poseen características que los hacen idóneos para su uso en aplicaciones con las tecnologías mencionadas anteriormente. Los principales parámetros de calidad que deben cumplir los algoritmos basados en tríos de minucias (Medina-Pérez et al. 2012b) son:

- Invarianza al orden de las minucias en el rasgo: Un buen algoritmo de comparación de huellas basado en tríos de minucias es capaz de determinar la correspondencia correcta entre las minucias de dos tríos similares independientemente del orden de los rasgos en el trío. Por ejemplo, la Figura 1.6.a muestra dos tríos de minucias similares y la Figura 1.6.b muestra dos tríos de minucias iguales que algunos algoritmos determinan que no son similares porque tratan de hacer coincidir erróneamente los pares  $(\mathbf{q}_1, \mathbf{p}_1)$ ,  $(\mathbf{q}_2, \mathbf{p}_2)$ ,  $(\mathbf{q}_3, \mathbf{p}_3)$  cuando los algoritmos invariantes al orden de las minucias en el rasgo hallan las coincidencias correctas  $(\mathbf{q}_1, \mathbf{p}_1)$ ,  $(\mathbf{q}_2, \mathbf{p}_3)$ ,  $(\mathbf{q}_3, \mathbf{p}_2)$  en la Figura 1.6.a y las coincidencias correctas  $(\mathbf{q}_1, \mathbf{p}_2)$ ,  $(\mathbf{q}_2, \mathbf{p}_3)$ ,  $(\mathbf{q}_3, \mathbf{p}_1)$  en la Figura 1.6.b. Los algoritmos que no son invariantes al orden de las minucias en el rasgo obtienen menos coincidencias de minucias de las que deberían obtener.
- Sensibilidad a la reflexión de los tríos: Un buen algoritmo de comparación de huellas basado en tríos de minucias es capaz de determinar que un trío de minucias y su versión reflejada no son similares. En esta investigación se considera que un trío compuesto por las minucias  $B = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$  es una versión reflejada de otro trío compuesto por las minucias  $A = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$  si las correspondencias  $(\mathbf{q}_1, \mathbf{p}_1)$ ,  $(\mathbf{q}_2, \mathbf{p}_2)$ ,  $(\mathbf{q}_3, \mathbf{p}_3)$  obtenidas al rotar y trasladar los puntos de  $B$ , son puntos equidistantes con respecto a un eje perpendicular a los segmentos de recta que une los puntos coincidentes. La Figura 1.7.a muestra un ejemplo de dos tríos obtenidos a través de la reflexión. Los algoritmos que no son sensibles a la reflexión de los tríos obtienen coincidencias falsas de minucias que no deberían obtener.
- Sensibilidad a las direcciones de las minucias relativas a los lados: Un buen algoritmo de comparación de huellas basado en tríos de minucias toma en cuenta las direcciones de las minucias relativas a los lados del triángulo para determinar que dos tríos son similares. Las direcciones de las minucias relativas a los lados del triángulo formado por las minucias del trío son los ángulos formados por la dirección de cada minucia y cada uno de los lados del triángulo donde dicha minucia es un vértice. La Figura 1.7.b muestra un ejemplo de dos tríos cuyos triángulos formados son iguales (solo que rotados y trasladados) pero cuyas direcciones de las minucias relativas a los lados de los triángulos difieren; este es un caso donde un algoritmo no sensible a las direcciones de las minucias relativas a los lados erróneamente hallaría máxima similaridad entre

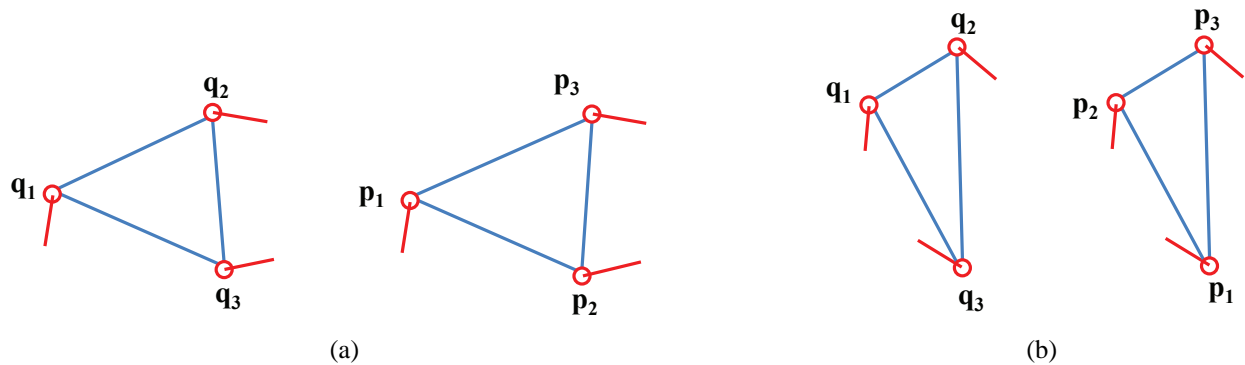


Figura 1.6: Dos pares de tríos de minucias realmente similares que son erróneamente considerados como no similares por algunos algoritmos de la literatura porque en el par (a) ordenan los rasgos de acuerdo a la longitud de los lados; mientras que en el par (b) hacen corresponder la minucia principal  $q_1$  del trío de la izquierda con la principal  $p_1$  del trío de la derecha.

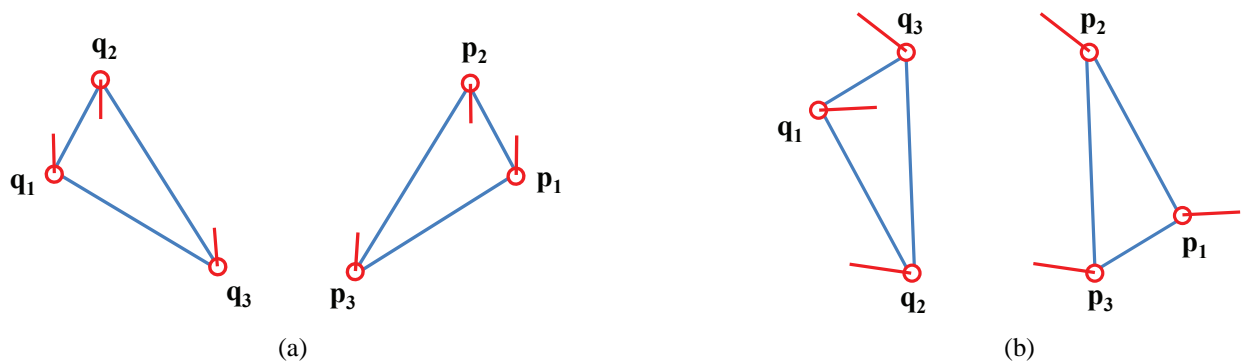


Figura 1.7: Dos pares de tríos de minucias no coincidentes porque en (a) un trío es una versión reflejada del otro; mientras que en (b) los pares de minucias  $(q_1, p_1)$ ,  $(q_2, p_2)$  y  $(q_3, p_3)$  tienen muy diferentes direcciones relativas a los lados de los triángulos.

los tríos cuando en realidad la similaridad debería ser baja por haber mucha diferencia entre las direcciones de las minucias relativas a los lados.

El análisis de la literatura revela que algunos algoritmos no son invariantes al orden de las minucias en los tríos por lo que no obtienen todos los pares de tríos verdaderos coincidentes entre las huellas que se comparan. Otros algoritmos no son sensibles a la reflexión de los tríos y/o a las direcciones de las minucias relativas a los lados así que obtienen más pares de tríos coincidentes que los que realmente hay entre las huellas que se comparan. Hay algoritmos que no son tolerantes a rasgos ausentes y/o a los errores del extractor de rasgos por lo que aumentan la cantidad de pares de minucias falsas coincidentes y disminuyen la cantidad de verdaderas coincidentes. Algunos de los algoritmos existentes realizan una



sola alineación de las huellas que comparan por lo que no encuentran todas las minucias verdaderas coincidentes que deberían hallar. Hay algoritmos basados en tríos de minucias que no son deterministas por lo que no garantizan que dos comparaciones de las mismas huellas den resultados acertados.

Basado en lo anterior, esta investigación se basa en la hipótesis de que si se crean nuevos algoritmos basados en tríos de minucias superando las limitaciones identificadas entonces se puede incrementar la eficacia de los algoritmos basados en tríos de minucias y se puede alcanzar una eficacia similar o superior a la de los algoritmos del estado del arte basados en otras representaciones.

## **1.2 Objetivo general**

Crear algoritmos para verificación e identificación de huellas basados en tríos de minucias que superen la eficacia de los algoritmos basados en tríos de minucias y que alcancen eficacia similar o superior a la de los algoritmos del estado del arte basados en otras representaciones.

## **1.3 Objetivos particulares**

1. Crear una forma de representación basada en tríos de minucias, invariante a la traslación y rotación, y sensible a las direcciones de las minucias relativas a los lados.
2. Crear una función de comparación de tríos de minucias, basada en la representación anterior, tolerante a las distorsiones no lineales de las huellas, invariante al orden de las minucias en el rasgo y sensible a la reflexión de los tríos.
3. Desarrollar un algoritmo para la verificación de huellas basado en la nueva forma de representación y función de comparación de tríos de minucias, con eficacia superior a los algoritmos basados en tríos de minucias y que alcance eficacia similar o superior a la de los algoritmos del estado del arte basados en otras representaciones.
4. Adaptar el nuevo algoritmo de comparación de huellas para aplicarlo en la identificación de huellas latentes, con eficacia similar o superior a los reportados en la literatura y costos computacionales aceptables.

## 1.4 Aportaciones

Las aportaciones de esta investigación son las siguientes:

1. Una forma de representación de tríos de minucias que no incluye una minucia principal y donde las minucias se encuentran ordenadas de acuerdo al sentido de las manecillas del reloj. El orden de acuerdo al sentido de las manecillas del reloj permite que al comparar dos tríos basta con rotar tres veces el trío de acuerdo al mismo sentido para encontrar la mejor coincidencia independiente de la rotación. Todos los rasgos del trío son relativos a las posiciones y orientaciones de las minucias logrando así la invarianza a la traslación de las huellas. Esta representación es sensible a las direcciones de las minucias relativas a los lados ya que incluye estas direcciones en su representación.
2. Una función de comparación de tríos de minucias que rota tres veces los tríos de acuerdo al sentido de las manecillas del reloj para encontrar la mejor coincidencia logrando así la sensibilidad a la reflexión de los tríos, la invarianza a la rotación de las huellas y la invarianza al orden de las minucias en los tríos. Para comparar los rasgos de los tríos se usan umbrales que permiten la tolerancia a las distorsiones locales de las huellas. La nueva función de comparación se basa en propiedades de los lados de los tríos para descartar comparaciones sin comparar todos los rasgos reduciendo así los tiempos de comparación de los rasgos.
3. Tres algoritmos de comparación de huellas que usan la nueva forma de representación y función de comparación de tríos de minucias para hallar múltiples clusters coincidencias locales logrando así la insensibilidad a la elección de una sola alineación. Por cada minucia, los algoritmos calculan los tríos que involucran a las siete minucias más cercanas lo que permite la tolerancia a los rasgos ausentes, la baja calidad de las huellas y los errores del extractor de rasgos. Los nuevos algoritmos usan una novedosa heurística y estrategia ávida para mezclar un subconjunto del total de clusters encontrados; logrando así la tolerancia a las distorsiones globales presentes en las huellas. Los algoritmos propuestos son deterministas y los resultados alcanzados muestran una sensibilidad a la individualidad de las huellas elevada con respecto a otros algoritmos propuestos en la literatura. Los resultados en la competencia internacional FVC-onGoing (Dorizzi et al. 2009) muestran que los algoritmos tienen costos computacionales aceptables.
4. Una base de datos pública<sup>2</sup> que servirá como referencia común para futuras investigaciones en la identificación de huellas palmares latentes.

---

<sup>2</sup>Disponible en <http://www.gpds.ulpgc.es>

5. Una plataforma pública<sup>3</sup> de experimentación en C# con múltiples algoritmos de comparación de huellas, protocolos de experimentación y extractores de rasgos que todos los investigadores pueden utilizar para futuras investigaciones.

## 1.5 Organización de la tesis

El capítulo 2 describe el estado del arte a partir de dos taxonomías donde se agrupan los algoritmos de comparación de huellas. El capítulo 2 también resume las limitaciones de los trabajos relacionados con esta investigación. El capítulo 3 introduce la nueva forma de representación y función de comparación de tríos de minucias. Este capítulo también presenta los nuevos algoritmos de comparación de huellas basados en la nueva forma de representación y comparación de tríos de minucias. Los capítulos 4 y 5 presentan el desempeño de los algoritmos en el contexto de la verificación de huellas y la identificación de huellas latentes respectivamente. El capítulo 6 resume las conclusiones de esta investigación y presenta nuevas líneas de investigación que se seguirán a partir de los resultados obtenidos.

---

<sup>3</sup>Disponible en <http://www.codeproject.com/Articles/97590/A-Framework-in-C-for-Fingerprint-Verification>



# 2

## Trabajo relacionado

Es este capítulo se describen los algoritmos de comparación de huellas de acuerdo a dos taxonomías: 1. algoritmos de comparación de huellas según el dominio del problema y 2. algoritmos de comparación de huellas según los rasgos que usan. Se hace mayor énfasis en los algoritmos basados en tríos de minucias porque constituyen el campo de la presente investigación. Finalmente se dan los argumentos que motivan esta investigación.

De acuerdo al dominio del problema, los algoritmos de comparación de huellas se clasifican en: *algoritmos para la verificación de huellas* y *algoritmos para la identificación de huellas latentes*. Según el tipo de rasgos que usan, se clasifican en: *algoritmos basados en las imágenes*, *algoritmos basados en características globales*, *algoritmos basados en características locales* y *algoritmos basados en micro características*.

Los algoritmos de comparación de huellas comparan dos huellas y devuelven un valor de similitud. De las dos huellas comparadas, en lo adelante llamaremos *huella plantilla* a aquella de identidad conocida y *huella consulta* a la huella de identidad desconocida que se desea comparar contra una o muchas huellas plantillas. En los sistemas de reconocimiento de huellas dactilares usualmente las huellas plantillas están guardadas en algún medio de almacenamiento mientras que las huellas consultas solo persisten mientras no terminan las comparaciones que la involucran.

## 2.1 Clasificación de los algoritmos de comparación de huellas según el dominio del problema

Los algoritmos se clasifican según el dominio del problema de acuerdo a las siguientes categorías: *algoritmos para la verificación de huellas* y *algoritmos para la identificación de huellas latentes*.

### 2.1.1 Algoritmos para la verificación de huellas

La verificación de huellas consiste en determinar si las huellas plantilla y consulta tienen la misma identidad; en otras palabras, determinar si son huellas del mismo dedo.

Los algoritmos para verificación de huellas (o simplemente *algoritmos para verificación*) trabajan con huellas tomadas en condiciones controladas por lo cual las imágenes suelen tener gran parte de la información de las huellas, buena claridad de las crestas y poco ruido. De aquí que actualmente se reporten altos valores de eficacia en la verificación de huellas, por ejemplo, en la competencia activa FVC-onGoing (Dorizzi et al. 2009) todos los algoritmos evaluados en la base de datos FMISO-HARD-1.0 (ver Cuadro 4.5) alcanzaron menos de un 5.0% de error (o lo que es lo mismo: más de un 95.0% de eficacia) para el parámetro *Equal Error Rate* (EER) (Cappelli et al. 2006).

Aunque se han reportado altos valores de eficacia de verificación al trabajar con huellas obtenidas por sensores de gran área y elevada resolución, no ocurre lo mismo con los algoritmos para verificación en los sistemas embebidos en arquitecturas ligeras de hardware (Cappelli et al. 2010, Maltoni et al. 2009). Además, existe la tendencia en los fabricantes a reducir el área de sensado para disminuir los costos de los sensores y aumentar sus dominios de aplicación (Maltoni et al. 2009, Prabhakar et al. 2011). Para este tipo de sensores los resultados de verificación alcanzados son bajos. Por ejemplo, el algoritmo ganador de la competencia de verificación de huellas FVC2006 (BioLab 2006) alcanzó aproximadamente un 77% de verdaderos positivos para un 0% de falsos positivos en la base de datos DB1\_A.

### 2.1.2 Algoritmos para la identificación de huellas latentes

Las personas suelen dejar pistas cuando cometen algún crimen. Entre las pistas más importantes están las huellas latentes. De aquí que la identificación eficaz de las huellas latentes contribuya al esclarecimiento de los hechos delictivos.

Algoritmo	Compañía	Tiempo promedio de comparación	EER	FMR100	FMR1000	ZeroFMR
Nitgen_ISO	Sí	3 ms	1.09 %	1.77 %	4.38 %	11.88 %
Triple_M_ISO	Sí	3 ms	1.10 %	1.64 %	3.16 %	11.61 %
EMB9200	Sí	7 ms	1.11 %	1.15 %	2.08 %	4.74 %
SFCore	Sí	12 ms	1.41 %	1.58 %	2.70 %	19.81 %
MM_FMISO	Sí	18 ms	1.50 %	1.88 %	3.25 %	4.41 %
MntModel	No	82 ms	1.59 %	1.76 %	2.82 %	4.71 %
MCC (Baseline)	No	237 ms	1.77 %	2.05 %	3.62 %	6.07 %
Fingerprint Matcher ISO	Sí	5 ms	1.91 %	2.25 %	3.72 %	5.71 %
FMTISO	Sí	14 ms	1.99 %	2.33 %	3.50 %	6.24 %
alguxntaix	No	5 ms	2.38 %	3.17 %	5.20 %	6.74 %
APF_FMISO	Sí	3 ms	2.55 %	3.15 %	4.58 %	14.35 %
SourceAFIS	No	235 ms	3.68 %	4.85 %	6.71 %	8.91 %
LSM	Sí	17 ms	3.83 %	5.03 %	8.73 %	12.71 %

Cuadro 2.1: Resultados previos en la base de datos FMISO-HARD-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009). Los parámetros de eficacia **EER**, **FMR100**, **FMR1000** y **ZeroFMR** se explican en la sección 4.1.

La identificación de huellas latentes consiste en determinar cuál o cuáles huellas plantillas en una base de datos tienen la misma identidad que una huella consulta dada. En este tipo de problemas las huellas consultas son huellas latentes formadas de manera involuntaria por las personas al manipular objetos; mientras que las huellas plantillas son tomadas en condiciones controladas (también conocidas como *impresiones*) a partir de sensores o de modelos dactilográficos (Komarinski 2005).

Investigaciones recientes muestran que la eficacia de los algoritmos para verificación se puede degradar al ser usados para la identificación de huellas latentes (y viceversa) (Decann y Ross 2012). Lograr elevada eficacia en la identificación de huellas latentes es aún más complicado que en la verificación (Jain y Feng 2011, Sankaran et al. 2011, Puertas et al. 2010), por ejemplo, Jain y Feng (Jain y Feng 2011) reportan un 74.0% de verdaderos positivos para un 0% de falsos positivos para una base de datos de menos de 100000 impresiones (ver Cuadro 2.2). Estos resultados se deben a que las huellas latentes presentan deformaciones mayores que las impresiones, la información de las crestas es de baja claridad y el tamaño de la huella es pequeño y con mucha información ausente (Jain y Feng 2011).

Algoritmo	Referencia	Impresiones adicionales a NIST SD27	Rasgos adicionales a minucias	Eficacia
VeriFinger SDK	(Yoon et al. 2010)	0	No	~58.0 %
VeriFinger SDK	(Yoon et al. 2011)	27000	No	~13.0 %
		27000	Yes	~39.0 %
Bozorth3	(Mikaelyan y Bigun 2012)	0	No	47.0 %
k-plet		0	No	55.0 %
Descriptor-based hough transform (DBHT)	(Paulino et al. 2011)	2000	No	62.4 %
VeriFinger SDK		2000	Yes	48.0 %
DBHT	(Liu, Arora, Cao, Jain y Lansing 2013) y (Paulino, Feng y Jain 2013)	31740	No	53.5 %
Neighboring minutiae-based descriptor-1 (NMD-1)	(Jain et al. 2008)	0	No	~63.0 %
		2000	No	~50.0 %
		0	Yes	~86.0 %
		2000	Yes	79.5 %
Neighboring minutiae-based descriptor-2 (NMD-2)	(Jain y Feng 2011)	29000	No	34.9 %
		29000	Yes	74.0 %

Cuadro 2.2: Resumen de la eficacia de identificación de huellas latentes reportada en la literatura para la base de datos NIST SD27 (Garris y McCabe 2000). En algunos casos los valores de eficacia fueron inferidos de gráficas en los artículos originales y esta es la razón por la cual aparecen en el cuadro con el símbolo de aproximación (~). La columna titulada **Impresiones adicionales a NIST SD27** se refiere a la cantidad de impresiones (huellas de alta calidad tomadas en condiciones controladas) que fueron incorporadas a la base de datos NIST SD27 para incrementar su tamaño.

## 2.2 Clasificación de los algoritmos de comparación de huellas según los rasgos que usan

De acuerdo a los rasgos que usan, los algoritmos se clasifican en: *algoritmos basados en las imágenes*, *algoritmos basados en características globales*, *algoritmos basados en características locales*, *algoritmos basados en micro características* y *algoritmos híbridos*.



### 2.2.1 Algoritmos basados en las imágenes

Estos algoritmos no realizan ninguna abstracción de las huellas sino que correlacionan directamente las imágenes en escala de grises o sus transformadas de Fourier (Kumar et al. 2013, Csongrády y Pivarciová 2011, Martínez-Díaz y Carmona-Troyo 2010). Para poder correlacionar las imágenes, numerosas transformaciones de traslación y rotación (alineaciones) deben ser probadas, lo que hace que estos métodos sean lentos. Aunque estos algoritmos no son invariantes a la traslación ni la rotación, logran un poco de tolerancia con la discretización del espacio de traslación y rotación. El problema con la discretización es que se pierde información, por lo que no se garantiza encontrar la transformación correcta de traslación y rotación. Otros algoritmos realizan la alineación basada en los puntos focales (núcleos y deltas de huellas (Maltoni et al. 2009)), siendo sensibles entonces a la ausencia de estos rasgos y a los errores en su extracción.

### 2.2.2 Algoritmos basados en características globales

Estos algoritmos comparan las huellas representadas por rasgos que describen los patrones globales de las huellas (Hayfron-Acquah y Sarah Gyimah 2014, Syarif et al. 2014, Guesmi et al. 2013, Feng et al. 2013). Algunos de estos rasgos son: *imagen direccional*, *puntos focales* y *FingerCode*. Al trabajar con rasgos globales de las huellas, estos algoritmos son los más tolerantes a la baja calidad de las huellas y a sus distorsiones no lineales, aunque son de baja sensibilidad a la individualidad de las huellas al trabajar con huellas de buena calidad como son las capturadas por sensores que abarcan la mayor parte de la tercera falange del dedo. La mayoría de las técnicas usadas por estos algoritmos para la alineación de las huellas coinciden con las usadas por los algoritmos basados en las imágenes, por lo tanto tienen las siguientes desventajas: son lentos y no son invariantes a la traslación ni a la rotación. Por otra parte, son poco tolerantes a la ausencia de los rasgos y a los errores del extractor de rasgos.

### 2.2.3 Algoritmos basados en características locales

Estos algoritmos comparan las huellas representadas en términos de características locales como *Scale-Invariant Feature Transform* (SIFT) (Zhou et al. 2013), las minucias (Maltoni et al. 2009) o descriptores de minucias (Jain et al. 2010). Las minucias son las características locales más populares en la literatura (Maltoni et al. 2009) y esto se debe a que constituyen

la base de las representaciones que utilizan los especialistas criminalistas para la comparación de huellas y en las cortes judiciales es aceptado el testimonio de estos expertos acerca de la identidad de un sospechoso basado en minucias coincidentes (Jain et al. 2010). La configuración global de las minucias es altamente discriminante (Maltoni et al. 2009) por lo que varios modelos teóricos se han basado en ellas para probar la individualidad de las huellas (Pankanti et al. 2002).

Por largo tiempo, la comparación de huellas basada solo en minucias fue tratada como el problema de la comparación de patrones de puntos 2D, pero estos algoritmos son costosos computacionalmente, son poco tolerantes a las distorsiones no lineales de las huellas y no son deterministas (Cappelli et al. 2010, Jea y Govindaraju 2005, Ramoser et al. 2002, Gwo-Cheng et al. 2007).

La tendencia en los algoritmos actuales es la de establecer la correspondencia de minucias a través de la comparación de descriptores de minucias (Jain et al. 2010). Los descriptores de minucias están formados por una o más minucias e información adicional que hace a la estructura invariante a la rotación y traslación. Los descriptores de minucias pueden ser clasificados fundamentalmente en *descriptores basados en las crestas*, *descriptores basados en texturas* y *descriptores basados en minucias vecinas*.

Los descriptores basados en las crestas asocian a cada minucia parte de la información contenida en la cresta a la que pertenece dicha minucia (Nguyen et al. 2013, Angeloni y Marana 2013, Huong Thuy et al. 2013, Jain y Feng 2011, Jain et al. 1997). Para esto se toman puntos de la cresta a intervalos constantes y se realizan mediciones con respecto a la minucia, de aquí que los costos de almacenamiento sean elevados (Jiang y Yau 2000). Para hallar correspondencias correctas entre minucias, estos algoritmos requieren que exista la mayor cantidad posible de información de las crestas, sin embargo estas condiciones no se garantizan al trabajar con huellas parciales por lo que estos algoritmos son poco tolerantes a rasgos ausentes. Estos descriptores toman en cuenta gran parte del área de la huella por lo que son poco tolerantes a las distorsiones no lineales de las huellas.

Los descriptores basados en texturas enriquecen cada minucia con información de la orientación o frecuencia de las crestas en puntos seleccionados alrededor de la minucia (Garg y Rane 2013, Paulino, Liu, Cao y Jain 2013, Tico y Kuosmanen 2003). Similar a los descriptores basados en crestas, algunos descriptores basados en texturas tienen elevados costos de almacenamiento debido a la cantidad de mediciones que debe realizar alrededor de las minucias. Otra desventaja de algunos de estos algoritmos es que son sensibles a la elección de una sola alineación.

Los descriptores basados en minucias vecinas combinan cada minucia con información de otras minucias que se encuentran a una distancia euclidiana menor que umbrales predefinidos (Asha y Chellappan 2014, Paulino, Feng y Jain 2013, Liu, Arora, Cao, Jain y Lansing 2013, Benhammadi y Bey 2013, Wen et al. 2013, Garg y Rane 2013, Cappelli, Ferrara y

Maltoni 2012). Estos algoritmos tienen bajos costos computacionales, así que son idóneos para el desarrollo de sistemas empotrados en arquitecturas ligeras (Cappelli et al. 2010). También son adecuados para trabajar en sistemas basados en estándares de interoperabilidad puesto que los estándares que han surgido indican una convergencia hacia los algoritmos basados solo en minucias (Cappelli et al. 2010). Uno de los descriptores más populares basados en minucias vecinas es el caso de los tríos de minucias (Maltoni et al. 2009), por lo que a continuación se describen los algoritmos basados en tríos de minucias propuestos en la literatura.

## Algoritmo JY

El algoritmo de Jiang y Yau (Jiang y Yau 2000) fue creado a partir de abstraer la manera en que los especialistas criminalistas comparan las huellas y es reconocido en la literatura como el primer algoritmo para verificación basado en tríos de minucias. Según los autores, los tríos de minucias son de alta tolerancia a las distorsiones no lineales presentes en las huellas mientras que su representación y comparación se pueden realizar de manera eficiente (Jiang y Yau 2000).

Dada una huella, por cada minucia de dicha huella el algoritmo construye un trío a partir de cada minucia y sus dos minucias más cercanas de acuerdo a la distancia euclidiana. Construir un trío de minucias por cada minucia de la huella provoca que el algoritmo sea poco tolerante a los errores del extractor de rasgos (Jea 2005) ya que la aparición de minucias falsas y/o la ausencia de minucias verdaderas provoca la aparición de tríos falsos y la ausencia de tríos verdaderos. La representación propuesta es una tupla que incluye mediciones de la minucia principal relativas a la minucia más cercana y a la más lejana (Figura 2.1), esta representación es invariante a la traslación y rotación. Para comparar dos tuplas los autores proponen una función de similitud basada en la suma de las diferencias pesadas componente a componente. Esta forma de representación y función de comparación no logran la invarianza al orden de las minucias en el rasgo.

Para alinear las huellas, este algoritmo usa el par de minucias coincidentes que maximiza la función de similitud de descriptores, siendo entonces sensible a la elección de una sola alineación (Figura 2.2).

## Algoritmo KV

A diferencia de la mayoría de los trabajos basados en tríos de minucias, el algoritmo propuesto por Kovács-Vajna (Kovács-Vajna 2000) combina varias formas de representación

para realizar la comparación de huellas.

KV compara la vecindad de 16x16 píxeles de cada minucia de la huella consulta contra las respectivas vecindades de las minucias en la huella plantilla y selecciona una lista de posibles minucias coincidentes localmente. Luego usa las minucias coincidentes localmente para ir generando y comparando triángulos incrementalmente. Cada trío que se explora se construye a partir de los coincidentes encontrados hasta el momento. Finalmente verifica las coincidencias encontradas comparando los perfiles en escala de grises para los diferentes pares de minucias entre las huellas.

Para determinar que dos tríos son coincidentes este algoritmo compara la longitud de los lados de los triángulos formados por los tríos, así como el ángulo formado por las rectas que conectan las minucias correspondientes entre los triángulos. Esta forma de representación y comparación de tríos de minucias no es sensible a la reflexión de los tríos ni a las direcciones de las minucias relativas a los lados.

En este algoritmo cada trío que se explora se construye a partir de los coincidentes encontrados hasta el momento lo cual crea una interdependencia fuerte entre los tríos de manera tal que un par tríos encontrados que sean falsos coincidentes afecta todo el proceso (Qi et al. 2005) (sensible a la elección de una sola alineación).

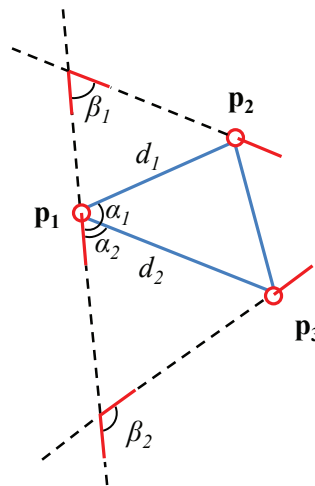


Figura 2.1: Los componentes de la representación propuesta por Jiang y Yau (Jiang y Yau 2000).  $d_1$  y  $d_2$  son las distancias de la minucia  $p_1$  a la minucia más cercana y más alejada respectivamente.  $\alpha_1$  y  $\alpha_2$  son los ángulos que forma la dirección de la minucia  $p_1$  con respecto a los segmentos de longitud  $d_1$  y  $d_2$  respectivamente.  $\beta_1$  y  $\beta_2$  son los ángulos que forma la dirección de la minucia  $p_1$  con respecto a las direcciones de las otras dos minucias. Esta representación también incluye el conteo de crestas y el tipo de las minucias.

## Algoritmo PN

El algoritmo propuesto por Parziale y Niel (Parziale y Niel 2004) trabaja sobre los tríos de minucias obtenidos a partir de la triangulación de Delaunay (poco tolerante a los errores del extractor de rasgos). Este algoritmo tiene la particularidad de que no compara tríos de minucias directamente sino que se basa en las comparaciones de pares de minucias para hallar los tríos coincidentes. Cada par de minucias está descrito por la distancia entre las minucias, los ángulos que forman las direcciones de las minucias con respecto al segmento que las une y el ángulo entre las direcciones de las minucias. Dos pares de minucias se consideran similares si se cumplen las siguientes condiciones:

- La razón entre el valor absoluto de las diferencias de los lados con respecto al menor de los lados es menor que un umbral.
- Las diferencias de los ángulos que forman las direcciones de las minucias con respecto al segmento que las une son menores que un umbral.
- La diferencia del ángulo entre las direcciones de las minucias es menor que un umbral.

PN realiza una alineación de las huellas por cada par de tríos que encuentra coincidentes lo cual está inspirado en la idea original de Udupa *et al.* (Udupa U et al. 2001) para un

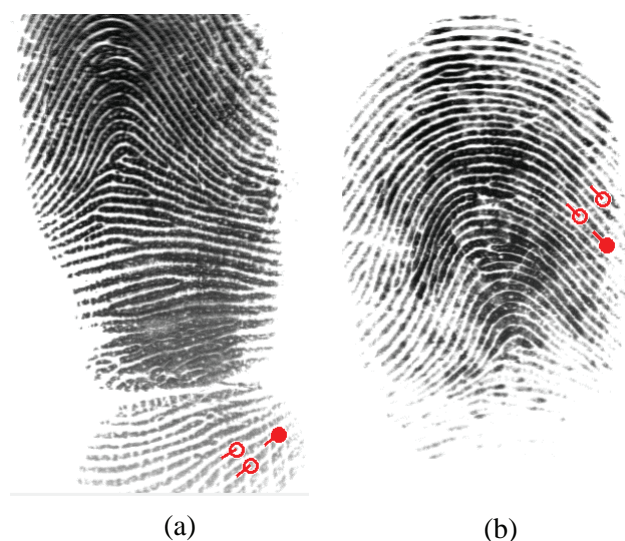


Figura 2.2: Las minucias coincidentes encontradas por el algoritmo de Jiang y Yau (Jiang y Yau 2000) al comparar las huellas (a) 14\_1 y (b) 14\_7 de la base de datos DB1\_A de la competencia FVC2004. Este es un ejemplo de falsas coincidencias de minucias que maximizan la similitud de descriptores. La minucia principal se encuentra con color relleno.

algoritmo basado en segmentos. Realizar una alineación por cada par de tríos similares resuelve el problema de la alineación basada en un solo par de minucias coincidentes pero tiene un costo de tiempo demasiado elevado en aquellos casos que se encuentran numerosos pares de tríos coincidentes.

El algoritmo no es sensible a la reflexión de los tríos porque al comparar los tríos de minucias por pares de minucias no tiene en cuenta la disposición espacial de un par con respecto a los demás.

## Algoritmo JG1

El algoritmo propuesto por Jea y Govindaraju (Jea y Govindaraju 2005) utiliza una representación de tríos de minucias inspirada en la de Jiang y Yau (Jiang y Yau 2000). La diferencia está en que las minucias secundarias se ordenan de acuerdo al sentido de las manecillas del reloj y que toma en cuenta el ángulo interno del triángulo con vértice en la minucia principal en lugar de los ángulos formados por la dirección de la minucia principal relativos a los lados del trío. Por otra parte, en esta representación no se tiene en cuenta el tipo de las minucias, argumentando que la estimación acertada del tipo de minucia es un problema complicado cuando varía la presión de los dedos en la toma de las huellas. Tampoco se tiene en cuenta el conteo de crestas argumentando que esta información no está presente en muchas de las bases de datos existentes.

Para determinar si dos tuplas son similares, el valor absoluto de las diferencias entre componente y componente debe ser inferior a umbrales que son calculados dinámicamente en dependencia de las tuplas que se comparan.

Similar al algoritmo JY, para construir los tríos de minucias, JG1 asocia a cada minucia sus dos vecinas más cercanas, lo que provoca que el algoritmo sea poco tolerante a los errores del extractor de rasgos.

Si bien esta forma de representación evita equivocarse en casos como la Figura 1.6(a), sí se equivoca en casos como la Figura 1.6(b) por lo que no es invariante al orden de las minucias en el rasgo. Como desventaja con respecto a la representación de Jiang y Yau (Jiang y Yau 2000), la representación usada en JG1 no es sensible a las direcciones de las minucias relativas a los lados.

Para calcular el valor de similitud se usa una red neuronal. Por el componente aleatorio en la inicialización de los pesos de la red neuronal este algoritmo es no determinista.

Este algoritmo realiza  $m$  alineaciones globales por lo que es insensible a la elección de una sola alineación. El inconveniente está en que si se selecciona un alto valor de  $m$  el algoritmo puede resultar muy lento, mientras que un valor pequeño de  $m$  puede ocasionar que el algoritmo ya no sea insensible a la elección de una sola alineación.

## Algoritmo JG2

Para eliminar las deficiencias de JG1, Jea y Govindaraju (Jea 2005) proponen esta versión mejorada. Una diferencia consiste en que toma en cuenta los ángulos formados por la dirección de la minucia principal relativos a los lados del trío, en lugar del ángulo interno del triángulo con vértice en la minucia principal, logrando así la sensibilidad a las direcciones de las minucias relativas a los lados. Otra diferencia consiste en que en este algoritmo se forman tríos tomando en cuenta más de dos vecinos por minucias, lográndose la tolerancia a los errores del extractor de rasgos. Para aumentar la velocidad de comparación, JG2 indexa los tríos discretizando la posición de cada minucia secundaria con respecto a la minucia principal. Como esta representación depende de una minucia principal, no es invariante al orden de las minucias en el rasgo.

## Algoritmo RUR

Este algoritmo fue propuesto por Reisman *et al.* (Reisman et al. 2005) para trabajar en sistemas embebidos en arquitecturas ligeras de hardware como las tarjetas inteligentes (Maltoni et al. 2009). RUR es un algoritmo híbrido donde los tríos de minucias son usados para alinear las huellas y luego realiza la comparación con un algoritmo basado en características globales (Ross et al. 2003).

Los tríos son representados tomando en cuenta los siguientes elementos:

- La altura con respecto al mayor lado del triángulo.
- La longitud del segmento que une la base de la altura con el vértice más cercano de acuerdo al sentido de las manecillas del reloj.
- La razón entre el lado mayor y el segmento que une la base de la altura con el vértice más cercano de acuerdo al sentido de las manecillas del reloj.
- El ángulo que forma el lado mayor con respecto al eje horizontal de la imagen.

Dos tríos son considerados como similares si las diferencias componente a componente son inferiores a umbrales. Esta forma de representación y función de comparación no son invariantes al orden de las minucias en el rasgo ni sensibles a las direcciones de las minucias relativas a los lados.

La huella plantilla está representada por un subconjunto de tríos de minucias cuyo tamaño puede variar según las restricciones de hardware en este tipo de aplicaciones. Los autores

proponen varias técnicas de evaluación de tríos que son aplicadas a la huella plantilla, pero la representación de la huella consulta incluye todos los posibles tríos de minucias hallados en la misma. La comparación de los tríos de minucias se realiza fuera del hardware de arquitectura ligera. Luego, este hardware recibe los parámetros de alineación obtenidos a partir de los tríos coincidentes y entonces realiza la comparación basada en características globales para determinar la similitud de las huellas.

## Algoritmo TB

La esencia del algoritmo de Tan y Bhanu (Tan y Bhanu 2006) consiste en aplicar múltiples transformaciones de escala, traslación y rotación que maximicen la cantidad de tríos de minucias coincidentes. Para eso codifican los posibles valores de escala, traslación y rotación en cromosomas que serán parte de un algoritmo genético. La función a optimizar primero verifica que haya una cantidad suficiente de minucias coincidentes y para esto cada minucia de la huella consulta es transformada y comparada con cada minucia de la huella plantilla usando la distancia euclidiana. En caso de existir una cantidad suficiente de minucias coincidentes entonces la función de evaluación retorna la cantidad de tríos de minucias similares.

Para describir los tríos de minucias este algoritmo usa la siguiente tupla  $(\sigma_{min}, \sigma_{med}, \phi, \eta, \lambda, \chi_1, \chi_2, \chi_3, \xi_1, \xi_2, \xi_3)$  donde:

- $\sigma_{min} = \min\{\sigma_i\}$  es el menor de los ángulos internos del triángulo formado por el trío de minucias.
- $\sigma_{med} = 180^\circ - \sigma_{max} - \sigma_{min}$ , siendo  $\sigma_{max}$  el mayor de los ángulos internos del triángulo.
- $\phi = \text{sign}(Z_{21} \times Z_{32})$  donde  $\text{sign}$  es la función signo y  $\times$  es el producto cruzado de dos números complejos.  $Z_{21} = Z_2 - Z_1$  y  $Z_{32} = Z_3 - Z_2$  siendo  $Z_i = x_i + jy_i$  el número complejo ( $j = \sqrt{-1}$ ) correspondiente a las coordenadas  $(x_i, y_i)$  de la minucia  $\mathbf{q}_i$ ,  $i = 1, 2, 3$ ; para lo cual consideramos  $\mathbf{q}_1$  como la minucia que coincide con el vértice del ángulo  $\sigma_{max}$ ;  $\mathbf{q}_2$  la del ángulo  $\sigma_{min}$  y  $\mathbf{q}_3$  la del ángulo  $\sigma_{med}$ .
- $\eta = 4\nu_1 + 2\nu_2 + \nu_3$ , donde  $\nu_i$  es el valor  $\nu$  de la minucia  $\mathbf{q}_i$ ,  $i = 1, 2, 3$ ; y  $\nu = 1$  si al recorrer la huella de arriba hacia abajo y de izquierda a derecha la minucia correspondiente es el inicio de una cresta o un surco, de lo contrario  $\nu = 2$ .
- $\lambda$  es el tamaño del mayor de los lados del triángulo.
- $\chi_i$  es la cantidad de minucias vecinas en una vecindad de  $32 \times 32$  centrada en  $\mathbf{q}_i$ ,  $i = 1, 2, 3$ .



- $\xi_1, \xi_2, \xi_3$  representan el conteo de crestas de los lados opuestos a las minucias  $\mathbf{q}_3, \mathbf{q}_1$  y  $\mathbf{q}_2$  respectivamente.

Dos tuplas son similares si las diferencias componente a componente son inferiores a umbrales. Esta forma de representación y similitud de tríos de minucias no toman en cuenta el ángulo de las minucias sino solo su dirección al recorrer la huella de arriba hacia abajo y de izquierda a derecha, siendo insensible a las direcciones de las minucias relativas a los lados.

Los algoritmos genéticos tienen un componente aleatorio por lo cual no se puede garantizar que dos ejecuciones del algoritmo obtengan el mismo resultado, de aquí que este método sea no determinista. Como reconocen los autores de este trabajo, la lentitud del algoritmo limita sus aplicaciones en problemas reales. En un trabajo más reciente (Ghazvini et al. 2011) se introdujeron algunas mejoras al algoritmo original pero con las mismas deficiencias básicas mencionadas anteriormente.

## Algoritmo FFCS

La representación propuesta por Feng *et al.* (Feng et al. 2006) es similar a la usada en el algoritmo JY, pero con la particularidad de que no usa el conteo de crestas y toma en cuenta los ángulos internos del triángulo. La función de comparación de tríos de minucias es también análoga a la usada en JY, de aquí que esta forma de representación y función de comparación no logran la invarianza al orden de las minucias en el rasgo. Similar a JY, para construir los tríos de minucias, este algoritmo asocia a cada minucia sus dos vecinas más cercanas lo que provoca que el algoritmo sea poco tolerante a los errores del extractor de rasgos.

A diferencia de JY, este algoritmo sí es insensible a la elección de una sola alineación. Para esto, el algoritmo considera como mejor alineación aquella que sea consistente con la mayor cantidad de tríos similares entre las huellas que se comparan.

## Algoritmo CTYZ

La representación propuesta por Chen *et al.* (Chen, Tian, Yang y Zhang 2006) toma en cuenta los siguientes elementos:

- Las longitudes de los lados del triángulo.
- Los ángulos internos del triángulo.

- Las diferencias de las orientaciones en una vecindad relativa a la minucia.
- Los ángulos formados por las direcciones de las minucias relativas a las bisectrices de los ángulos internos del triángulo.

El orden en esta tupla depende del orden en que se recorran las minucias en la huella. Para comparar dos tuplas proponen una medida difusa que toma en cuenta las diferencias componente a componente. Esta representación y función de comparación no son invariantes al orden de las minucias en el rasgo ni son sensibles a la reflexión de los tríos. Este algoritmo no realiza una comparación global de las minucias encontradas como coincidentes localmente.

Este algoritmo es lento en comparación con la mayoría de los basados en tríos de minucias, lo que se debe a que trabaja con todos los tríos de minucias cuyos lados estén en acotados por umbrales que son bien relajados, por ejemplo los umbrales mínimo y máximo usados para la competencia FVC2004 (Maio et al. 2004) son 10 y 150 respectivamente.

## Algoritmo XCF

La representación y función de comparación de tríos de minucias propuesta por Xu *et al.* (Xu et al. 2007) es similar a la de Jiang y Yau (Jiang y Yau 2000) pero no incluye el tipo de las minucias, el conteo de crestas, ni los ángulos de la minucia principal con respecto a los lados del trío. Esta representación tiene la particularidad de que incluye los ángulos internos del triángulo. La representación y función de comparación de tríos de minucias no son invariantes al orden de las minucias en el rasgo, no es sensible a la reflexión de los tríos, ni es sensible a las direcciones de las minucias relativas a los lados.

Los tríos usados en XCF son los obtenidos de la triangulación de Delaunay. Este tipo de triangulación es poco tolerante a los errores del extractor de rasgos.

XCF haya todos los posibles tríos similares entre las huellas que se comparan. Por cada par de tríos coincidentes, el algoritmo determina sus vecinos coincidentes cuyas transformaciones de traslación y rotación sean consistentes y este proceso lo repite hasta no poder expandir más la región. Finalmente fusiona todas las regiones coincidentes encontradas. El algoritmo básicamente realiza múltiples alineaciones, una por cada región que expande con lo cual logra la insensibilidad a la elección de una sola alineación.

## Algoritmo ZGZ

La representación propuesta por Zheng *et al.* (Zheng et al. 2009) ordena las minucias en el trío de acuerdo a la longitud de los lados. Dos tríos son similares si las diferencias de la razón

entre los lados correspondientes son inferiores a un umbral predeterminado y la diferencia de las direcciones relativas entre las minucias están en un umbral predeterminado. Esta representación y comparación de tríos de minucias no es sensible a la reflexión de los tríos ni a la dirección de las minucias relativas a los lados.

La alineación que realiza ZGZ es similar al de PN. ZGZ considera dos huellas similares si la cantidad de minucias coincidentes sobrepasan un umbral predeterminado.

## Algoritmo HK

Para este algoritmo Hoyle (Hoyle y Hsiao 2011) propone una representación para los tríos que incluye las siguientes características: el lado menor, la suma de los lados, la menor diferencia entre los ángulos de las minucias, la suma de las diferencias entre los ángulos de las minucias y el tipo de las minucias. Para considerar dos tríos como coincidentes, deben coincidir los tipos de las minucias mientras que las diferencias entre los demás componentes deben ser inferiores a umbrales predeterminados. Esta representación y función de comparación es invariante a la traslación y rotación, así como tolerante a las distorsiones no lineales de la huella; pero no es sensible a la reflexión de los tríos ni a las direcciones de las minucias relativas a los lados.

HK trabaja con todos los posibles tríos de minucias cuyas dimensiones de los lados estén en intervalos predeterminados. Primero el algoritmo compara todos los tríos de la huella consulta contra los de la huella plantilla. Luego verifica las coincidencias globales de los triángulos que fueron hallados similares, para esto construye y compara nuevos rasgos a partir de los pares y tríos de triángulos coincidentes localmente.

El algoritmo es costoso computacionalmente porque trabaja con muchos tríos, obtiene muchas comparaciones de tríos que son falsas positivas y por la cantidad de posibles estructuras que tiene que formar dinámicamente y comparar. Aunque este algoritmo fue creado para trabajar en el contexto de la identificación de huellas latentes, solo es probado con una base de datos de impresiones.

### 2.2.4 Algoritmos basados en micro características

Los algoritmos comparan las huellas representadas en términos de características microscópicas (también conocidas como características de tercer nivel (Maltoni et al. 2009)) como los poros, los contornos de las crestas y las crestas incipientes (Vatsa 2014, Anthonioz y Champod 2014, Angeloni y Marana 2013, Malathi y Meena 2011, Liu et al. 2011). Estos algoritmos son los más sensibles a la individualidad de las huellas, permitiendo establecer la

identidad de huellas incluso contando con fragmentos pequeños; por esto, en algunos contextos, presentan una eficacia superior a los basados en imágenes, los basados en características globales y los basados en características locales. Tienen la desventaja de que son los menos tolerantes a la baja calidad de las huellas puesto que las huellas deben ser adquiridas con sensores de alta resolución (más de 1000 ppp) y la mayoría de los sistemas y bases de datos existentes trabajan con huellas de 500 ppp. Por otra parte, estos algoritmos son costosos computacionalmente debido a la gran cantidad de micro características que deben ser procesadas durante la comparación.

### 2.2.5 Algoritmos híbridos

Estos algoritmos combinan varias formas de representación en un mismo algoritmo o combinan varios algoritmos ya sea con igual o diferentes formas de representación (Vatsa 2014, Babatunde et al. 2014, Liu, Arora, Cao, Jain y Lansing 2013, Babatunde et al. 2013, Marcialis et al. 2013, Garg y Rane 2013, Wahby Shalaby y Omair Ahmad 2013). Si bien estos algoritmos son los que alcanzan mayor eficacia, sus aplicaciones son restringidas al día de hoy por el costo computacional que presentan.

## 2.3 Síntesis y conclusiones

Como puede apreciarse en los epígrafes anteriores, los algoritmos basados en tríos de minucias poseen el siguiente conjunto de ventajas:

- Son tolerantes a las distorsiones locales presentes en las huellas (Jiang y Yau 2000).
- Tienen una elevada eficacia y rapidez en comparación con algoritmos basados en otras representaciones (Chen, Tian y Yang 2006, Parziale y Niel 2004), especialmente en aplicaciones con huellas parciales (Jea y Govindaraju 2005).
- Permiten una fácil interconexión con sistemas y dispositivos porque los estándares de almacenamiento más utilizados son los que usan solo minucias (Cappelli et al. 2010).
- Son apropiados para los sistemas embebidos en arquitecturas ligeras de hardware puesto que su representación y comparación se puede realizar con más eficiencia (Reisman et al. 2005).

A pesar de estas ventajas, el análisis de los diferentes algoritmos de la literatura basados en tríos de minucias revela que tienen los siguientes problemas de calidad (Cuadro 2.3):

Algoritmos	Limitaciones					
	I	II	III	IV	V	VI
JY	X			X	X	
KV		X	X	X		
PN		X			X	
JG1	X		X		X	X
JG2	X					X
RUR	X		X			
TB			X			X
FFCS	X				X	
CTYZ	X	X				
XCF	X	X	X		X	
ZGZ		X	X			
HK		X	X			
GSM			X			X

Cuadro 2.3: Resumen de problemas de calidad que degradan la eficacia de los algoritmos de comparación de huellas basados en tríos de minucias

- I Dependencia del orden de las minucias en el rasgo.
- II Insensibilidad a la reflexión de los tríos.
- III Ignoran las direcciones de las minucias relativas a los lados.
- IV Sensibilidad a la elección de una sola alineación basada en información local.
- V Sensibilidad a errores del extractor de rasgos que genera rasgos ausentes o incorrectos.
- VI No determinismo.

El siguiente capítulo propone una forma de representación y función de comparación de tríos de minucias invariante al orden de las minucias en los tríos para obtener todos los pares de tríos verdaderos coincidentes entre las huellas que se comparan. La nueva forma de representación es sensible a la reflexión de los tríos y a las direcciones de las minucias relativas a los lados para lograr así una mayor discriminación entre tríos que no son coincidentes. Para hallar los tríos de minucias se toma en cuenta cada minucia y varias de sus vecinas para lograr así que la comparación de las huellas sea robusta a los rasgos ausentes y a los errores del extractor de rasgos. Basada en la nueva forma de representación y función de comparación de tríos de minucias se proponen algoritmos de comparación de huellas que no tienen las limitaciones de los algoritmos en el Cuadro 2.3. Los nuevos algoritmos realizan múltiples alineaciones de las huellas que comparan sin discretizar el espacio de transformación y por

cada alineación hallan un cluster de coincidencias; algunos de los clusters encontrados son mezclados para encontrar todas las minucias coincidentes tolerando así las distorsiones no lineales presentes en las huellas. Los algoritmos que se proponen son deterministas así que dos ejecuciones con los mismos parámetros garantizan obtener los mismos resultados.

# 3

## Algoritmos de comparación de huellas basados en tríos de minucias

En este capítulo se introduce una nueva forma de representación y función de comparación de tríos de minucias que superan las deficiencias de las propuestas en la literatura. Se introducen reglas que permiten determinar tríos no coincidentes sin tener que comparar todos los rasgos. Se crean algoritmos de comparación basados en la nueva forma de representación y función de comparación que permiten tolerar las distorsiones no lineales tanto en impresiones como en huellas latentes. Téngase en cuenta que en la descripción de los algoritmos se han omitido algunos detalles que actualmente se encuentran en proceso de patentado.

### **3.1 Representación de los rasgos**

En esta sección se introduce una nueva forma de representación para los tríos de minucias. Primero se definen varios conceptos básicos para entender la nueva representación y luego se explican cada uno de los componentes de la nueva representación.

1. DEFINICIÓN. *Las minucias son los puntos donde la continuidad de las crestas se rompe.*

En el contexto de esta aplicación, las minucias se representan como  $(x, y, \theta)$ ; siendo  $(x, y)$  el punto donde se encuentra la minucia y  $\theta$ , la dirección de la cresta en ese punto.

**2. DEFINICIÓN.** El conjunto de todas las posibles minucias (universo  $U$ ) se define como  $U = \{(x, y, \theta) : (x, y, \theta) \in \mathbb{R}^3\}$ , donde  $\mathbb{R}$  es el conjunto de los números reales.

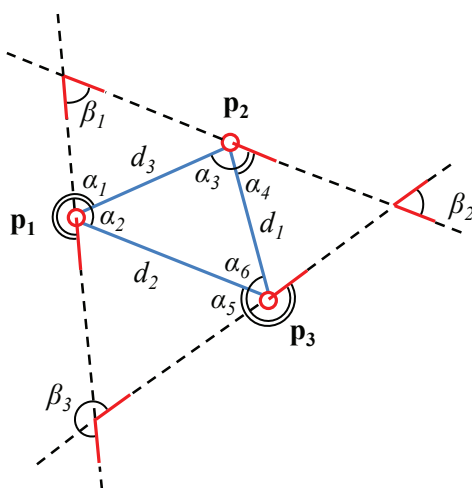


Figura 3.1: Los componentes de la nueva forma de representación de tríos de minucias.

**3. DEFINICIÓN.** Un  $m$ -triplet (Medina-Pérez et al. 2011) es una tupla  $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, d_1, d_2, d_3, d_{max}, d_{mid}, d_{min}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \beta_1, \beta_2, \beta_3)$  (ver la Figura 3.1); donde:

- Las minucias  $\mathbf{p}_1 \in U$ ,  $\mathbf{p}_2 \in U$  y  $\mathbf{p}_3 \in U$  son ordenadas de acuerdo al sentido de las manecillas del reloj comenzando por  $\mathbf{p}_1$ .
- $\mathbf{p}_1 = (x_1, y_1, \theta_1)$ ,  $\mathbf{p}_2 = (x_2, y_2, \theta_2)$ ,  $\mathbf{p}_3 = (x_3, y_3, \theta_3)$ .
- $d_1 = d_e(\mathbf{p}_2, \mathbf{p}_3)$ ,  $d_2 = d_e(\mathbf{p}_1, \mathbf{p}_3)$ ,  $d_3 = d_e(\mathbf{p}_1, \mathbf{p}_2)$ ; siendo  $d_e$  la función que calcula la distancia euclidiana entre dos minucias y se define como:

$$d_e(\mathbf{p}_j, \mathbf{p}_k) = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \quad (3.1)$$



siendo  $\mathbf{p}_j = (x_j, y_j, \theta_j)$  y  $\mathbf{p}_k = (x_k, y_k, \theta_k)$ .

- En el triángulo formado por las minucias del trío,  $d_{max}$ ,  $d_{mid}$  y  $d_{min}$  son las dimensiones de los lados mayor, medio y menor respectivamente. Esta información es usada para determinar si dos tríos no son coincidentes sin necesidad de comparar todos sus componentes.
- $\alpha_1 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_1, \mathbf{p}_2), \theta_1)$ ,  $\alpha_2 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_1, \mathbf{p}_3), \theta_1)$ ,  $\alpha_3 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_2, \mathbf{p}_1), \theta_2)$ ,  $\alpha_4 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_2, \mathbf{p}_3), \theta_2)$ ,  $\alpha_5 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_3, \mathbf{p}_1), \theta_3)$ ,  $\alpha_6 = \text{ad}_{2\pi}(\text{ang}(\mathbf{p}_3, \mathbf{p}_2), \theta_3)$ ;

siendo:

$$\text{ad}_{2\pi}(\alpha, \beta) = \begin{cases} \beta - \alpha & \text{si } \beta > \alpha \\ \beta - \alpha + 2\pi & \text{en caso contrario} \end{cases} \quad (3.2)$$

$$\text{ang}(\mathbf{p}_i, \mathbf{p}_j) = \begin{cases} \arctan(\Delta y / \Delta x) & \text{si } \Delta x > 0 \wedge \Delta y \geq 0 \\ \arctan(\Delta y / \Delta x) + 2\pi & \text{si } \Delta x > 0 \wedge \Delta y < 0 \\ \arctan(\Delta y / \Delta x) + \pi & \text{si } \Delta x < 0 \\ \pi/2 & \Delta x = 0 \wedge \Delta y > 0 \\ 3\pi/2 & \Delta x = 0 \wedge \Delta y < 0 \end{cases} \quad (3.3)$$

donde  $\mathbf{p}_i = (x_i, y_i, \theta_i)$ ,  $\mathbf{p}_j = (x_j, y_j, \theta_j)$ ,  $\Delta y = y_i - y_j$  y  $\Delta x = x_i - x_j$ .

- $\beta_1 = \text{ad}_{2\pi}(\theta_1, \theta_2)$ ,  $\beta_2 = \text{ad}_{2\pi}(\theta_2, \theta_3)$ ,  $\beta_3 = \text{ad}_{2\pi}(\theta_3, \theta_1)$  (ver 3.2).

Los m-triplets son sensibles a las direcciones de las minucias relativas a los lados (ángulos  $\alpha$ ) (Medina-Pérez et al. 2012b). Las minucias en esta representación son ordenadas de acuerdo al sentido de las manecillas del reloj; por tanto, para comparar los m-triplets, la función de similitud considera las tres rotaciones de las minucias en el sentido de las manecillas del reloj logrando así la invarianza al orden de las minucias en el rasgo y la sensibilidad a la reflexión de los tríos (Medina-Pérez et al. 2012b). La próxima sección contiene la definición formal de este procedimiento.

Los m-triplets no toman en cuenta el tipo de las minucias ni el conteo de crestas puesto que en experimentaciones realizadas se alcanzó mayor eficacia sin esta información. Esto coincide con varios autores que plantean que la estimación acertada del tipo de minucia

es un problema complicado y se puede alcanzar una eficacia más elevada en los algoritmos cuando se prescinde de esta información o cuando se le da un peso bajo (Tong et al. 2005, Jea y Govindaraju 2005, Yu et al. 2005). Por otra parte, la dependencia del conteo de crestas puede degradar sustancialmente la eficacia de los algoritmos puesto que su estimación es muy inexacta en áreas ruidosas, cerca de los puntos focales (Maltoni et al. 2009), y cuando el segmento sobre el cual se hará el conteo se encuentra parcialmente superpuesto sobre una cresta (Maltoni et al. 2009).

### 3.2 Función de similitud de m-triplets

En esta sección se propone una función de similitud de m-triplets que toma en cuenta las tres rotaciones de las minucias de acuerdo al sentido de las manecillas del reloj para evitar encontrar tríos reflejados como coincidentes. Los m-triplets se comparan componente a componente teniendo en cuenta umbrales de similitud; esta es una estrategia seguida por los trabajos previos sobre tríos de minucias (Jiang y Yau 2000, Kovács-Vajna 2000, Parziale y Niel 2004, Jea 2005, Jea y Govindaraju 2005, Reisman et al. 2005, Tan y Bhanu 2006, Feng et al. 2006, Chen, Tian, Yang y Zhang 2006, Xu et al. 2007, Zheng et al. 2009, Hoyle y Hsiao 2011) para tolerar las deformaciones implícitas en las huellas a nivel local. La función de comparación integra en un valor las similitudes entre los lados del triángulo, los ángulos de las minucias, las direcciones de las minucias relativas a los lados y los ángulos entre las direcciones de las minucias.

La siguiente ecuación compara dos m-triplets  $\mathbf{t}$  y  $\mathbf{r}$ :

$$s_s(\mathbf{t}, \mathbf{r}) = \max \left\{ s_b(\mathbf{t}, \mathbf{r}), s_b(\mathbf{t}, \text{shift}(\mathbf{r})), s_b(\mathbf{t}, \text{shift}(\text{shift}(\mathbf{r}))) \right\} \quad (3.4)$$

Donde  $\text{shift}(\mathbf{r})$  retorna el m-triplet  $\mathbf{r}$  rotado de acuerdo al sentido de las manecillas del reloj y se define como sigue:

$$\text{shift}((\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, d_1, d_2, d_3, d_{max}, d_{mid}, d_{min}, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \beta_1, \beta_2, \beta_3)) = (\mathbf{p}_3, \mathbf{p}_1, \mathbf{p}_2, d_3, d_1, d_2, d_{max}, d_{mid}, d_{min}, \alpha_5, \alpha_6, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_3, \beta_1, \beta_2) \quad (3.5)$$

Por su parte, la función  $s_b$  se define con la expresión:

$$s_b(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } s_\theta(\mathbf{t}, \mathbf{r}) = 0 \vee s_d(\mathbf{t}, \mathbf{r}) = 0 \vee s_\alpha(\mathbf{t}, \mathbf{r}) = 0 \vee s_\beta(\mathbf{t}, \mathbf{r}) = 0 \\ 1 - (1 - s_d(\mathbf{t}, \mathbf{r}))(1 - s_\alpha(\mathbf{t}, \mathbf{r}))(1 - s_\beta(\mathbf{t}, \mathbf{r})), & \text{en caso contrario} \end{cases} \quad (3.6)$$

La función 3.6 está compuesta por las funciones  $s_\theta$ ,  $s_d$ ,  $s_\alpha$  y  $s_\beta$ , que consideran las similitudes entre los ángulos de las minucias, los lados del triángulo, las direcciones de las minucias relativas a los lados y los ángulos entre las direcciones de las minucias respectivamente. De acuerdo con 3.6, dos m-triplets son totalmente distintos si difieren en al menos una función componente. Si todas las funciones componentes retornan un valor superior a cero, la regla del producto devuelve una alta similitud cuando al menos una función componente está cerca de 1. Nótese que la función  $s_\theta$  solo es tomada en cuenta para descartar tríos no similares puesto que  $s_\theta$  no toma valores entre 0 y 1 ya que esto significaría que dos huellas tendrían un valor de similitud que variaría con la rotación lo cual es falso.

La función  $s_\theta$  toma ventaja de la limitada rotación presente en los problemas de verificación (Medina-Pérez et al. 2009). Se incorporó esta información en la función de similitud de m-triplets para incrementar la discriminación de minucias en este tipo de problemas. De esta manera, dos m-triplets son diferentes si al menos una de las diferencias de los ángulos de las minucias es superior a  $\pi/4$  (Medina-Pérez et al. 2011). Esta función toma valor 1 si las direcciones de las minucias comparadas entre los tríos difieren no más de  $\pi/4$  y toma valor 0 si las direcciones de al menos un par de minucias comparadas difieren más de  $\pi/4$ . La función  $s_\theta$  se define como sigue:

$$s_\theta(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } \exists i = 1 \dots 3 (\text{ad}_\pi(\theta_i^{\mathbf{t}}, \theta_i^{\mathbf{r}}) > \pi/4) \\ 1 & \text{en caso contrario} \end{cases} \quad (3.7)$$

donde las minucias  $\mathbf{p}_i^{\mathbf{t}} = (x_i^{\mathbf{t}}, x_i^{\mathbf{t}}, \theta_i^{\mathbf{t}})$  ( $i = 1 \dots 3$ ) forman el m-triplet  $\mathbf{t}$  y las minucias  $\mathbf{p}_i^{\mathbf{r}} = (x_i^{\mathbf{r}}, x_i^{\mathbf{r}}, \theta_i^{\mathbf{r}})$  ( $i = 1 \dots 3$ ) forman el m-triplet  $\mathbf{r}$ ; mientras que la función  $\text{ad}_\pi$  calcula la diferencia entre las direcciones de las minucias y se define como sigue:

$$\text{ad}_\pi(\theta_i^{\mathbf{t}}, \theta_i^{\mathbf{r}}) = \text{mín} \{ |\theta_i^{\mathbf{t}} - \theta_i^{\mathbf{r}}|, 2\pi - |\theta_i^{\mathbf{t}} - \theta_i^{\mathbf{r}}| \} \quad (3.8)$$

La función  $s_d$  compara m-triplets a partir de la longitud de los lados del triángulo formado por las minucias y se define por:

$$s_d(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } \exists i = 1 \dots 3 (|d_i^{\mathbf{t}} - d_i^{\mathbf{r}}| > t_d) \\ 1 - \text{máx}_{i=1 \dots 3} \{ |d_i^{\mathbf{t}} - d_i^{\mathbf{r}}| \} / t_d & \text{en caso contrario} \end{cases} \quad (3.9)$$

siendo  $d_i^{\mathbf{t}}$  y  $d_i^{\mathbf{r}}$  ( $i = 1 \dots 3$ ) los valores de las distancias en los m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  respectivamente; mientras que  $t_d = 15$  es el umbral reportado en (Pankanti et al. 2002) para hacer coincidir el 97.5% de las minucias de acuerdo a la distancia euclidiana en huellas con resolución de 500 ppp. La función 3.9 retorna 0 si al menos la diferencia entre un par de lados es mayor que el

umbral  $t_d$ . Devuelve 1 si las tres diferencias de lados son 0; esto es, los triángulos formados por ambos m-triplets son iguales.

La función  $s_\alpha$  compara m-triplets basada en los ángulos formados por las direcciones de las minucias relativas a los lados de los triángulos (vea los ángulos  $\alpha$  en la Figura 3.1) y se define:

$$s_\alpha(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } \exists i = 1 \dots 6 (\text{ad}_\pi(\alpha_i^{\mathbf{t}}, \alpha_i^{\mathbf{r}}) > t_a) \\ 1 - \text{máx}_{i=1\dots 6} \{\text{ad}_\pi(\alpha_i^{\mathbf{t}}, \alpha_i^{\mathbf{r}})\} / t_a & \text{en caso contrario} \end{cases} \quad (3.10)$$

siendo  $\alpha_i^{\mathbf{t}}$  y  $\alpha_i^{\mathbf{r}}$  ( $i = 1 \dots 6$ ) los valores de las direcciones de las minucias relativas a los lados de los triángulos en los m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  respectivamente; mientras que  $t_a = \pi/8$  es el umbral reportado en (Pankanti et al. 2002) para hacer coincidir el 97.5 % de las minucias de acuerdo a la diferencia de sus direcciones en huellas con resolución de 500 ppp.

La función  $s_\beta$  compara m-triplets basada en las direcciones relativas entre las minucias (ver los ángulos  $\beta$  en la Figura 3.1):

$$s_\beta(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } \exists i = 1 \dots 3 (\text{ad}_\pi(\beta_i^{\mathbf{t}}, \beta_i^{\mathbf{r}}) > t_a) \\ 1 - \text{máx}_{i=1\dots 3} \{\text{ad}_\pi(\beta_i^{\mathbf{t}}, \beta_i^{\mathbf{r}})\} / t_a & \text{en caso contrario} \end{cases} \quad (3.11)$$

siendo  $\beta_i^{\mathbf{t}}$  y  $\beta_i^{\mathbf{r}}$  ( $i = 1 \dots 3$ ) las direcciones relativas entre las minucias en los m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  respectivamente.

Las funciones 3.10 y 3.11 retornan 0 si al menos dos de los ángulos comparados difieren más que el umbral  $t_a$ . Mientras menos difieren los ángulos, mayor será el valor retornado por las ecuaciones; así retornan 1 si los ángulos comparados son iguales.

Para evitar comparar todos los componentes de tríos no similares se proponen los siguientes teoremas:

**3.1. TEOREMA.** *Dados dos m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  cuyas dimensiones de los lados mayores son*

*$d_{max}^{\mathbf{t}}$  y  $d_{max}^{\mathbf{r}}$  respectivamente, si  $|d_{max}^{\mathbf{t}} - d_{max}^{\mathbf{r}}| > t_d$  entonces  $s_s(\mathbf{t}, \mathbf{r}) = 0$ .*

**3.2. TEOREMA.** *Dados dos m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  cuyas dimensiones de los lados medios son  $d_{mid}^{\mathbf{t}}$*

*y  $d_{mid}^{\mathbf{r}}$  respectivamente, si  $|d_{mid}^{\mathbf{t}} - d_{mid}^{\mathbf{r}}| > t_d$  entonces  $s_s(\mathbf{t}, \mathbf{r}) = 0$ .*

**3.3. TEOREMA.** *Dados dos m-triplets  $\mathbf{t}$  y  $\mathbf{r}$  cuyas dimensiones de los lados menores son*

*$d_{min}^{\mathbf{t}}$  y  $d_{min}^{\mathbf{r}}$  respectivamente, si  $|d_{min}^{\mathbf{t}} - d_{min}^{\mathbf{r}}| > t_d$  entonces  $s_s(\mathbf{t}, \mathbf{r}) = 0$ .*

Las pruebas de estos teoremas aparecen en las siguientes secciones. Obviamos la prueba del Teorema 3.3 porque es muy similar a la del Teorema 3.1.

Basada en estos teoremas se modifica la función 3.4 para detectar si  $\mathbf{t}$  y  $\mathbf{r}$  no son similares sin evaluar  $s_b(\mathbf{t}, \mathbf{r})$ ,  $s_b(\mathbf{t}, \text{shift}(\mathbf{r}))$  y  $s_b(\mathbf{t}, \text{shift}(\text{shift}(\mathbf{r})))$ ; reduciendo así el tiempo de cálculo de la similitud. Finalmente, la función de comparación de m-triplets se define como sigue:

$$s_t(\mathbf{t}, \mathbf{r}) = \begin{cases} 0 & \text{si } (|d_{max}^{\mathbf{t}} - d_{max}^{\mathbf{r}}| > t_d) \vee (|d_{mid}^{\mathbf{t}} - d_{mid}^{\mathbf{r}}| > t_d) \vee (|d_{min}^{\mathbf{t}} - d_{min}^{\mathbf{r}}| > t_d) \\ \text{máx} \left\{ s_b(\mathbf{t}, \mathbf{r}), s_b(\mathbf{t}, \text{shift}(\mathbf{r})), s_b(\mathbf{t}, \text{shift}(\text{shift}(\mathbf{r}))) \right\} & \text{en caso contrario} \end{cases} \quad (3.12)$$

## Prueba del Teorema 3.1

De la definición de m-triplets se infiere que:

$$d_{max}^{\mathbf{r}} \geq d_{mid}^{\mathbf{r}}, \quad (3.13)$$

$$d_{max}^{\mathbf{r}} \geq d_{min}^{\mathbf{r}}, \quad (3.14)$$

$$d_{mid}^{\mathbf{r}} \geq d_{min}^{\mathbf{r}}. \quad (3.15)$$

Asúmase sin pérdida de generalidad que  $d_{max}^{\mathbf{t}} > d_{max}^{\mathbf{r}}$ ; entonces de la hipótesis del teorema 3.1 se infiere que:

$$d_{max}^{\mathbf{t}} - d_{max}^{\mathbf{r}} > t_d, \quad (3.16)$$

que es equivalente a:

$$d_{max}^{\mathbf{t}} - t_d > d_{max}^{\mathbf{r}}. \quad (3.17)$$

A partir de 3.17 y 3.13 se infiere que  $d_{max}^{\mathbf{t}} - t_d > d_{mid}^{\mathbf{r}}$ ; lo cual equivale a:

$$d_{max}^{\mathbf{t}} - d_{mid}^{\mathbf{r}} > t_d. \quad (3.18)$$

A partir de 3.17 y 3.14 se infiere que  $d_{max}^{\mathbf{t}} - t_d > d_{min}^{\mathbf{r}}$ ; lo cual equivale a:

$$d_{max}^{\mathbf{t}} - d_{min}^{\mathbf{r}} > t_d. \quad (3.19)$$

Las expresiones 3.16, 3.18 y 3.19 muestran que  $d_{max}^{\mathbf{t}}$  difieren más que el umbral  $t_d$  con

respecto a  $d_{max}^r$ ,  $d_{mid}^r$  y  $d_{min}^r$  respectivamente. Por tanto,  $s_d(\mathbf{t}, \mathbf{r})$  retorna 0 por cada rotación de  $\mathbf{r}$  de acuerdo al sentido de las manecillas del reloj y consecuentemente  $s_s(\mathbf{t}, \mathbf{r})$  también retorna 0.

## Prueba del Teorema 3.2

De la definición de m-triplets se infiere que:

$$d_{max}^t \geq d_{mid}^t. \quad (3.20)$$

Asúmase sin pérdida de generalidad que  $d_{mid}^t > d_{mid}^r$ ; entonces de la hipótesis del teorema 3.2 se infiere que:

$$d_{mid}^t - d_{mid}^r > t_d, \quad (3.21)$$

que es equivalente a:

$$d_{mid}^t - t_d > d_{mid}^r. \quad (3.22)$$

A partir de 3.22 y 3.15 se infiere que  $d_{mid}^t - t_d > d_{min}^r$ ; lo cual equivale a:

$$d_{mid}^t - d_{min}^r > t_d. \quad (3.23)$$

A partir de 3.20 y 3.21 se infiere que

$$d_{max}^t - d_{mid}^r > t_d. \quad (3.24)$$

A partir de 3.20 y 3.23 se infiere que

$$d_{max}^t - d_{min}^r > t_d. \quad (3.25)$$

Las expresiones 3.21, 3.23, 3.24 y 3.25 muestran que  $d_{mid}^t$  y  $d_{max}^t$  difiere más que el umbral  $t_d$  con respecto a  $d_{mid}^r$  y  $d_{min}^r$ . Por tanto,  $s_d(\mathbf{t}, \mathbf{r})$  retorna 0 por cada rotación de  $\mathbf{r}$  de acuerdo al sentido de las manecillas del reloj y consecuentemente  $s_s(\mathbf{t}, \mathbf{r})$  también retorna 0.

### 3.3 Cálculo de los tríos de minucias

Dada una huella descrita por el conjunto de minucias  $I$ , los m-triplets se calculan como sigue. Por cada  $\mathbf{p} \in I$ , hallar sus 7 vecinos más cercanos en  $I$  y construir todos los m-triplets que incluyen a  $\mathbf{p}$  y dos de sus 7 vecinos más cercanos descartando los tríos duplicados. Esta estrategia permite la tolerancia a los rasgos ausentes y a los errores del extractor de rasgos (Medina-Pérez et al. 2012b). Si bien el uso de un número de vecinos inferior a 7 aumenta la eficiencia del algoritmo, pruebas empíricas en las bases de entrenamiento de las competencias FVC2002 (Maio et al. 2002), FVC2004 (Maio et al. 2004) y FVC2006 (Cappelli et al. 2007) mostraron el mejor balance entre eficacia y eficiencia para el valor 7.

Adicionalmente, los m-triplets en las huellas son ordenados de acuerdo a la longitud del lado mayor ( $d_{max}$ ) para realizar búsqueda binaria cuando se requiera encontrar los tríos similares a un m-triplet dado (el Teorema 3.1 garantiza la exactitud de este procedimiento).

### 3.4 Algoritmos de comparación basados en múltiples clusters de coincidencias

Si bien la forma de representación y función de comparación de tríos de minucias propuestas son tolerantes a las distorsiones de las huellas a nivel local, se requiere de otra estrategia para tolerar las distorsiones a nivel global en las huellas puesto que los tríos de minucias solo abarcan información local de las imágenes. La creación de algoritmos tolerantes a las distorsiones globales en las huellas constituye uno de los retos más importantes en la temática actualmente (Paulino, Feng y Jain 2013, Jain y Feng 2011). Este tipo de distorsiones provocan la aparición de múltiples clusters de coincidencias pero este problema no ha sido tratado adecuadamente en los algoritmos del estado del arte. En esta investigación un cluster de coincidencias es un conjunto de pares de minucias de dos huellas que cumplen con ciertas restricciones geométricas. Esta es la principal motivación para crear nuevos algoritmos cuyo principio de funcionamiento principal se resume como: las comparaciones de huellas verdaderamente coincidentes (*comparaciones genuinas*) tienen más clusters de minucias coincidentes que el número de clusters que se encuentran en las comparaciones de huellas falsas coincidentes (*comparaciones impostoras*).

La Figura 3.2 muestra un ejemplo de las distorsiones mencionadas anteriormente. El ejemplo muestra gráficamente la diferencia de la distancia entre minucias pertenecientes a diferentes clusters. La huella de la derecha muestra con flechas de color gris la diferencia con respecto a las distancias en la huella izquierda. Las flechas de color gris evidencian una

distorsión en la huella derecha hasta de 42 píxeles. Nótese que esta distancia es más del doble necesitado (15 píxeles) para hacer coincidir el 97.5 % de las minucias de acuerdo a (Pankanti et al. 2002). Así, si se usara un umbral de distancia de 42 píxeles en lugar de 15, aumentaría la cantidad de minucias falsas coincidentes que se encontrarían al comparar huellas de dedos diferentes. Para tolerar estas distorsiones, los algoritmos propuestos en esta sección usan un valor pequeño de umbral para hallar múltiples clusters de minucias coincidentes y luego verifican varias condiciones para determinar cuáles clusters mezclar.

Los nuevos algoritmos se basan en una definición de cluster cuya esencia consiste en agrupar pares de minucias coincidentes encontrados al realizar una alineación. La nueva definición de cluster se formaliza como sigue:

4. DEFINICIÓN. Sean dos conjuntos de minucias  $L = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$  e  $I = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$ , un “cluster de minucias coincidentes” (o simplemente “cluster”) es un conjunto de pares de minucias  $C = \{(\mathbf{q}_{j,1}, \mathbf{p}_{i,1}), (\mathbf{q}_{j,2}, \mathbf{p}_{i,2}), \dots, (\mathbf{q}_{j,l}, \mathbf{p}_{i,l})\}$  con centroide  $r_C = (\mathbf{q}_{j,h}, \mathbf{p}_{i,h}) \in C$  que cumple las siguientes condiciones para cada par  $(\mathbf{q}_{j,k}, \mathbf{p}_{i,k}) \in C$ :

- $\mathbf{q}_{j,k} \in L \wedge \mathbf{p}_{i,k} \in I$ .
- $\nexists (\mathbf{q}_{j,g}, \mathbf{p}_{i,g}) \in C [(\mathbf{q}_{j,k} = \mathbf{q}_{j,g} \wedge \mathbf{p}_{i,k} \neq \mathbf{p}_{i,g}) \vee (\mathbf{q}_{j,k} \neq \mathbf{q}_{j,g} \wedge \mathbf{p}_{i,k} = \mathbf{p}_{i,g})]$ .
- $\sigma_e(\psi(\mathbf{q}_{j,k}, \mathbf{q}_{j,h}, \mathbf{p}_{i,h}), \mathbf{p}_{i,k}) > 0 \wedge \sigma_\theta(\psi(\mathbf{q}_{j,k}, \mathbf{q}_{j,h}, \mathbf{p}_{i,h}), \mathbf{p}_{i,k}) > 0$  donde:
  - La función  $\psi$  transforma una minucia  $\mathbf{a}$  en otra  $\mathbf{d}$  usando como referencia un par de minucias  $(\mathbf{b}, \mathbf{c})$  y se define como:

$$\begin{aligned} \psi : U \times U \times U &\rightarrow U \\ (\mathbf{a}, \mathbf{b}, \mathbf{c}) &\rightarrow \mathbf{d} \end{aligned} \tag{3.26}$$

- La función  $\sigma_e$  calcula la similitud entre dos minucias de acuerdo a la distancia entre ellas y se define de la siguiente manera:

$$\begin{aligned} \sigma_e : U \times U &\rightarrow [0; 1] \\ (\mathbf{a}, \mathbf{b}) &\rightarrow d \end{aligned} \tag{3.27}$$



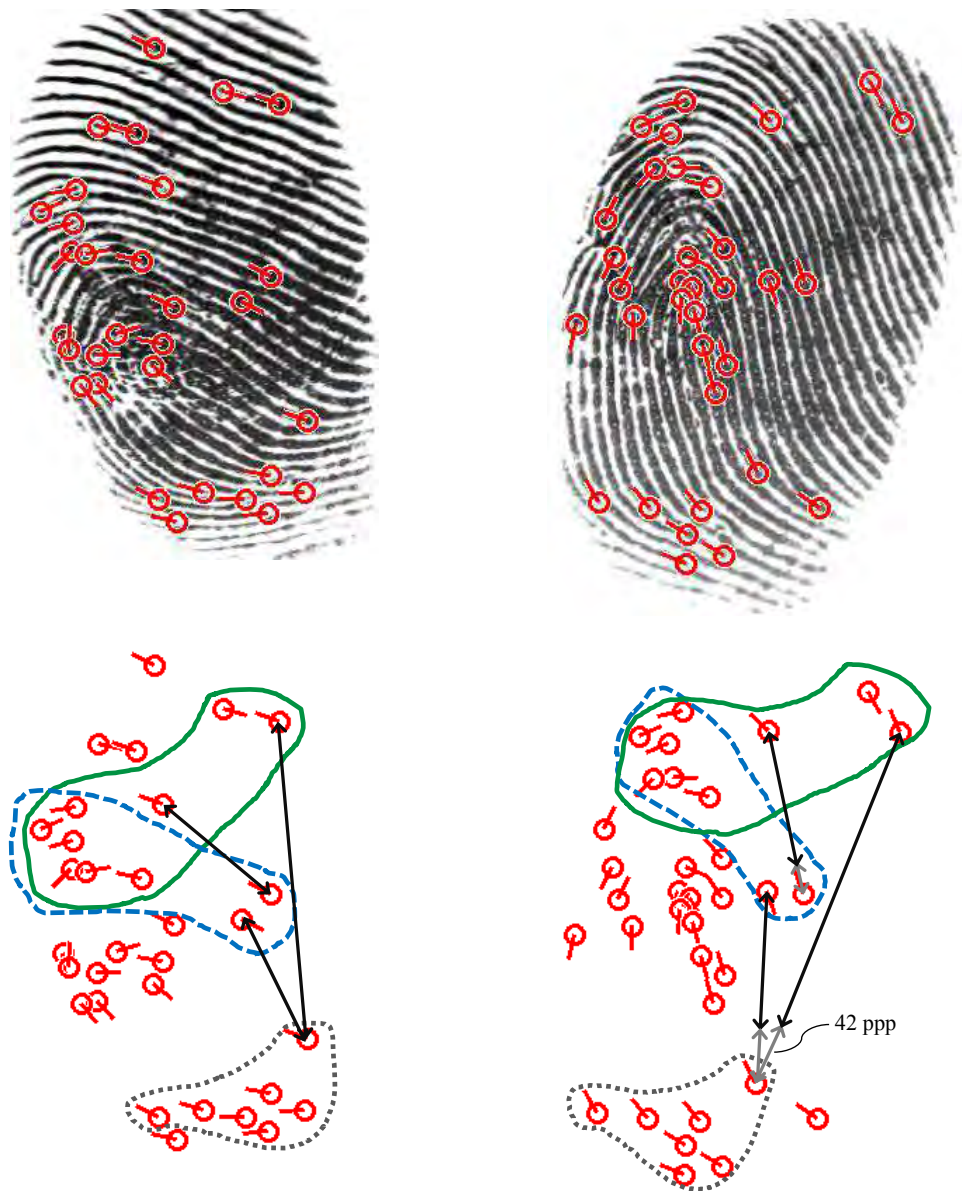


Figura 3.2: Dos impresiones (adquiridas con escáner a 500 ppp) con tres clusters de coincidencias encontrados con el algoritmo propuesto. Las dos imágenes inferiores solo muestran las minucias de las huellas de encima para ilustrar con mayor claridad los clusters. Las minucias en la huella de la derecha se encuentran más lejos que en la izquierda debido a las distorsiones no lineales presentes en las huellas. La mayoría de las minucias que no pertenecen a ningún cluster son minucias falsas encontradas erróneamente por el extractor de rasgos.

- La función  $\sigma_\theta$  calcula la similitud entre dos minucias de acuerdo a la diferencia

entre las direcciones de las minucias y se define de la siguiente manera:

$$\begin{aligned} \sigma_\theta : U \times U &\rightarrow [0; 1] \\ (\mathbf{a}, \mathbf{b}) &\rightarrow d \end{aligned} \quad (3.28)$$

La Figura 3.3 muestra un diagrama general de los algoritmos que se describen a continuación.

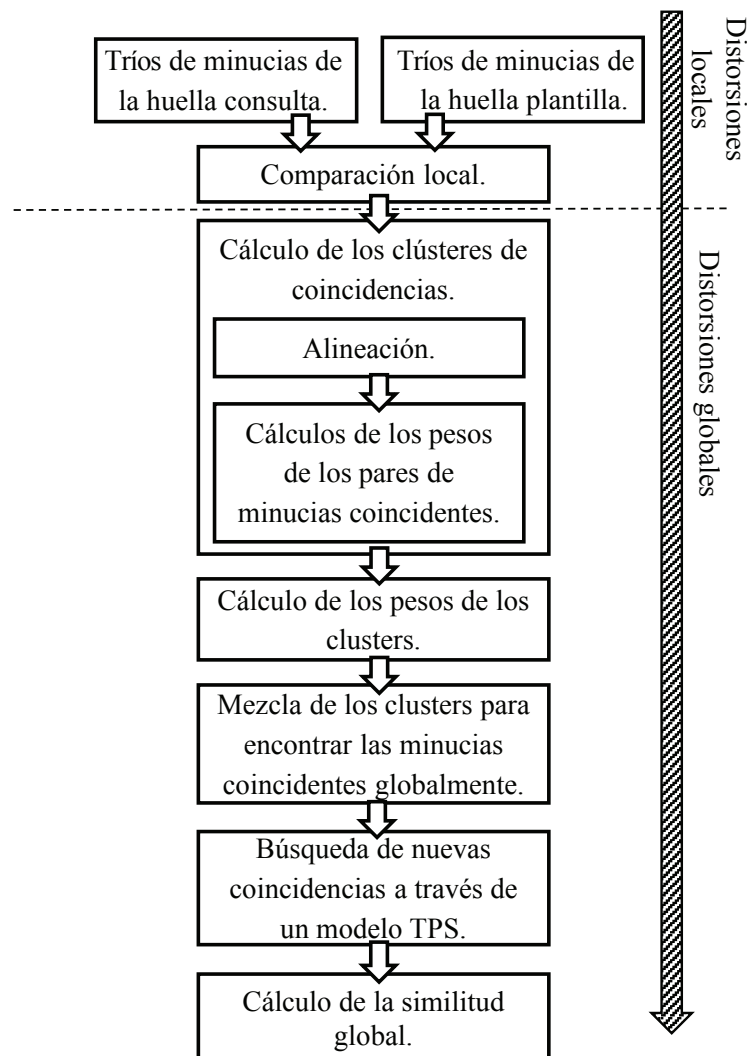


Figura 3.3: Esquema general de los algoritmos propuestos.

Para hallar las coincidencias entre las huellas a nivel local primero por cada m-triplet de la huella consulta se realiza búsqueda binaria para encontrar los m-triplets similares de la

huella plantilla. Luego se utiliza una estrategia ávida (Cormen et al. 2009) para determinar los pares de minucias coincidentes localmente a partir de los tríos similares entre las dos huellas.

Por cada par de minucias coincidentes localmente se realiza una alineación de las huellas y se halla un cluster de minucias coincidentes; esta es una estrategia que ha mostrado que eleva la eficacia de algoritmos basados en otros descriptores de minucias (Medina-Pérez et al. 2009, Medina-Pérez et al. 2012a). El par de minucias sobre el cual se realiza la alineación es considerado el centroide del cluster encontrado. El peso de cada centroide se calcula de tres maneras diferentes dando lugar a algoritmos diferentes en dependencia del problema en el cual se aplicarán (verificación de huellas dactilares, verificación de huellas palmares e identificación de huellas latentes dactilares y palmares) y en los tres casos se toma en cuenta la posición y dirección relativa entre las minucias de cada par en el cluster. Un centroide tiene alto valor de peso cuando su cluster contiene muchos pares de minucias coincidentes y baja deformación de las huellas. Este peso se calcula de tres formas diferentes dando lugar a 3 algoritmos para cada uno de los siguientes problemas respectivamente:

- Verificación de huellas dactilares.
- Verificación de huellas palmares.
- Identificación de huellas dactilares y palmares latentes.

El peso de cada cluster se calcula sumando el peso de los pares de minucias contenidos en dicho cluster. El impacto de las grandes distorsiones no lineales se reduce tomando en cuenta las relaciones globales entre los clusters de minucias coincidentes. Un subconjunto de los clusters es mezclado de mayor a menor peso de los clusters para encontrar la mayoría de los pares de minucias coincidentes. Los detalles de estos pasos se omiten porque actualmente están siendo patentados.

Un modelo Thin Plate Spline (TPS) (Bazen y Gerez 2003) es creado a partir de los pares de minucias coincidentes encontrados hasta este punto. TPS es un tipo de Spline que modela las superficies como si fueran planos elásticos sobre los cuales se ejercen fuerzas en diferentes direcciones. El uso de TPS permite crear un modelo de las deformaciones de las huellas y nuevas coincidencias se pueden encontrar en la medida que mejor se adapten al modelo creado. Por cada una de las nuevas coincidencias encontradas se alinean las huellas, se calcula el cluster de coincidencias y se calcula el peso del centroide.

El valor final de similitud es la suma de los pesos de los pares de minucias coincidentes encontrados como resultado de la mezcla de clusters y del uso del modelo TPS construido.

La Figura 3.4 muestra un ejemplo del impacto en el algoritmo propuesto para verificación de huellas dactilares si tuviera las mismas deficiencias de los algoritmos basados en tríos de minucias en la literatura. La Figura 3.4.a muestra el resultado del algoritmo original

propuesto en esta investigación al comparar dos huellas del mismo dedo. La Figura 3.4.b muestra el resultado del algoritmo al ordenar las minucias en los tríos por los valores de distancia en lugar de ordenarlas de acuerdo al sentido de las manecillas del reloj, siendo el algoritmo dependiente del orden de las minucias en el rasgo. La Figura 3.4.c muestra el resultado del algoritmo al comparar los lados de los tríos sin tener ningún orden establecido, siendo el algoritmo insensible a la reflexión de los tríos. La Figura 3.4.d muestra el resultado del algoritmo al ignorar las direcciones de las minucias relativas a los lados; esto es, al no tener en cuenta los ángulos  $\alpha$  en la representación propuesta de tríos de minucias. La Figura 3.4.e muestra el resultado del algoritmo al seleccionar solo el cluster de mayor peso en lugar de todos los clusters encontrados, siendo el algoritmo sensible a la elección de una sola alineación basada en información local. La Figura 3.4.f muestra el resultado del algoritmo al usar la triangulación de Dalaunay (Parziale y Niel 2004) la cual halla menos tríos que la propuesta de esta investigación, siendo el algoritmo sensible a errores del extractor de rasgos que genera rasgos ausentes o incorrectos. La Figura 3.4.g muestra el resultado del algoritmo al mezclar los clusters de manera aleatoria en lugar de ordenarlos por su peso, siendo el algoritmo no determinista. Nótese que cuando el algoritmo es modificado teniendo en cuenta cada una de las deficiencias identificadas en la literatura se obtienen menos minucias coincidentes que las obtenidas por el algoritmo sin alteraciones.

### 3.5 Síntesis y conclusiones.

En este capítulo se proponen nuevos algoritmos para la comparación de huellas cuyas contribuciones más distintivas del resto de la literatura se resumen como sigue:

- Los algoritmos se basan en una nueva forma de representación de tríos de minucias donde las minucias se ordenan de acuerdo al sentido de las manecillas del reloj y donde se tiene en cuenta la distancia entre las minucias, las diferencias entre las direcciones de las minucias y las direcciones de las minucias relativas a los lados del triángulo formado por las minucias (Medina-Pérez et al. 2011). Al ser estos rasgos relativos a las minucias del trío, la forma de representación es invariante a la traslación y a la rotación de las huellas (Medina-Pérez et al. 2012b).
- Los tríos de minucias se calculan a partir de cada minucia y sus siete vecinos más cercanos para evitar los problemas ocasionados por los rasgos ausentes y por los errores del extractor de rasgos. El número siete fue estimado empíricamente en las bases de datos de entrenamiento de las competencias FVC2002 (Maio et al. 2002), FVC2004 (Maio et al. 2002) y FVC2006 (Cappelli et al. 2007).

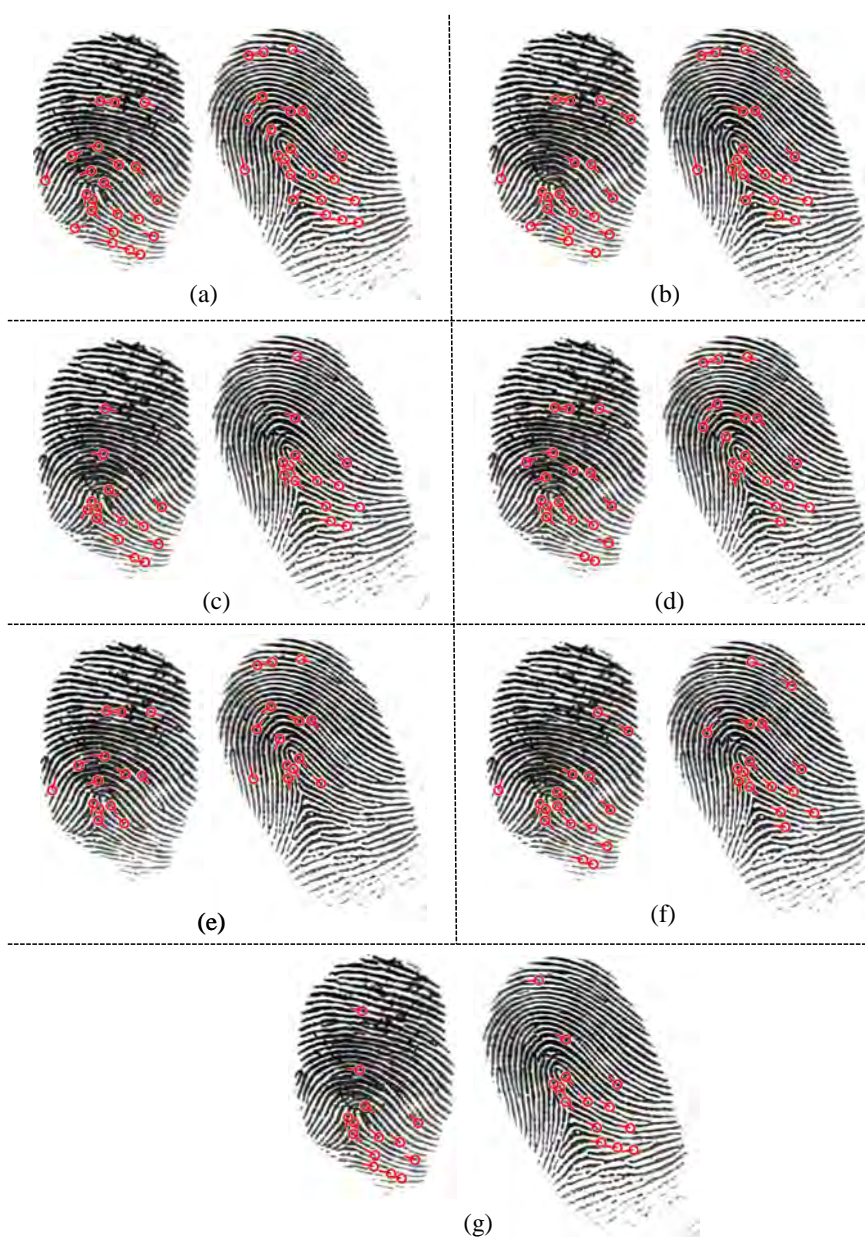


Figura 3.4: Resultados (a) del algoritmo propuesto para verificación de huellas dactilares sin variaciones y del algoritmo modificado con las mismas limitaciones identificadas en la literatura: (b) siendo dependiente del orden de las minucias en el rasgo; (c) siendo insensible a la reflexión de los tríos; (d) ignorando las direcciones de las minucias relativas a los lados; (e) eligiendo una sola alineación basada en información local; (f) siendo sensible a errores del extractor de rasgos que genera rasgos ausentes o incorrectos; (g) siendo no determinista.

- Para comparar los rasgos se propone una nueva función que rota tres veces los tríos de acuerdo al sentido de las manecillas del reloj para seleccionar la mejor rotación (Medina-

Pérez et al. 2011). De esta forma se logra la invarianza al orden de las minucias en el rasgo y la sensibilidad a la reflexión de los tríos (Medina-Pérez et al. 2012b). La función de comparación de tríos usa umbrales de tolerancia para comparar los rasgos de los tríos logrando así la tolerancia a las distorsiones no lineales presentes a nivel local en las huellas (Medina-Pérez et al. 2012b). Esta función permite descartar tríos no similares sin necesidad de comparar todos sus rasgos (Medina-Pérez et al. 2012b).

- Los pares de minucias coincidentes localmente se hayan a partir de un algoritmo ávido que busca seleccionar la menor cantidad de pares coincidentes con los valores de similitud más altos mientras que intenta minimizar la repetición de minucias en los pares seleccionados. Esta estrategia ha sido exitosamente usada en el contexto de diversos descriptores de minucias (Medina-Pérez et al. 2009, Medina-Pérez et al. 2012a).
- Para comparar las huellas se realizan múltiples alineaciones y por cada alineación se halla un cluster de coincidencias. Para esto se propone una nueva definición de cluster que no depende de la conectividad entre los tríos de minucias que lo componen para lograr una mayor tolerancia a las minucias ausentes y los errores del extractor de rasgos.
- Para alinear las huellas no se discretiza el espacio de transformación de las minucias y con esto se evita perder información.
- Se realiza un número dinámico de alineaciones para hallar clusters de coincidencias en toda el área común de las huellas que se comparan.
- Para tolerar las distorsiones no lineales a nivel global en las huellas se mezclan un subconjunto de los clusters encontrados verificando determinadas restricciones geométricas entre los centroides de clusters y comprobando que no haya pares de minucias coincidentes diferentes compartiendo una minucia común. Esta heurística tiene como objetivo obtener muchos pares de minucias verdaderamente coincidentes manteniendo baja la cantidad de falsas coincidencias.
- Dos pares de minucias con similares características a nivel local, no necesariamente tienen el mismo peso porque los nuevos algoritmos también toman en cuenta las características globales para calcular dicho peso. Más aún, para la identificación de huellas latentes se propone una forma de calcular el peso diferente a la usada para la verificación de huellas dactilares y a la usada para la verificación de huellas palmares, dando lugar así a tres algoritmos de comparación:
  - M3gl2: Es la versión del algoritmo cuya función para calcular el peso de los pares de minucias coincidentes se optimizó para la verificación de huellas dactilares.

- M3gl2.5: Es la versión del algoritmo cuya función para calcular el peso de los pares de minucias coincidentes se optimizó para la verificación de huellas palmares.
- M3gl3: Es la versión del algoritmo cuya función para calcular el peso de los pares de minucias coincidentes se optimizó para la identificación de huellas latentes dactilares y palmares.

Las principales deficiencias de los algoritmos propuestos son:

- Los tríos de minucias son menos discriminantes que las representaciones que toman en consideración más información local como los Cylinder Codes (Cappelli et al. 2010). Es válido aclarar que el uso de representaciones como los Cylinder Codes permiten mejorar la eficacia pero sus aplicaciones son más limitadas porque requieren de mayor espacio en memoria.
- La eficacia de los algoritmos pudiera incrementarse si se usaran rasgos globales de la huella como las imágenes de orientación (Maltoni et al. 2009) adicionales a los tríos de minucias; sin embargo, estas técnicas están fuera del marco de esta investigación.
- Al ser basados en minucias, la eficacia de los algoritmos propuestos se afecta con la disminución de la cantidad de minucias.





# 4

## Resultados experimentales en la verificación de huellas

En este capítulo se prueba la eficacia de los algoritmos propuestos en el contexto de la verificación de huellas. La eficacia para la verificación de huellas se prueba en 15 bases de datos de 4 competencias internacionales.

### 4.1 Protocolos de evaluación

En esta investigación se utilizan los protocolos de evaluación de las competencias internacionales de verificación de huellas FVC2002 (Maio et al. 2002), FVC2004 (Maio et al. 2004) y FVC2006 (Cappelli et al. 2007) para determinar la sensibilidad de los algoritmos a la individualidad de las huellas. A continuación se hace una breve descripción de estos protocolos y los detalles pueden encontrarse en (Cappelli et al. 2006).

Cada una de estas competencias contiene 4 bases de datos de prueba y 4 de entrenamiento. Cada base de datos de prueba de las competencias FVC2002 y FVC2004 contiene 8 impresiones de 100 dedos; mientras que cada base de datos de prueba de la competencia FVC2006 contiene 12 impresiones de 140 dedos. Para determinar la eficacia de cada algoritmo, se compara cada huella con aquellas que son del mismo dedo (comparaciones genuinas)

mientras que la primera huella de cada dedo también se compara con la primera huella de los dedos diferentes (comparaciones impostoras). A partir de estas comparaciones se determinan los siguientes indicadores de eficacia y eficiencia:

- *Curva ROC* (Fawcett 2006) muestra qué tantas comparaciones falsas no coincidentes (*FNMR* por las siglas en inglés de “False NonMatch Rate”) comete un algoritmo para las diferentes cantidades de comparaciones falsas coincidentes (*FMR* por las siglas en inglés de “False Match Rate”). Esta curva se representa en escala logarítmica para que la mayor área sea la ocupada por el intervalo de 0 a 0.1 de las comparaciones. Mientras más cerca esté la curva del origen de coordenadas mejor será el algoritmo porque tendrá menos comparaciones falsas no coincidentes para pequeñas cantidades de comparaciones falsas coincidentes.
- *Equal Error Rate* (EER): es el valor de FMR, o FNMR, para el umbral de similitud en el cual se cumple que las cantidades FMR y FNMR se igualan. Mientras menor sea este valor mejor es el algoritmo porque indica que existe un umbral de similitud para el cual se cometen una cantidad pequeña de ambos tipos de errores.
- *Zero False Match Rate* (ZeroFMR): es la cantidad de comparaciones falsas no coincidentes para la cual no hay comparaciones falsas coincidentes. Mientras menor sea este valor mejor será el algoritmo puesto que indica que logra reconocer un elevado número de verdaderos positivos sin cometer falsos positivos.
- *100 False Match Rate* (100FMR): el valor de FNMR para cuando FMR tiene un valor de 1/100. Mientras menor sea este valor y el de 1000FMR mejor será el algoritmo porque logra reconocer un elevado número de huellas cometiendo aceptando pocas comparaciones falsas coincidentes.
- *1000 False Match Rate* (1000FMR): el valor de FNMR para cuando FMR tiene un valor de 1/1000.
- *Tiempo de comparación* (Tiempo): el promedio de tiempo de comparación de huellas.

La primera base de datos de prueba de la competencia FVC2006 contiene imágenes en muy baja resolución (250 ppp) la cual es muy baja para la extracción de minucias de manera eficaz (Maltoni et al. 2009). Adicionalmente, estas huellas son de muy pequeña área conteniendo entre 1 y 5 minucias cada una siendo impráctico la aplicación de algoritmos basados en minucias para comparar estas huellas. En este contexto, los algoritmos de mejor eficacia son los basados en las imágenes o en características globales. Por tal razón en esta investigación no se experimenta con esta base de datos.

Los algoritmos evaluados en esta sección son:

1. El algoritmo propuesto en esta investigación para verificación de huellas dactilares (M3gl2) y el algoritmo propuesto para verificación de huellas palmares (M3gl2.5) (descritos en la sección 3.5).
2. El algoritmo JY (Jiang y Yau 2000) es evaluado puesto que es el primer algoritmo basado en tríos de minucias (Maltoni et al. 2009) para la verificación de huellas propuesto en la literatura.
3. El algoritmo TK (Tico y Kuosmanen 2003) puesto que usa un descriptor basado en textura, frecuentemente citado y evaluado en la literatura actual. Además, es interés de esta investigación la comparación con algoritmos basados en descriptores diferentes a los basados en minucias vecinas.
4. El algoritmo PN (Parziale y Niel 2004) puesto que es el algoritmo de la literatura con representación de tríos de minucias más parecida a la nuestra.
5. El algoritmo MCC (Cappelli et al. 2010, Cappelli et al. 2011, Ferrara et al. 2012) en su versión 1.4 del 2014 porque es el algoritmo de instituciones académicas con mejor eficacia en la competencia FVC-onGoing teniendo una SDK (Booch et al. 2007) gratis para la investigación; además, es de interés de esta investigación hacer comparaciones con algoritmos que usan descriptores basados en minucias vecinas y que no son tríos de minucias.

Exceptuando MCC, los demás algoritmos evaluados fueron implementados por los autores de esta investigación. No se implementaron más algoritmos de la literatura basados en tríos de minucias porque los autores no brindan suficiente detalles de estos algoritmos que permitan su implementación exacta sin cometer errores. No obstante, el mayor valor de esta investigación está en que se identificaron los problemas teóricos de los algoritmos existentes basados en tríos de minucias y se proponen nuevas formulaciones matemáticas que superan las limitaciones identificadas.

Los rasgos de las huellas en las competencias FVC2002, FVC2004 y FVC2006 son extraídos con la herramienta Verifinger SDK (Neurotechnology Inc. 2014) ya que estuvo entre los 4 mejores lugares en estas competencias y sus autores proveen una licencia de prueba por 30 días que puede ser usada para la investigación.

En total, se experimenta en 11 bases de datos (4 en FVC2002 con 800 huellas cada una; 4 en FVC2004 con 800 huellas cada una y 3 en FVC2006 con 1680 huellas cada una) con 5 algoritmos así que en esta sección se construyen 11 curvas ROC y se hacen  $11 \times 5 \times 5 = 275$  mediciones entre los indicadores de eficacia (EER, FMR100, FMR1000 y ZeroFMR) y eficiencia (tiempo promedio). Además de la comparación de los algoritmos a través de curvas ROC, los valores para cada indicador de eficacia y eficiencia son comparados usando la prueba de Friedman (Demšar 2006). El análisis estadístico para los indicadores EER, FMR100,

FMR1000, ZeroFMR y tiempo de comparación, es otra novedad de esta investigación puesto que, en la literatura consultada, la primera vez que se hace (Medina-Pérez et al. 2014) es en el contexto de esta investigación. Se aplica la prueba de Friedman por las siguientes razones:

- Como no se conoce la distribución a priori de los datos que se están comparando, se recomienda usar una prueba no paramétrica y la prueba de Friedman ha demostrado ser útil en la comparación de múltiples algoritmos evaluados en múltiples bases de datos (Demšar 2006).
- Esta prueba compara el ranking promedio de cada algoritmo por tanto se puede aplicar en cualquier problema donde las evaluaciones de los algoritmos se pueden ordenar (Demšar 2006, Derrac et al. 2011).
- La cantidad de algoritmos y base de datos evaluados se encuentra en el rango correcto sugerido en (Derrac et al. 2011) para la aplicación de la prueba de Friedman.

Como la prueba de Friedman no permite identificar entre cuáles pares de algoritmos hay diferencia significativas, se realiza el análisis post-hoc Bergmann-Hommel (García y Herrera 2008) ya que varios autores (García y Herrera 2008, Derrac et al. 2011) consideran que es el de mejor rendimiento para las comparaciones de múltiples algoritmos en múltiples bases de datos.

El algoritmo M3gl2 también es evaluado en las bases de datos FMISO-HARD-1.0 y FMISO-STD-1.0 de la competencia internacional en curso FVC-onGoing (Dorizzi et al. 2009) para comparar su rendimiento con los algoritmos de la literatura y comerciales que aparecen en dicha competencia. En estas bases de datos los algoritmos no requieren de un extractor de rasgos puesto que solo trabajan con las minucias que fueron previamente calculadas. Vale la pena resaltar que FVC-onGoing es la competencia más reciente para verificación de huellas y se encuentra activa por lo que los resultados publicados en ella pueden variar con el tiempo.

El Cuadro 4.1 muestra las diferentes tecnologías usadas en la captura de las imágenes para la verificación de huellas así como las dimensiones de las imágenes.

La Figura 4.1 muestra que las imágenes por base de datos difieren en características como: área capturada de los dedos, claridad del fondo de las imágenes, ruido introducido por los escáneres y el contraste entre crestas y surcos.

Aunque no es un objetivo inicialmente propuesto en esta investigación, el algoritmo M3gl2.5 es evaluado en las bases de datos PV-PARTIAL-1.0 y PV-FULL-1.0 de la competencia internacional en curso FVC-onGoing (Dorizzi et al. 2009) para mostrar el desempeño en la verificación de huellas palmares.

Con excepción de los experimentos en la competencias FVC-onGoing, el resto de los experimentos fueron realizados en una plataforma pública (Medina-Pérez et al. 2014) creada en esta investigación para fomentar el desarrollo de las investigaciones de reconocimiento de huellas.

Base de datos	Tecnología	Dimensiones de imágenes
FVC2002-DB1_A	Sensor óptico (Identix TouchView II)	$388 \times 374ppp^2$
FVC2002-DB2_A	Sensor óptico (Biometrika FX2000)	$296 \times 560ppp^2$
FVC2002-DB3_A	Sensor capacitivo (Precise Biometrics 100 SC)	$300 \times 300ppp^2$
FVC2002-DB4_A	Generador sintético (SFinGE v2.51)	$288 \times 384ppp^2$
FVC2004-DB1_A	Sensor óptico (CrossMatch V300)	$640 \times 480ppp^2$
FVC2004-DB2_A	Sensor óptico (Digital Persona U.are.U 4000)	$328 \times 364ppp^2$
FVC2004-DB3_A	Sensor de barrido térmico (Atmel FingerChip)	$300 \times 480ppp^2$
FVC2004-DB4_A	Generador sintético (SFinGE v3.0)	$288 \times 384ppp^2$
FVC2006-DB2_A	Sensor óptico (BiometriKa)	$400 \times 560ppp^2$
FVC2006-DB3_A	Sensor de barrido térmico (Atmel)	$400 \times 500ppp^2$
FVC2006-DB4_A	Generador sintético (SFinGE v3.0)	$288 \times 384ppp^2$

Cuadro 4.1: Tecnologías usadas en la captura de las imágenes para la verificación de huellas así como las dimensiones de las imágenes. Las tecnologías usadas para la creación de las bases de datos FMISO-HARD-1.0 y FMISO-STD-1.0 de la competencia FVC-onGoing no fueron publicadas por sus autores.

## 4.2 Resultados en las competencias FVC2002, FVC2004 y

### FVC2006

Las curvas ROC (Figura 4.2, Figura 4.3, Figura 4.4) muestran que M3gl2 supera en eficacia a los demás algoritmos basados en tríos de minucias. Para analizar en más detalle los resultados, por cada algoritmo se miden 4 indicadores de eficacia (EER, FMR100, FMR1000 y ZeroFMR) y uno de eficiencia (tiempo promedio). El análisis del Cuadro 4.2 y el Cuadro 4.3 confirma que el algoritmo M3gl2 mejora la eficacia de los demás algoritmos basados en tríos de minucias. Los resultados apoyan la hipótesis de esta investigación de que se puede obtener mejor eficacia al superar las limitaciones de los algoritmos basados en tríos de minucias existentes.

Aunque no existe una clara superioridad entre M3gl2 y MCC, el conteo de los mejores resultados por indicador de eficacia muestra que ambos algoritmos empatan en 3 ocasiones mientras que M3gl gana en 22 ocasiones y MCC en 19. En cuanto a eficiencia, el algoritmo MCC es siempre el más rápido y esto se debe a un conjunto de optimizaciones introducidas por sus autores en (Cappelli, Ferrara y Maltoni 2012) que también pudieran aplicarse a M3gl2 pero no es parte de los objetivos de esta investigación.

Adicionalmente, en esta sección se aplican la prueba de Friedman (Demšar 2006) y el

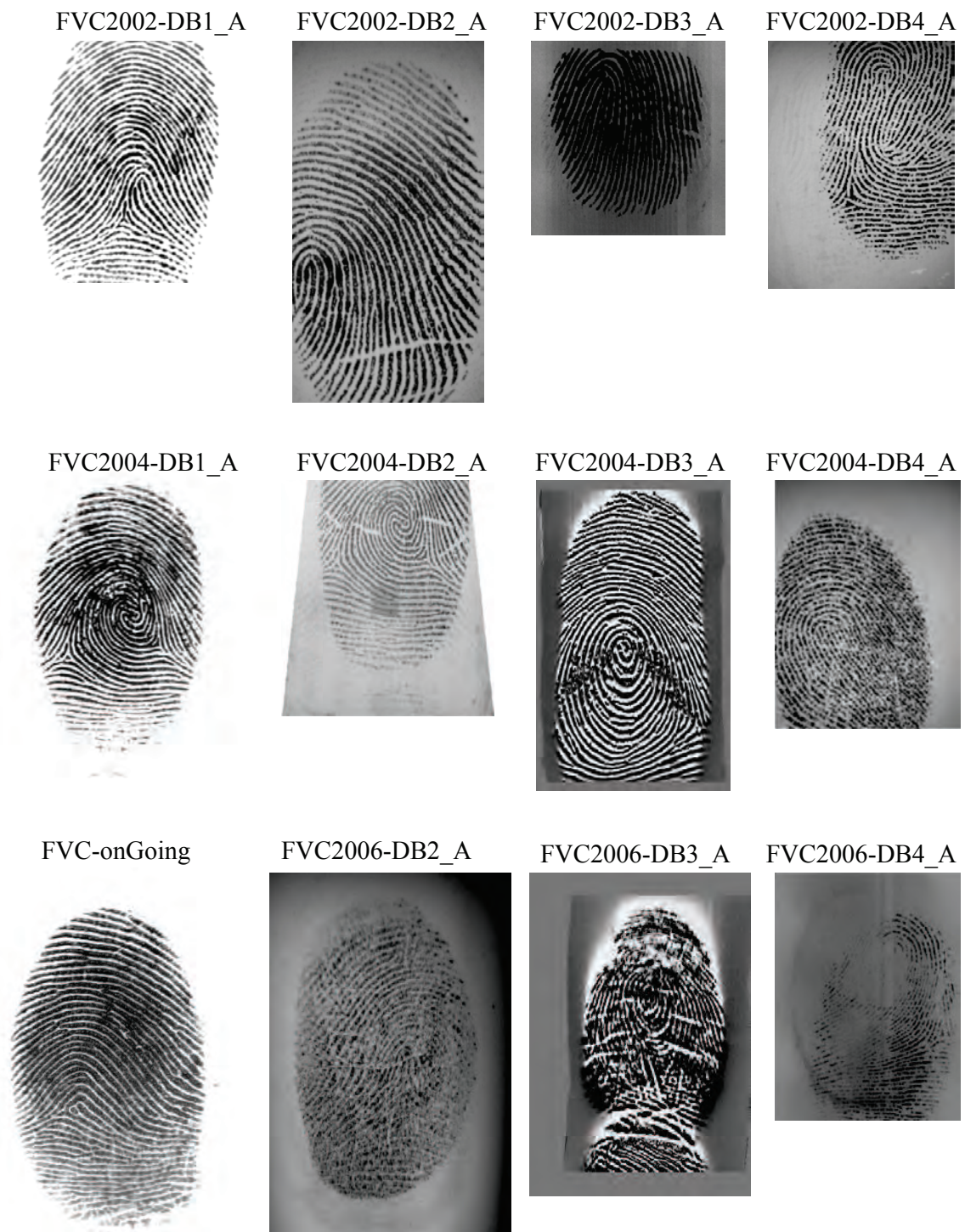


Figura 4.1: Ejemplos de huellas pertenecientes a cada una de las bases de datos para la verificación usadas en esta investigación.

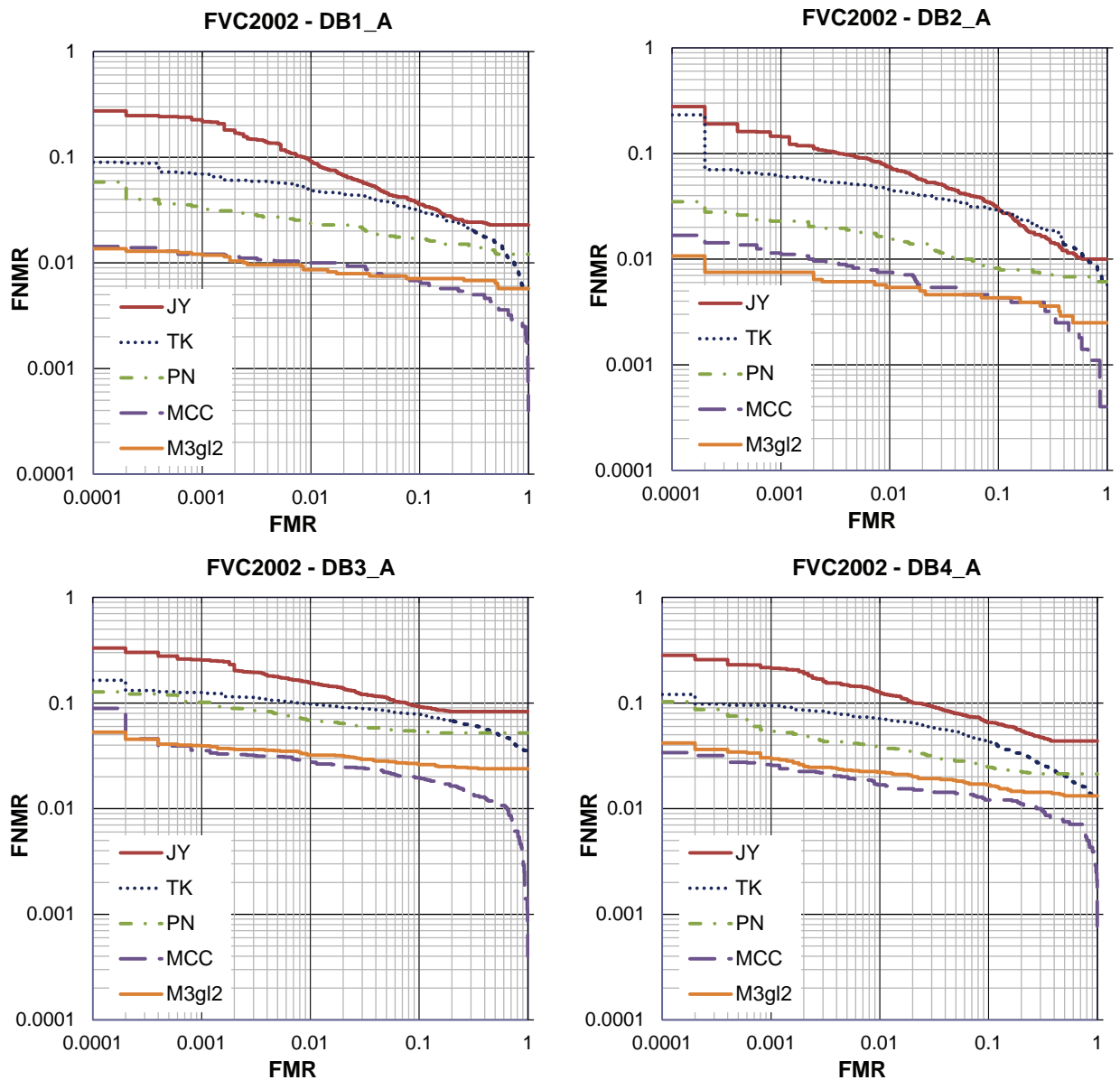


Figura 4.2: Curvas ROC con los resultados de los algoritmos en la competencia internacional FVC2002 (Maio et al. 2002).

análisis post-hoc Bergmann-Hommel (García y Herrera 2008) para determinar si existen diferencias significativas entre los resultados correspondientes a los indicadores de eficacia y eficiencia medidos. En esta sección se usan los diagramas de *Diferencias Críticas* (DC) (Demšar 2006) para mostrar los resultados de las pruebas estadísticas. Los diagramas DC presentan de manera compacta el orden de cada algoritmo de acuerdo a un indicador de rendimiento. Estos diagramas también muestran la magnitud de las diferencias entre algorit-

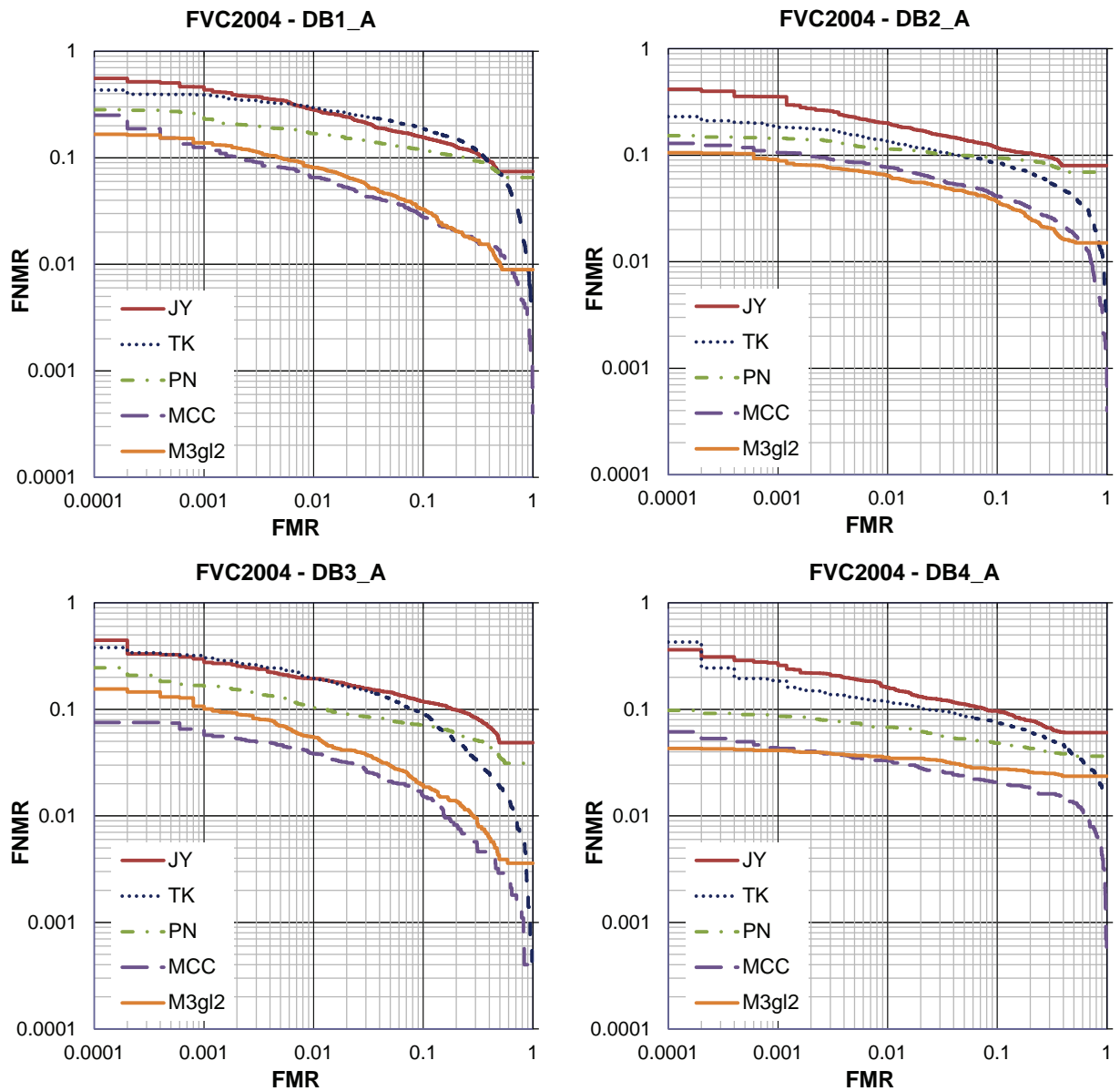


Figura 4.3: Curvas ROC con los resultados de los algoritmos en la competencia internacional FVC2004 (Maio et al. 2004).

mos y la significancia de las diferencias observadas (Demšar 2006). En los diagramas DC, los algoritmos son mejores mientras más a la derecha se encuentren, mientras que los algoritmos que comparten una línea gruesa no tienen diferencia estadística significativa.

La Figura 4.5 muestra que, M3gl2 mejora claramente el rendimiento del resto de los algoritmos basados en tríos de minucias y permite alcanzar eficacias similares a uno de los algoritmos más competitivos del estado del arte como es MCC incluso llegando a sobrepasarlo



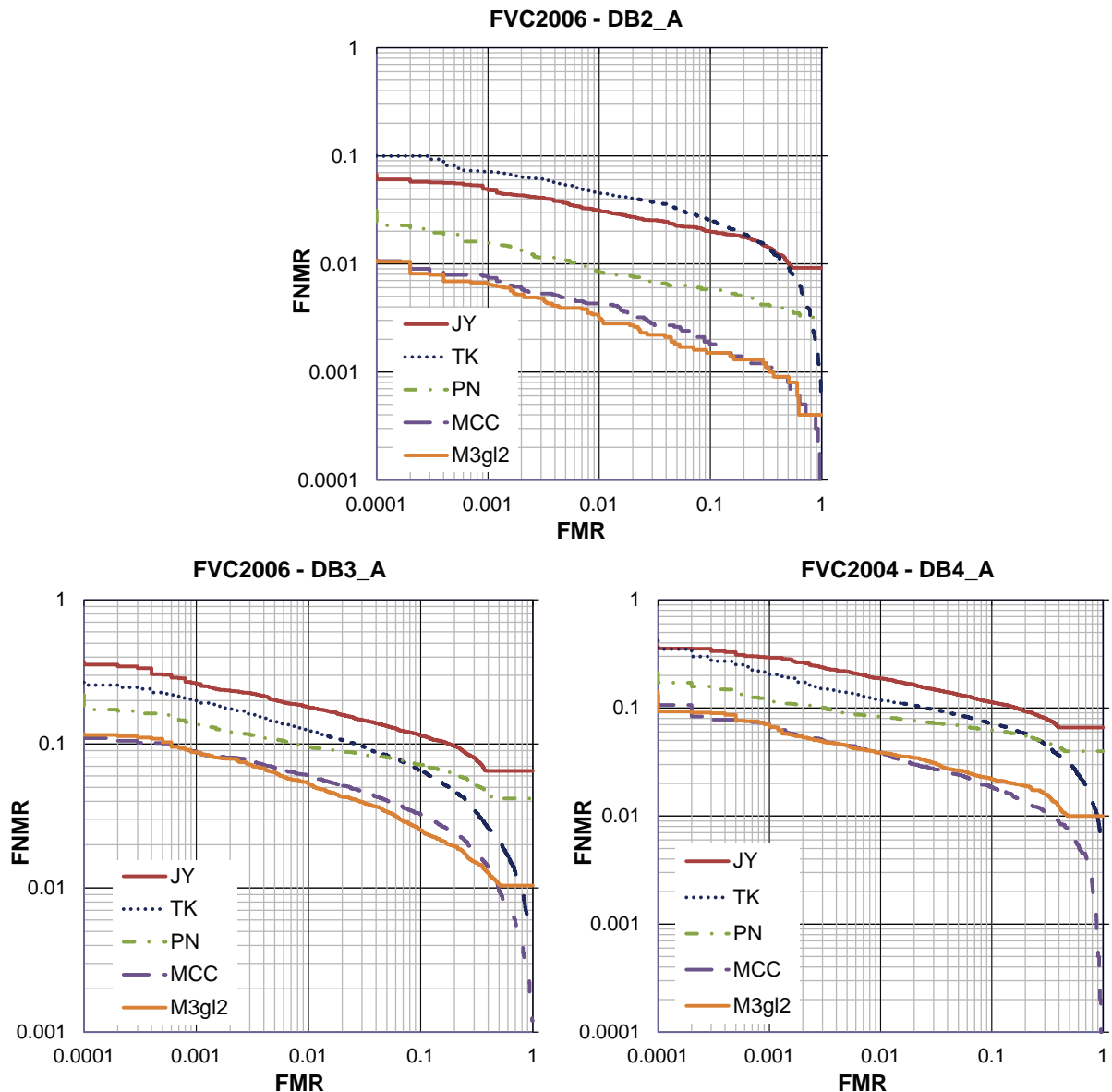


Figura 4.4: Curvas ROC con los resultados de los algoritmos en la competencia internacional FVC2006 (Dorizzi et al. 2009).

ligeramente en algunos indicadores.

La herramienta propuesta por los autores de MCC no retorna las minucias coincidentes en cada comparación ni incluye los códigos fuente de su algoritmo para estudiar su ejecución; esto dificulta el análisis de por qué MCC supera en algunos casos a M3gl2. Una posible hipótesis es que MCC usa un descriptor de minucias con más información que los tríos de minucias lo cual permite una mayor discriminación de las huellas a nivel local. Para

Base de datos	Algoritmos	EER (%)	FMR100 (%)	FMR1000 (%)	ZeroFMR (%)	Tiempo promedio (ms)
FVC2002 DB1_A	JY	4.7	9.1	22.7	27.4	1.1
	TK	3.9	5.0	7.0	9.0	2.7
	PN	2.2	2.4	3.4	5.8	7.9
	MCC	1.0	1.0	<b>1.2</b>	<b>1.4</b>	<b>1.0</b>
	M3gl2	<b>0.9</b>	<b>0.9</b>	<b>1.2</b>	<b>1.4</b>	2.0
FVC2002 DB2_A	JY	4.3	7.5	14.6	27.8	2.1
	TK	3.5	4.5	6.3	23.2	4.7
	PN	1.3	1.5	2.3	3.5	17.4
	MCC	0.8	0.8	1.1	1.7	<b>1.4</b>
	M3gl2	<b>0.4</b>	<b>0.5</b>	<b>0.8</b>	<b>1.1</b>	2.7
FVC2002 DB3_A	JY	9.4	15.6	25.8	33.1	0.5
	TK	7.9	9.8	12.6	16.4	1.3
	PN	5.4	6.9	10.2	12.8	1.9
	MCC	<b>2.3</b>	<b>2.8</b>	<b>3.6</b>	8.9	<b>0.4</b>
	M3gl2	2.9	3.2	4.0	<b>5.3</b>	0.8
FVC2002 DB4_A	JY	7.3	12.6	21.7	28.3	0.8
	TK	5.1	7.1	9.4	12.1	1.8
	PN	3.1	3.9	5.6	10.3	3.5
	MCC	<b>1.5</b>	<b>1.7</b>	<b>2.6</b>	<b>3.4</b>	<b>0.7</b>
	M3gl2	2.0	2.2	3.0	4.2	1.3

Cuadro 4.2: Resultados experimentales en las bases de datos de la competencia FVC2002 (Maio et al. 2002). En negritas aparecen los mejores resultados por cada base de datos y por cada indicador de eficacia o eficiencia.

probar esta hipótesis, se modificó M3gl2 cambiando los tríos de minucias por Cylinder Codes (representación usada por MCC) y el algoritmo obtenido superó en mayor medida la eficacia de MCC pero estos resultados están fuera del ámbito de esta investigación.

La relevancia de los resultados obtenidos en la presente investigación está en que no solo se superó la eficacia de otros algoritmos basados en tríos de minucias sino que se mostró que se podía alcanzar una eficacia similar o superior a la obtenida por un algoritmo basado en un descriptor de minucias que usa más información que los tríos de minucias.

### 4.3 Resultados en la competencia FVC-onGoing

FVC-onGoing (Dorizzi et al. 2009) es la única competencia activa de comparación de huellas. En ella se han evaluado más de 1600 algoritmos de comparación de huellas dacti-

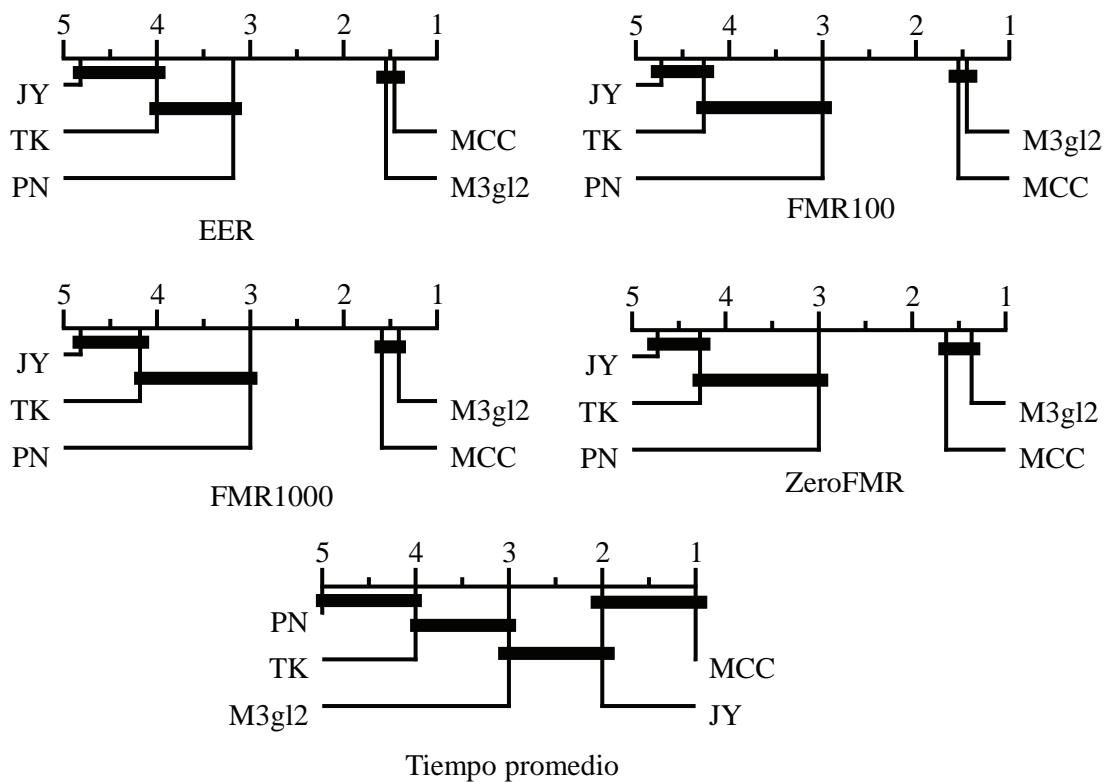


Figura 4.5: Diagramas de Diferencias Críticas (Demšar 2006) donde se muestra el orden de los algoritmos y la significancia estadística de las diferencias para los indicadores de eficacia y eficiencia.

lares y más de 200 algoritmos de comparación de huellas palmares. Las secciones siguientes muestran los resultados de los algoritmos propuestos en el contexto de esta competencia.

### 4.3.1 Resultados en la verificación de huellas dactilares

El Cuadro 4.4 muestra que, entre 20 resultados públicos, M3gl2 se encuentra entre los mejores 11 resultados de eficacia. El Cuadro 4.5 muestra que, entre 13 resultados públicos, M3gl2 se sitúa entre los 9 de mejor eficacia. Es necesario mencionar que en estas bases de datos se han evaluado más de 1600 algoritmos y la mayoría de los resultados publicados no son académicos ni de código abierto y algunos, como MM\_FMISO de la compañía Neurotechnology, llevan más de 14 años siendo desarrollados por empresas globalizadas. Si se compara M3gl2 con aquellos que no son desarrollados por compañías, el algoritmo propuesto se encuentra entre los tres con mejores resultados de eficacia con una diferencia de menos de 0.7% de eficacia con respecto al mejor algoritmo.

### 4.3.2 Resultados en la verificación de huellas palmares

Los objetivos de esta investigación se enmarcaron en el trabajo con huellas dactilares en principio. No obstante, el desarrollo de la investigación condujo a la hipótesis de que el algoritmo propuesto para la comparación de huellas sería eficaz también para comparar huellas palmares porque la mezcla de clusters abarca toda el área de las huellas palmares tolerando la distorsiones presentes en estas huellas. Es por esto que el algoritmo propuesto (M3gl2.5) es probado en las bases de datos de verificación de huellas palmares PV-FULL-1.0 y PV-Partial-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009).

El Cuadro 4.6 y el Cuadro 4.7 muestran que el algoritmo propuesto está entre los dos algoritmos más eficaces en la comparación de huellas palmares. Este es un buen resultado teniendo en cuenta que, aunque solo son públicos los resultados de 5 algoritmos, en esta modalidad de la competencia se han evaluado 197 algoritmos. Para que se tenga una medida de la importancia de este resultado, basta mencionar que nuestro algoritmo supera a *MinutiaeClusterFull* y *MinutiaeClusterPartial* que son algoritmos propuestos recientemente (Liu, Jain y Tian 2013) en una de las revistas más importante de la temática (*IEEE Transactions on Pattern Analysis and Machine Intelligence*) y por uno de los investigadores más reconocidos en el reconocimiento de huellas (Anil Jain).

## 4.4 Síntesis y conclusiones

El nuevo algoritmo para la verificación de huellas dactilares (M3gl2) es evaluado en 11 bases de datos públicas y su desempeño es comparado con otros 5 algoritmos mostrando que la eficacia de los algoritmos basados en tríos de minucias puede incrementarse al superar las limitaciones identificadas en esta investigación.

M3gl2 es también evaluado en dos bases de datos no públicas de la competencia internacional en curso FVC-onGoing alcanzando lugares de eficacia entre el 9 y el 11 entre más de 1600 algoritmos evaluados. El nuevo algoritmo queda entre los tres de mejor eficacia de los algoritmos que no son creados por compañías.

El algoritmo M3gl2.5 es evaluado en dos bases de datos de huellas palmares de la competencia internacional en curso FVC-onGoing estando entre los dos mejores algoritmos en cuanto a eficacia.

Base de datos	Algoritmos	EER (%)	FMR100 (%)	FMR1000 (%)	ZeroFMR (%)	Tiempo promedio (ms)
FVC2004 DB1_A	JY	14.2	28.3	46.3	55.7	1.1
	TK	16.0	29.5	39.0	43.1	2.8
	PN	11.3	16.9	26.1	28.3	7.0
	MCC	<b>4.0</b>	<b>6.5</b>	<b>12.5</b>	25.0	<b>0.8</b>
	M3gl2	4.5	8.1	13.8	<b>16.6</b>	1.8
FVC2004 DB2_A	JY	11.2	20.0	35.4	41.6	0.9
	TK	8.7	13.5	18.9	23.1	2.2
	PN	9.4	11.4	14.5	15.3	4.8
	MCC	5.1	7.7	10.6	12.9	<b>0.62</b>
	M3gl2	<b>4.3</b>	<b>6.5</b>	<b>9.1</b>	<b>10.6</b>	1.4
FVC2004 DB3_A	JY	11.5	19.4	29.8	44.6	2.1
	TK	9.4	19.6	32.0	38.0	4.4
	PN	7.2	10.4	16.8	24.5	14.0
	MCC	<b>2.7</b>	<b>3.8</b>	<b>6.5</b>	<b>7.5</b>	<b>1.2</b>
	M3gl2	3.5	5.5	10.7	15.5	2.7
FVC2004 DB4_A	JY	9.3	16.0	27.3	36.3	0.9
	TK	7.9	11.8	18.6	42.9	2.3
	PN	5.2	6.8	8.8	9.8	5.8
	MCC	<b>2.6</b>	<b>3.3</b>	4.4	6.1	<b>0.7</b>
	M3gl2	3.3	3.5	<b>4.1</b>	<b>4.3</b>	1.5
FVC2006 DB2_A	JY	2.5	3.1	4.9	6.7	2.3
	TK	3.5	4.5	7.1	10.0	4.9
	PN	0.9	0.8	1.6	3.1	22.7
	MCC	0.5	0.4	0.8	<b>1.1</b>	<b>1.4</b>
	M3gl2	<b>0.4</b>	<b>0.3</b>	<b>0.7</b>	<b>1.1</b>	2.8
FVC2006 DB3_A	JY	11.2	18.0	26.6	37.2	1.7
	TK	7.3	12.5	20.3	26.8	3.5
	PN	7.5	9.6	13.7	22.0	11.3
	MCC	4.3	6.1	8.9	<b>11.5</b>	<b>1.0</b>
	M3gl2	<b>3.7</b>	<b>5.3</b>	<b>8.8</b>	11.7	2.3
FVC2006 DB4_A	JY	11.0	18.7	29.4	37.5	0.9
	TK	7.7	11.8	21.5	41.9	2.3
	PN	6.5	8.3	11.9	21.2	5.0
	MCC	<b>2.8</b>	3.9	7.2	<b>13.1</b>	<b>0.6</b>
	M3gl2	3.1	<b>3.8</b>	<b>7.1</b>	14.2	1.4

Cuadro 4.3: Resultados experimentales en las bases de datos de las competencias FVC2004 (Maio et al. 2004) y FVC2006 (Dorizzi et al. 2009). En negritas aparecen los mejores resultados por cada base de datos y por cada indicador de eficacia o eficiencia.

Algoritmo	Compañía	Tiempo promedio de comparación	EER	FMR100	FMR1000	ZeroFMR
MM_FMISO	Sí	18 ms	0.19 %	0.10 %	0.33 %	1.12 %
EMB9200	Sí	9 ms	0.23 %	0.16 %	0.29 %	0.70 %
Triple_M_ISO	Sí	3 ms	0.23 %	0.23 %	0.36 %	1.61 %
SFCore	Sí	18 ms	0.26 %	0.18 %	0.35 %	1.22 %
TigerAFIS	Sí	21 ms	0.30 %	0.21 %	0.42 %	1.74 %
Tiger ISO	Sí	21 ms	0.32 %	0.23 %	0.45 %	1.35 %
MntModel	No	116 ms	0.38 %	0.33 %	0.51 %	1.07 %
Fingerprint Matcher ISO	Sí	4 ms	0.39 %	0.32 %	0.59 %	1.43 %
MCC (Baseline)	No	242 ms	0.41 %	0.29 %	0.60 %	1.84 %
FMTISO	Sí	9 ms	0.44 %	0.40 %	0.56 %	1.12 %
M3gl2	No	540 ms	0.51 %	0.45 %	0.75 %	1.71 %
FMISO	No	41 ms	0.52 %	0.46 %	0.70 %	2.14 %
ABVD	No	74 ms	0.54 %	0.49 %	0.72 %	1.55 %
APF_FMISO	Sí	3 ms	0.58 %	0.55 %	0.80 %	1.66 %
alguxntaix	No	4 ms	0.84 %	0.84 %	1.17 %	2.28 %
MAJU_VisPRS_FIS_Matcher	No	23 ms	0.86 %	0.92 %	1.78 %	7.41 %
MAJU_VISPRS_FIS_algo	No	21 ms	0.87 %	0.92 %	1.41 %	3.36 %
BioPass Finger Matching	Sí	106 ms	0.99 %	1.12 %	1.65 %	4.41 %
Authentik8	Sí	48 ms	1.02 %	1.85 %	2.48 %	26.00 %
LSM	Sí	16 ms	1.10 %	1.13 %	1.61 %	4.39 %
SourceAFIS	No	244 ms	1.17 %	1.20 %	1.80 %	4.05 %
SpNN	No	1309 ms	33.05 %	71.24 %	78.17 %	89.96 %

Cuadro 4.4: Resultados de M3gl2 en la base de datos FMISO-STD-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009).

Algoritmo	Compañía	Tiempo promedio de comparación	EER	FMR100	FMR1000	ZeroFMR
Nitgen_ISO	Sí	3 ms	1.09 %	1.77 %	4.38 %	11.88 %
Triple_M_ISO	Sí	3 ms	1.10 %	1.64 %	3.16 %	11.61 %
EMB9200	Sí	7 ms	1.11 %	1.15 %	2.08 %	4.74 %
SFCore	Sí	12 ms	1.41 %	1.58 %	2.70 %	19.81 %
MM_FMISO	Sí	18 ms	1.50 %	1.88 %	3.25 %	4.41 %
MntModel	No	82 ms	1.59 %	1.76 %	2.82 %	4.71 %
MCC (Baseline)	No	237 ms	1.77 %	2.05 %	3.62 %	6.07 %
Fingerprint Matcher ISO	Sí	5 ms	1.91 %	2.25 %	3.72 %	5.71 %
M3gl2	No	455 ms	1.92 %	2.28 %	3.80 %	6.66 %
FMTISO	Sí	14 ms	1.99 %	2.33 %	3.50 %	6.24 %
alguxntaix	No	5 ms	2.38 %	3.17 %	5.20 %	6.74 %
APF_FMISO	Sí	3 ms	2.55 %	3.15 %	4.58 %	14.35 %
SourceAFIS	No	235 ms	3.68 %	4.85 %	6.71 %	8.91 %
LSM	Sí	17 ms	3.83 %	5.03 %	8.73 %	12.71 %

Cuadro 4.5: Resultados de M3gl2 en la base de datos FMISO-HARD-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009).

Algoritmo	Tiempo promedio de comparación	EER	FMR100	FMR1000	ZeroFMR
MCC	241 ms	2.25 %	2.64 %	4.00 %	20.11 %
M3gl2.5	911 ms	2.92 %	3.68 %	5.07 %	9.64 %
Mnt	428 ms	4.07 %	5.43 %	7.14 %	100.0 %
MinutiaeClusterFull	86 ms	5.60 %	6.07 %	6.86 %	10.21 %
RegionGrow+ GlobalMatch (Baseline)	2148 ms	10.73 %	28.25 %	44.25 %	57.14 %

Cuadro 4.6: Resultados de M3gl2.5 en la base de datos PV-FULL-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009).

<b>Algoritmo</b>	<b>Tiempo promedio de comparación</b>	<b>EER</b>	<b>FMR100</b>	<b>FMR1000</b>	<b>ZeroFMR</b>
MCC	189 ms	2.50 %	3.50 %	4.25 %	5.25 %
M3gl2.5	725 ms	4.25 %	5.0 %	5.75 %	7.75 %
MinutiaeClusterPartial	115 ms	5.0 %	5.50 %	8.75 %	9.25 %
RegionGrow+ GlobalMatch (Baseline)	1159 ms	6.26 %	16.25 %	25.50 %	40.50 %

Cuadro 4.7: Resultados de M3gl2.5 en la base de datos PV-PARTIAL-1.0 de la competencia internacional FVC-onGoing (Dorizzi et al. 2009).



# 5

## Resultados experimentales en la identificación de huellas latentes

En este capítulo se prueba la eficacia del algoritmo propuesto para la identificación de huellas latentes. Las pruebas se realizan en la base de datos NIST SD27 y una base de datos adicional de más de un millón de huellas latentes.

### 5.1 Protocolos de evaluación

A diferencia de las bases de datos de verificación, en la literatura consultada sobre identificación de huellas dactilares latentes solo existe la base de datos pública NIST SD27 (Garris y McCabe 2000), la que tiene el inconveniente de ser pequeña (contiene 258 huellas latentes y 258 impresiones). La base de datos NIST SD27 es incrementada con las bases de datos NIST SD4 (Watson y Wilson 1992) y NIST SD14 (Watson 2001) para reproducir los experimentos descritos en (Jain et al. 2008, Yoon et al. 2010, Yoon et al. 2011, Mikaelyan y Bigun 2012, Paulino et al. 2011, Jain y Feng 2011).

Existen otros artículos (Paulino, Feng y Jain 2013, Liu, Arora, Cao, Jain y Lansing 2013) de identificación de huellas latentes donde se experimenta con la base de datos NIST SD27 pero adicionan huellas que no están públicas y, por tanto, no se pueden reproducir los

experimentos.

Nuestro algoritmo para identificación de huellas latentes (M3gl3) es evaluado siguiendo el protocolo propuesto en la literatura actual (Paulino, Feng y Jain 2013, Mikaelyan y Bigun 2012, Jain y Feng 2011). En este protocolo las huellas latentes son comparadas contra todas las impresiones ya sean del mismo o diferentes dedos. Con esta información se construye la curva CMC (Jain y Feng 2011) (*Cumulative Match Characteristic curve*). Esta curva representa la cantidad de huellas latentes identificadas correctamente para las diferentes posiciones de ranking al ordenar las comparaciones descendientemente según el valor de similitud. El algoritmo ideal logra un 100 % de identificación para el ranking 1.

El algoritmo propuesto es solo comparado contra aquellos basados en minucias para ser justos puesto que M3gl3 solo usa minucias y es aceptado en la literatura que el uso de rasgos adicionales eleva la eficacia de identificación (Jain et al. 2008, Jain y Feng 2011, Yoon et al. 2011, Liu, Arora, Cao, Jain y Lansing 2013). Las minucias son calculadas con la herramienta Verifinger SDK (Neurotechnology Inc. 2014) excepto para NIST SD27 porque esta base de datos incluye las minucias de sus huellas.

Para probar la escalabilidad del nuevo algoritmo, la base de datos NIST SD27 es aumentada con más de 1 millón de huellas entre sintéticas y reales.

Aunque no está dentro de los objetivos de esta investigación, en esta sección se prueba también el desempeño de M3gl3 en el contexto de la identificación de huellas palmares latentes. En la literatura consultada no se encontró ninguna base de datos pública para la identificación de huellas palmares latentes es por esto que en esta investigación se propone una nueva base de datos: LPIDB v1.0.

LPIDB v1.0 está compuesta por 380 huellas palmares latentes obtenidas de 100 palmas de 51 sujetos voluntarios. La edad de los sujetos varía entre 4 y 81 años con diferentes profesiones (estudiantes, trabajadores manuales, etc). Cada sujeto contribuyó con dos impresiones palmares (mano izquierda y mano derecha) y múltiples huellas latentes que simulan escenarios realistas como la apertura de puertas, empujando sillas, sosteniendo un cuchillo, apoyándose sobre una mesa, cargando diferentes objetos, etc. Para aumentar la dificultad de la identificación, LPIDB v1.0 fue incrementada con las 1280 impresiones palmares de la base de datos THUPALMLAB (Dai et al. 2012). Las minucias de las huellas latentes fueron extraídas manualmente por los autores gracias al entrenamiento recibido de especialistas criminalistas mientras que las minucias en las impresiones fueron extraídas usando Verifinger SDK (Neurotechnology Inc. 2014). El protocolo de evaluación en esta base de datos es como al seguido para la identificación de huellas dactilares latentes.

Aunque la plataforma de experimentación propuesta en (Medina-Pérez et al. 2014) fue en principio concebida para la verificación de huellas; su diseño permitió la incorporación de los protocolos de evaluación para la identificación de huellas latentes sin modificar ningún módulos de la plataforma.

## 5.2 Resultados en la base de datos NIST SD27

Base de datos	Algoritmo	Referencias	Eficacia (ranking 1)
NIST SD27	VeriFinger SDK	(Yoon et al. 2010)	~58.0 %
	Bozorth3	(Mikaelyan y Bigun 2012)	47.0 %
	k-plet	(Mikaelyan y Bigun 2012)	55.0 %
	NMD-1	(Jain et al. 2008)	~63.0 %
	MCC	(Cappelli et al. 2010) (Cappelli et al. 2011) (Ferrara et al. 2012)	<b>83.3 %</b>
	M3gl3	Esta investigación	<b>83.3 %</b>
NIST SD27 + 2000 huellas de NIST SD14	DBHT	(Paulino et al. 2011)	62.4 %
	MCC	(Cappelli et al. 2010) (Cappelli et al. 2011) (Ferrara et al. 2012)	76.0 %
	M3gl3	Esta investigación	<b>79.1 %</b>
NIST SD27 + 2000 huellas de NIST SD4	NMD-1	(Jain et al. 2008)	~50.0 %
	MCC	(Cappelli et al. 2010) (Cappelli et al. 2011) (Ferrara et al. 2012)	76.4 %
	M3gl3	Esta investigación	<b>77.5 %</b>
NIST SD27 + 27000 huellas de NIST SD14	VeriFinger SDK	(Yoon et al. 2011)	~13.0 %
	MCC	(Cappelli et al. 2010) (Cappelli et al. 2011) (Ferrara et al. 2012)	67.4 %
	M3gl3	Esta investigación	<b>68.6 %</b>
NIST SD27 + 27000 huellas de NIST SD14 + 2000 huellas de NIST SD4	NMD-2	(Jain y Feng 2011)	34.9 %
	MCC	(Cappelli et al. 2010) (Cappelli et al. 2011) (Ferrara et al. 2012)	67.4 %
	M3gl3	Esta investigación	<b>68.2 %</b>

Cuadro 5.1: Comparación de eficacia de identificación en el ranking 1 usando la base de datos NIST SD27 (Garris y McCabe 2000), NIST SD4 (Watson y Wilson 1992) y NIST SD14 (Watson 2001). Los mejores resultados se encuentran en negritas.

Los experimentos descritos en (Jain et al. 2008, Yoon et al. 2010, Yoon et al. 2011, Mikaelyan y Bigun 2012, Paulino et al. 2011, Jain y Feng 2011) son reproducidos y el Cuadro 5.1 muestra que el algoritmo propuesto (M3gl3) alcanza un resultado igual o superior al

resto de los algoritmos. Vale la pena mencionar que algunos de los algoritmos comparados logran una mejor eficacia (ver Cuadro 2.2) cuando se combinan las minucias con información adicional pero esta sección se refiere a las versiones basadas solo en minucias para hacer justas las comparaciones.

Usualmente, la identificación de huellas latentes se hace sobre bases de datos de millones de huellas. Es por esto que, para probar el algoritmo en un contexto análogo al de los sistemas reales, las base de datos NIST SD27 es aumentada con 2000 impresiones de NIST SD4, 27000 impresiones de NIST SD14, 106921 impresiones de un laboratorio forense y 997337 impresiones sintéticas generadas usando el software *SFinGe Version 4.1 (build 1746) Demo* (Maltoni et al. 2009). SFinGe ha sido usado en varias competencias de verificación de huellas mostrando que los resultados en las bases de datos sintéticas son similares a los resultados en bases de datos reales (Cappelli et al. 2006).

En muchas ocasiones los peritos saben la correcta orientación de las huellas latentes, de esta manera al comparar la huella latente contra las impresiones se puede restringir el ángulo de rotación e incrementar la eficacia de la identificación. Es por esto que cada algoritmo se evaluó sin restricción de rotación de las huellas y con rotación restringida a  $\pi/4$ .

De los algoritmos comparados en el Cuadro 5.1, los únicos probados en los siguientes experimentos son M3gl3 y MCC puesto que los autores de los demás no proveen implementaciones públicas ni suficiente nivel de detalle en sus descripciones que permitan la implementación sin errores. Además, en comparación con M3gl3 y MCC, los demás algoritmos están muy lejos en términos de eficacia.

Cada huella latente es comparada con cada impresión; de aquí que el número de comparaciones que se realizan en total es: 2 (algoritmos) x 284 (huellas latentes) x 1133516 (impresiones) x 2 (restricciones de rotación) = 1287674176 (comparaciones). Las comparaciones son ordenadas de mayor a menor valor de similitud y se construye una gráfica de ranking de identificación contra tasa de identificación (curva CMC (Jain y Feng 2011)).

El análisis de los resultados mostrados en la Figura 5.1 revela que:

- El algoritmo propuesto con rotación restringida supera a MCC en 4.0 % como promedio; mientras que la versión sin rotación restringida supera a MCC en 15.2 % como promedio.
- La tasa de identificación de los algoritmos se reduce cuando no se restringen las rotaciones de las huellas. Específicamente la eficacia de M3gl3 se reduce 3.5 % como promedio mientras que la eficacia de MCC se reduce 14.7 % como promedio.

La calidad de las huellas latente tiene un impacto directo en la eficacia de los algoritmos de comparación. La evaluación de acuerdo a diferentes calidades de huellas permite entender mejor el rendimiento de los algoritmos. Las base de datos NIST SD27 agrupa sus huellas latentes en tres categorías de calidad (“Good”, “Bad” y “Ugly”) de acuerdo al criterio de

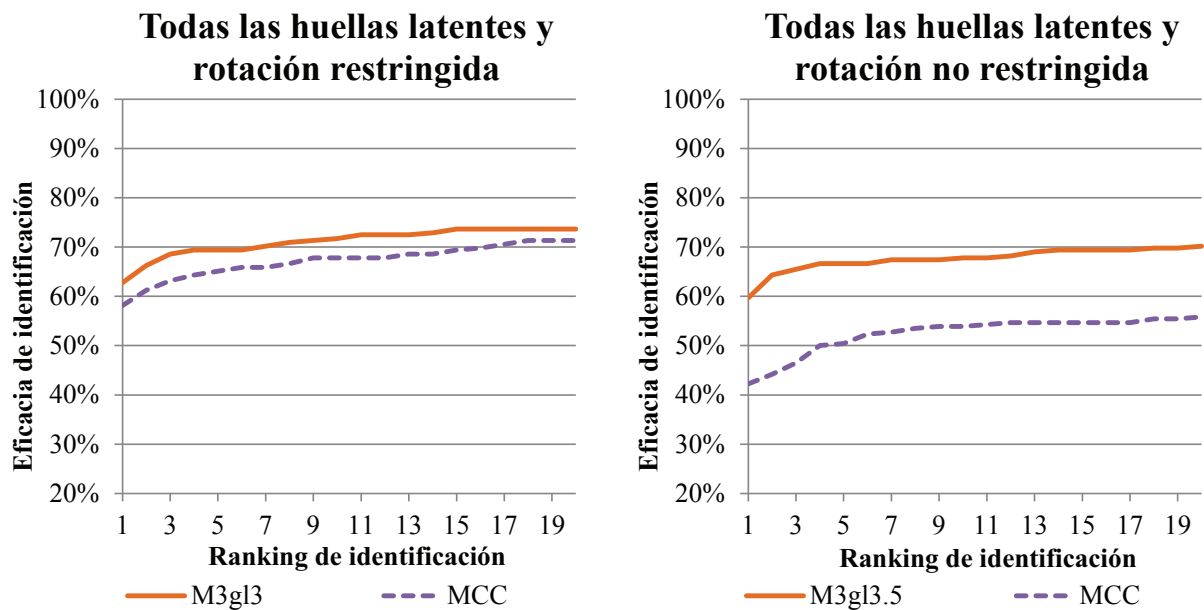


Figura 5.1: Curvas CMC (Jain y Feng 2011) con la eficacia de los algoritmos restringiendo y sin restringir la rotación al identificar las 258 huellas latentes de NIST SD27 contra una base de datos de 1133516 impresiones.

expertos dactiloscópicos del FBI. La Figura 5.2 muestra el resultado de M3gl3 y MCC para cada una de las categorías de huellas en NIST SD27.

El análisis de los resultados mostrados en la Figura 5.2 revela que:

- Para el caso de la identificación de huellas latentes de la categoría “Good”:
  - El algoritmo propuesto con rotación restringida supera a MCC en 5.0% como promedio; mientras que la versión sin rotación restringida supera a MCC en 11.0% como promedio.
  - Cuando no se restringen las rotaciones de las huellas, la tasa de identificación de M3gl3 se reduce 4.7% como promedio mientras que la eficacia de MCC se reduce 10.2% como promedio.
- Para el caso de la identificación de huellas latentes de la categoría “Bad”:
  - El algoritmo propuesto con rotación restringida supera a MCC en 4.4% como promedio; mientras que la versión sin rotación restringida supera a MCC en 19.5% como promedio.
  - Cuando no se restringen las rotaciones de las huellas, la tasa de identificación de M3gl3 se reduce 2.3% como promedio mientras que la eficacia de MCC se reduce 17.4% como promedio.

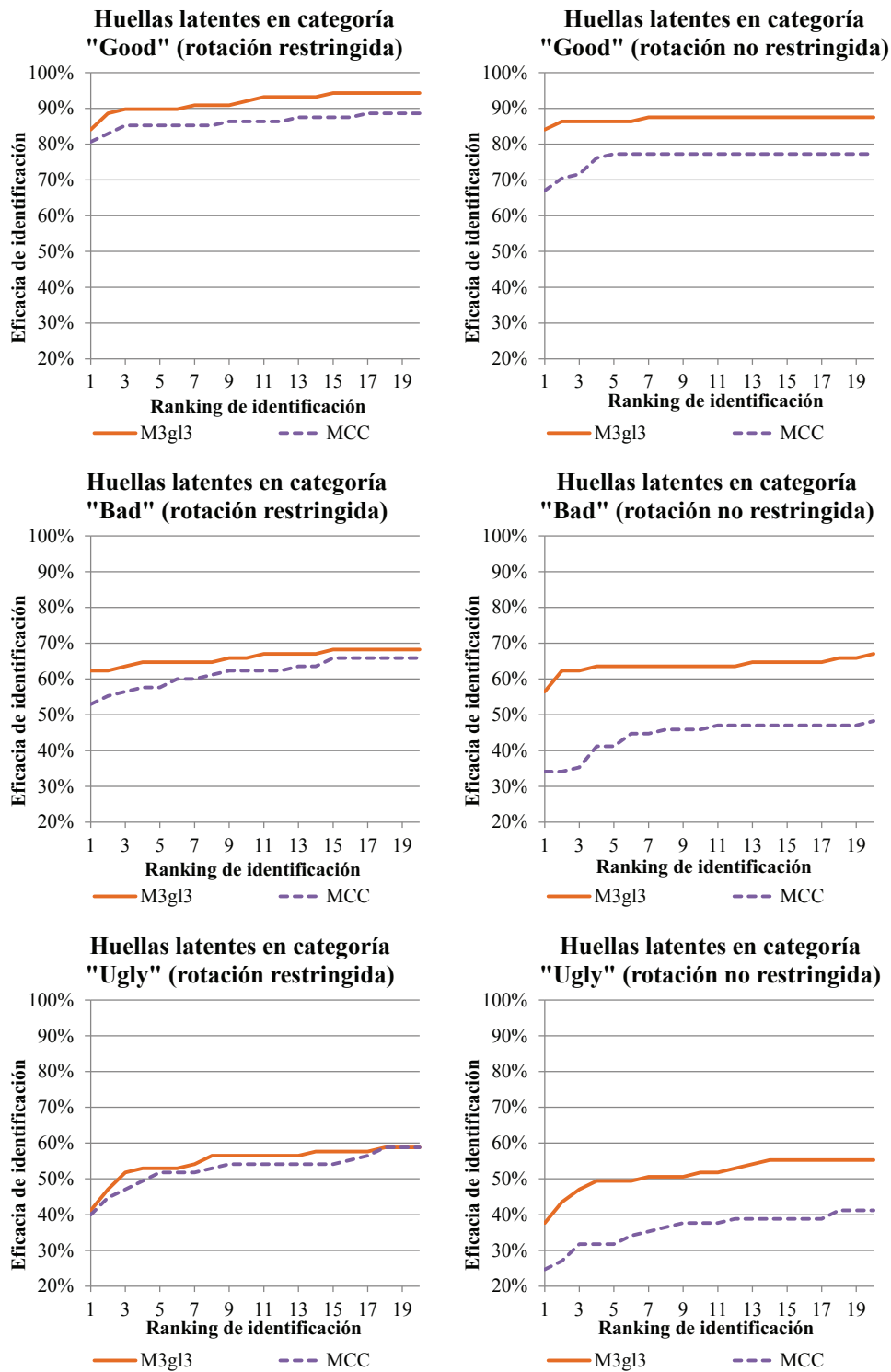


Figura 5.2: Curvas CMC (Jain y Feng 2011) con la eficacia de los algoritmos restringiendo y sin restringir la rotación al identificar las 258 huellas latentes de NIST SD27 divididas por categorías contra una base de datos de 1133516 impresiones.

- Para el caso de la identificación de huellas latentes de la categoría “Ugly”:
  - El algoritmo propuesto con rotación restringida supera a MCC en 2.1 % como promedio; mientras que la versión sin rotación restringida supera a MCC en 15.2 % como promedio.
  - Cuando no se restringen las rotaciones de las huellas, la tasa de identificación de M3gl3 se reduce 3.6 % como promedio mientras que la eficacia de MCC se reduce 16.7 % como promedio.

El tiempo de comparación promedio de M3gl3 y MCC es 0.46 ms y 0.44 ms respectivamente.

### 5.3 Resultados en la base de datos LPIDB v1.0

La identificación de huellas palmares latentes es un tema de interés en la actualidad (Liu, Jain y Tian 2013, Wang et al. 2012, Jain y Feng 2009) por su impacto en el proceso de esclarecimiento de delitos. Debido a la novedad del tema, no existe aún base de datos pública donde probar los algoritmos en igualdad de condiciones. Otra limitación de las investigación en este contexto, es que los algoritmos propuestos no están descritos con suficiente nivel de detalle como para ser reproducidos y los autores no proveen códigos fuentes ni ejecutables que se puedan reutilizar. Estas son las motivaciones para crear una nueva base de datos pública (LPIDB v1.0<sup>1</sup>) que sirva de punto de referencia común para las investigaciones futuras.

El principio de funcionamiento del nuevo algoritmo a través del cual halla múltiples clusters de coincidencias y los mezcla para abarcar la mayor parte del área de las huellas lleva a pensar que el nuevo algoritmo pudiera desempeñarse bien en la identificación de huellas palmares latentes. Basados en esta hipótesis, M3gl3 es probado en la base de datos LPIDB v1.0 aumentada con THUPALMLAB (Dai et al. 2012). También es evaluado MCC porque ha demostrado su alta eficacia en la comparación de huellas palmares (Cappelli, Ferrara y Maio 2012).

La Figura 5.3 muestra que la eficacia M3gl3 es superior a la de MCC para todos los ranking de identificación. El análisis de los resultados revela lo siguiente:

- M3gl3 con rotación restringida supera a MCC en 1.5 % como promedio; mientras que la versión sin rotación restringida supera a MCC en 4.6 % como promedio.
- Cuando no se restringen las rotaciones de las huellas, la tasa de identificación de M3gl3 se reduce 3.3 % como promedio mientras que la eficacia de MCC se reduce 6.4 % como promedio.

---

<sup>1</sup>Disponible en <http://www.gpds.ulpgc.es>

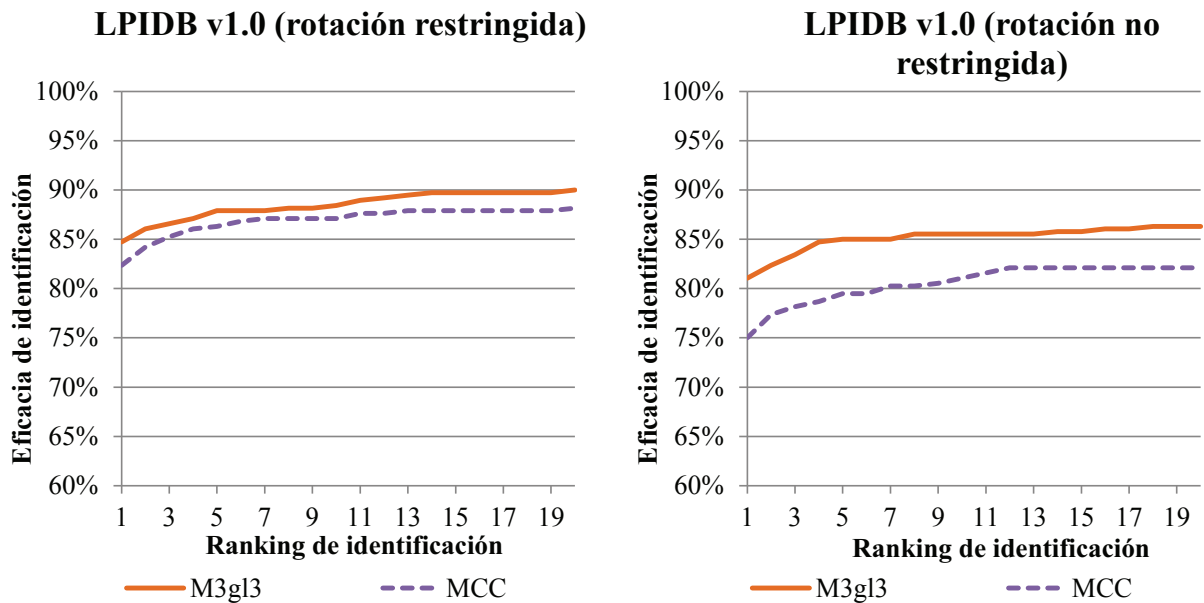


Figura 5.3: Curvas CMC (Jain y Feng 2011) con la eficacia de los algoritmos restringiendo y sin restringir la rotación al identificar las 380 huellas latentes de LPIDB v1.0 contra una base de datos con las 102 impresiones palmares de LPIDB v1.0 y las 1280 impresiones de THUPALMLAB.

## 5.4 Síntesis y conclusiones

El nuevo algoritmo para la identificación de huellas latentes (M3gl3) es evaluado en las bases de datos NIST SD27, NIST SD4 y NIST SD14 mostrando eficacia mejor o igual que la reportada en el estado del arte.

La base de datos NIST SD27 es incrementada con más de un millón de impresiones dactilares para evaluar el rendimiento de M3gl3 a gran escala. Los resultados muestran que la eficacia de M3gl3 supera la de MCC en 4.0% como promedio cuando se usa la información de la rotación de las huellas; mientras que M3gl3 supera a MCC en 15.2% como promedio cuando se desconoce la rotación correcta de las huellas. Las pruebas también muestran que M3gl3 supera a MCC independientemente de la calidad de las huellas latentes que se identifican.

Se propone una nueva base de datos pública para la identificación de huellas palmares latentes (LPIDB v1.0). Esta base de datos es incrementada con 1280 impresiones palmares de la base de datos THUPALMLAB y los experimentos muestran que M3gl3 es también eficaz para la identificación de huellas palmares latentes.



# 6

## Conclusiones y trabajos futuros

Los algoritmos de comparación de huellas basados en tríos de minucias tienen una eficacia y rapidez similar, y a veces superior, en comparación con otros tipos de algoritmos de comparación. Además, tienen la ventaja de poderse usar tanto en los sistemas embebidos en arquitecturas ligeras de hardware como en los sistemas basados en estándares de almacenamiento de huellas representadas por minucias, tecnologías cuya popularidad ha aumentado en los últimos años.

Los algoritmos basados en tríos de minucias propuestos en la literatura presentan al menos una de las siguientes limitaciones que degradan su eficacia: la dependencia del orden de las minucias en el rasgo; la insensibilidad a la reflexión de los tríos; se ignora las direcciones de las minucias relativas a los lados; la elección de una sola alineación basada en información local; la sensibilidad a errores del extractor de rasgos que genera rasgos ausentes o incorrectos y el no determinismo.

En esta investigación se proponen tres algoritmos de comparación de huellas basados en una nueva forma de representación y función de comparación de tríos de minucias que superan las limitaciones de los algoritmos del estado del arte. Los nuevos algoritmos mezclan múltiples clusters de coincidencias para ser robustos a las distorsiones presentes en las huellas.

El algoritmo de comparación (M3gl2) propuesto para la verificación de huellas dactilares es probado en 11 bases de datos públicas mostrando mejores resultados de eficacia que el resto de los algoritmos evaluados. M3gl2 es también evaluado en la competencia internacional en curso FVC-onGoing situándose entre los tres mejores lugares de los algoritmos que no

son creados por compañías.

El algoritmo de comparación (M3gl3) propuesto para la identificación de huellas latentes es probado en la identificación de las 258 huellas latentes contra diferentes bases de datos de dimensiones que varían entre 258 y más de un millón de impresiones mostrando una eficacia igual o superior a los resultados reportados en los artículos del estado del arte. Los experimentos realizados muestran que el algoritmo es robusto ante diferentes calidades de las huellas y su eficacia se reduce poco (en comparación con el algoritmo MCC) cuando se desconoce la orientación de las huellas latentes.

Los nuevos algoritmos están diseñados para ser robustos a las grandes distorsiones propias de las huellas. Al ser la superficie de la huella dactilar pequeña, no es sencillo apreciar sus ventajas. De aquí que, para verificar la hipótesis de mayor robustez a distorsiones frente a otros algoritmos, se prueba la eficacia en la comparación de huellas palmares ya que estas ocupan mayor área y tienen más distorsiones que las huellas dactilares. El algoritmo creado para la verificación de huellas palmares es evaluado en dos bases de datos no públicas de la competencia FVC-onGoing situándose entre los dos algoritmos de mejor eficacia entre 197 algoritmos evaluados. Para evaluar el algoritmo creado para la identificación de huellas latentes se crea una base de datos de huellas latentes palmares (LPIDB v1.0) donde el nuevo algoritmo muestra resultados satisfactorios.

Todos los experimentos de esta investigación se realizaron utilizando una nueva plataforma de experimentación en C# que actualmente está pública en la web para fomentar el desarrollo de las investigaciones de reconocimiento de huellas.

Algunos contextos de aplicación son más flexibles en cuanto a los rasgos que se pueden usar. Para estos casos se estudiará enriquecer la información de las minucias para mejorar la eficacia de los algoritmos. Otros contextos de aplicación en los cuales se estudiará la aplicación de los algoritmos propuestos son: comparación de huellas latentes contra huellas latentes (Sankaran et al. 2011), identificación de huellas latentes solapadas (Feng et al. 2012), identificación de huellas latentes simultáneas (Vatsa et al. 2011) e identificación de huellas 3D (Liu y Zhang 2014).

## 6.1 Publicaciones

Las publicaciones de los autores relacionadas con esta investigación son:

1. Medina-Pérez, M. A., Morales-Moreno, A., Ferrer-Ballester, M. A., García-Borroto, M., Loyola-González, O. and Altamirano-Robles, L.: Latent fingerprint identification using deformable minutiae clustering, Neurocomputing (*Aceptado*).
2. Morales-Moreno, A., Medina-Pérez, M. A., Ferrer-Ballester, M. A., García-Borroto,

- M. and Altamirano-Robles, L.: 2014, LPIDB v1.0 - Latent Palmprint Identification Database, International Joint Conference on Biometrics (IJCB2014), Florida, USA.
3. Medina-Pérez, M. A., Loyola-González, O., Gutiérrez-Rodríguez, A. E., García-Borroto, M. and Altamirano-Robles, L.: 2014, Introducing an experimental framework in C# for fingerprint recognition, *Lecture Notes in Computer Science* **8495**, 132–141.
  4. Medina-Pérez, M. A., García-Borroto, M., Gutiérrez-Rodríguez, A. E. and Altamirano-Robles, L.: 2012, Improving fingerprint verification using minutiae triplets, *Sensors* **12**, 3418–3437.
  5. Medina-Pérez, M. A., García-Borroto, M., Gutiérrez-Rodríguez, A. and Altamirano-Robles, L.: 2012, Improving the multiple alignments strategy for fingerprint verification, *Lecture Notes in Computer Science* **7329**, 147–154.
  6. Medina-Pérez, M. A., García-Borroto, M., Gutiérrez-Rodríguez, A. and Altamirano-Robles, L.: 2011, Robust fingerprint verification using m-triplets, *International Conference on Hand-Based Biometrics (ICHB 2011)*, Hong Kong, China, pp. 1–5.
  7. Medina-Pérez, M. A., Gutiérrez-Rodríguez, A. and García-Borroto, M.: 2009, Improving fingerprint matching using an orientation-based minutia descriptor, *Lecture Notes in Computer Science* **5856**, 121–128.

## 6.2 Patentes

Como resultado de esta investigación se generó la siguiente patente:

1. Medina-Pérez, M. A., Morales-Moreno, A., Ferrer-Ballester, M. A., García-Borroto, M. and Altamirano-Robles, L.: Sistema y método para la comparación de huellas dactilares y palmares basada en múltiples clústeres deformables de minucias coincidentes, (*En trámite*), México.



# Bibliografía

- Angeloni, M. D. A. and Marana, A. N.: 2013, Improving the ridge based fingerprint recognition method using sweat pores, *The Seventh International Conference on Digital Society (ICDS 2013)*, Nice, France, pp. 113–119.
- Anthonioz, A. and Champod, C.: 2014, Integration of pore features into the evaluation of fingerprint evidence, *Journal of Forensic Sciences* **59**(1), 82–93.
- Asha, S. and Chellappan, C.: 2014, Partial fingerprint matching using minutiae subset, in D. P. Mohapatra and S. Patnaik (eds), *Intelligent Computing, Networking, and Informatics*, Vol. 243 of *Advances in Intelligent Systems and Computing*, Springer India, pp. 445–452.
- Babatunde, I. G., Charles, A. O. and Officer, A. C.: 2013, Fingerprint matching by neighbourhood characteristics, *Science and Information Conference (SAI 2013)*, London, UK, pp. 434–442.
- Babatunde, I. G., Charles, A. O. and Officer, A. C.: 2014, Spatial relation approach to fingerprint matching, in L. Chen, S. Kapoor and R. Bhatia (eds), *Intelligent Systems for Science and Information*, Vol. 542 of *Studies in Computational Intelligence*, Springer International Publishing, pp. 87–110.
- Bazen, A. M. and Gerez, S. H.: 2003, Fingerprint matching by thin-plate spline modelling of elastic deformations, *Pattern Recognition* **36**(8), 1859–1867.
- Benhammadi, F. and Bey, K. B.: 2013, Embedded fingerprint matching on smart card, *International Journal of Pattern Recognition and Artificial Intelligence* **27**(2), 1350006–1–1350006–20.
- BioLab: 2006, FVC2006: the Fourth International Fingerprint Verification Competition, <http://bias.csr.unibo.it/fvc2006/>.
- Booch, G., Maksimchuk, R. A., Engle, M. E., Young, B. J., Conallen, J. and Houston, K. A.: 2007, *Object-oriented analysis and design with applications*, third edn, Pearson Education, Inc.

- Cao, K., Liu, E. and Jain, A. K.: 2014, Segmentation and enhancement of latent fingerprints : A coarse to fine ridge structure dictionary, *IEEE Transactions on Pattern Analysis and Machine Intelligence (To Appear)* .
- Cappelli, R., Ferrara, M., Franco, A. and Maltoni, D.: 2007, Fingerprint verification competition 2006, *Biometric Technology Today* **15**(7-8), 7–9.
- Cappelli, R., Ferrara, M. and Maio, D.: 2012, A Fast and Accurate Palmprint Recognition System Based on Minutiae, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* **42**(3), 956–962.
- Cappelli, R., Ferrara, M. and Maltoni, D.: 2010, Minutia Cylinder-Code: a new representation and matching technique for fingerprint recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(12), 2128–2141.
- Cappelli, R., Ferrara, M. and Maltoni, D.: 2011, Fingerprint indexing based on Minutia Cylinder-Code, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(5), 1051–1057.
- Cappelli, R., Ferrara, M. and Maltoni, D.: 2012, Minutiae-Based Fingerprint Matching, in C. Liu and V. K. Mago (eds), *Cross Disciplinary Biometric Systems*, Springer Berlin Heidelberg, chapter 5, pp. 117–150.
- Cappelli, R., Maio, D., Maltoni, D., Wayman, J. L. and Jain, A. K.: 2006, Performance evaluation of fingerprint verification systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(1), 3–18.
- Chen, X., Tian, J. and Yang, X.: 2006, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, *IEEE Transaction on Image Processing* **15**(3), 767–776.
- Chen, X., Tian, J., Yang, X. and Zhang, Y.: 2006, An algorithm for distorted fingerprint matching based on local triangle feature set, *IEEE Transactions on Information Forensics and Security* **1**(2), 169–177.

- Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C.: 2009, *Introduction to Algorithms*, 3rd edn, MIT Press.
- Csongrády, T. and Pivarciová, E.: 2011, Spectral biometrical recognition of fingerprints, *Central European Journal of Computer Science* **1**(2), 41–51.
- Dai, J., Feng, J. and Zhou, J.: 2012, Robust and efficient ridge-based palmprint matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(8), 1618–32.
- Decann, B. and Ross, A.: 2012, Can a “poor” verification system be a “good” identification system? A preliminary study, *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, Tenerife, Spain, pp. 31–36.
- Demšar, J.: 2006, Statistical comparisons of classifiers over multiple data sets, *Journal of machine learning research* **7**, 1–30.
- Derrac, J., García, S., Molina, D. and Herrera, F.: 2011, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation* **1**(1), 3–18.
- Dorizzi, B., Cappelli, R., Ferrara, M., Maio, D., Maltoni, D., Houmani, N., Garcia-Salicetti, S. and Mayoue, A.: 2009, Fingerprint and on-line signature verification competitions at ICB 2009, *International Conference on Biometrics (ICB 2009)*, Alghero, Italy, pp. 725–732.
- Fawcett, T.: 2006, An introduction to ROC analysis, *Pattern Recognition Letters* **27**, 861–874.
- Federal Bureau of Investigation: 2012, FBI stories: 30-year-old murder solved – fingerprint technology played key role, <http://www.fbi.gov/news/stories/2012/september/30-year-old-murder-solved>.
- Feng, J.: 2008, Combining minutiae descriptors for fingerprint matching, *Pattern Recognition* **41**(1), 342–352.
- Feng, J., Shi, Y. and Zhou, J.: 2012, Robust and efficient algorithms for separating latent overlapped fingerprints, *IEEE Transactions on Information Forensics and Security* **7**(5), 1498–1510.

- Feng, J., Zhou, J. and Jain, A. K.: 2013, Orientation field estimation for latent fingerprint enhancement, *IEEE transactions on pattern analysis and machine intelligence* **35**(4), 925–40.
- Feng, Y., Feng, J., Chen, X. and Song, Z.: 2006, A novel fingerprint matching scheme based on local structure compatibility, *18th International Conference on Pattern Recognition (ICPR 2006)*, Vol. 4, Hong Kong, China, pp. 374–377.
- Ferrara, M., Maltoni, D. and Cappelli, R.: 2012, Noninvertible Minutia Cylinder-Code representation, *IEEE Transactions on Information Forensics and Security* **7**(6), 1727–1737.
- García, S. and Herrera, F.: 2008, An extension on “ Statistical comparisons of classifiers over multiple data sets ” for all pairwise comparisons, *Journal of machine learning research* **9**, 2677–2694.
- Garg, R. and Rane, S.: 2013, A keypoint descriptor for alignment-free fingerprint matching, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 2994–2998.
- Garris, M. D. and McCabe, R. M.: 2000, NIST Special Database 27: Fingerprint minutiae from latent and matching tenprint images, *Technical report*, National Institute of Standards and Technology, Gaithersburg, MD, USA.
- Ghazvini, M., Sufikarimi, H. and Mohammadi, K.: 2011, Fingerprint matching using genetic algorithm and triangle descriptors, *19th Iranian Conference on Electrical Engineering*, Tehran, Iran, pp. 1–6.
- Guesmi, H., Trichili, H., Alimi, A. M. and Solaiman, B.: 2013, Fingerprint verification system based on curvelet transform and possibility theory, *Multimedia Tools and Applications* pp. 1–20.
- Gupta, A. and Sutton, R.: 2010, Pore sub-features reproducibility in direct microscopic and livescan images-their reliability in personal identification, *Journal of Forensic Sciences* **55**(4), 970–975.



- Gwo-Cheng, C., Shyh-Kang, J. and Shung-Shing, L.: 2007, A fingerprint matching algorithm based on alignment using lpd and gcd minutia descriptors, *First IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2007)*, Crystal City, VA, pp. 1–5.
- Hayfron-Acquah, J. B. and Sarah Gyimah, M.: 2014, Classification and recognition of fingerprints using self organizing maps (som), *International Journal of Computer Science* **11**(1), 153–159.
- Hoyle, K. and Hsiao, M. S.: 2011, *Minutiae triplet-based features with extended ridge information for determining sufficiency in fingerprints*, Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA.
- Huong Thuy, N. T., Huan, H. X. and Ky, N. N.: 2013, An efficient method for fingerprint matching based on local point model, *International Conference on Computing, Management and Telecommunications (ComManTel 2013)*, Ho Chi Minh City, Vietnam, pp. 334–339.
- Jain, A. K. and Feng, J.: 2009, Latent Palmprint Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(6), 1032–1047.
- Jain, A. K. and Feng, J.: 2011, Latent Fingerprint Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(1), 88–100.
- Jain, A. K., Feng, J. and Nandakumar, K.: 2008, On matching latent fingerprints, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops CVPRW '08*, Anchorage, AK, pp. 1–8.
- Jain, A. K., Feng, J. and Nandakumar, K.: 2010, Fingerprint Matching, *Computer* **43**(2), 36–44.
- Jain, A. K., Hong, L., Bolle, R. and Lin, H.: 1997, On-Line Fingerprint Verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(4), 302–314.
- Jain, A. K., Ross, A. A. and Nandakumar, K.: 2011, *Introduction to Biometrics*, Springer.
- Jea, T.-Y.: 2005, *Minutiae-based partial fingerprint recognition*, PhD thesis, State University of New York.

- Jea, T. Y. and Govindaraju, V.: 2005, A minutia-based partial fingerprint recognition system, *Pattern Recognition* **38**(10), 1672–1684.
- Jiang, X. and Yau, W. Y.: 2000, Fingerprint minutiae matching based on the local and global structures, *15th International Conference on Pattern Recognition*, Vol. 2, Barcelona, Spain, pp. 1038–1041.
- Komarinski, P.: 2005, *Automated Fingerprint Identification Systems (AFIS)*, Elsevier Academic Press.
- Kovács-Vajna, Z. M.: 2000, A fingerprint verification system based on Triangular Matching and Dynamic Time Warping, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1266–1276.
- Kumar, R., Chandra, P. and Hanmandlu, M.: 2013, Local Directional Pattern (LDP) based fingerprint matching using SLFNN, *IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Ieee, Shimla, India, pp. 493–498.
- Liu, E., Arora, S. S., Cao, K., Jain, A. K. and Lansing, E.: 2013, A feedback paradigm for latent fingerprint matching, *6th IAPR International Conference on Biometrics (ICB-2013)*, Madrid, Spain, pp. 1–8.
- Liu, E., Jain, A. K. and Tian, J.: 2013, A Coarse to Fine Minutiae-Based Latent Palmprint Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(10), 2307–2322.
- Liu, F. and Zhang, D.: 2014, 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model, *Pattern Recognition* **47**(1), 178–193.
- Liu, F., Zhao, Q. and Zhang, D.: 2011, A novel hierarchical fingerprint matching approach, *Pattern Recognition* **44**(8), 1604–1613.
- Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L. and Jain, A. K.: 2002, FVC2002: Second Fingerprint Verification Competition, *16th International Conference on Pattern Recognition (ICPR 2002)*, Vol. 3, Quebec City, Canada, pp. 811–814.

- Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L. and Jain, A. K.: 2004, FVC2004: Third Fingerprint Verification Competition, *International Conference on Biometric Authentication, ICBA 2004*, Hong Kong, China, pp. 1–7.
- Malathi, S. and Meena, C.: 2011, Improved partial fingerprint matching based on score level fusion using pore and sift features, *International Conference on Process Automation, Control and Computing (PACC 2011)*, Tamilnadu, India, pp. 1–4.
- Maltoni, D., Maio, D., Jain, A. K. and Prabhakar, S.: 2009, *Handbook of fingerprint recognition*, second edn, Springer-Verlag, London.
- Marcialis, G. L., Roli, F. and Didaci, L.: 2013, Multimodal fingerprint verification by score-level fusion: An experimental investigation, *Journal of Intelligent and Fuzzy Systems* **24**(1), 51–60.
- Martínez-Díaz, S. and Carmona-Troyo, J. A.: 2010, Fingerprint verification with non-linear composite correlation filters, *2nd Mexican Conference on Pattern Recognition (MCP R 2010)*, Vol. LNCS 6256, Puebla, Mexico, pp. 90–97.
- Medina-Pérez, M. A., García-Borroto, M., Gutiérrez-Rodríguez, A. and Altamirano-Robles, L.: 2011, Robust fingerprint verification using m-triplets, *International Conference on Hand-Based Biometrics (ICHB 2011)*, Hong Kong, China, pp. 1–5.
- Medina-Pérez, M. A., García-Borroto, M., Gutiérrez-Rodríguez, A. and Altamirano-Robles, L.: 2012a, Improving the multiple alignments strategy for fingerprint verification, *Lecture Notes in Computer Science* **7329**, 147–154.
- Medina-Pérez, M. A., García-Borroto, M., Gutierrez-Rodríguez, A. E. and Altamirano-Robles, L.: 2012b, Improving fingerprint verification using minutiae triplets, *Sensors* **12**, 3418–3437.
- Medina-Pérez, M. A., Gutiérrez-Rodríguez, A. and García-Borroto, M.: 2009, Improving fingerprint matching using an orientation-based minutia descriptor, *Lecture Notes in Computer Science* **5856**, 121–128.

- Medina-Pérez, M. A., Loyola-González, O., Gutierrez-Rodríguez, A. E., García-Borroto, M. and Altamirano-Robles, L.: 2014, Introducing an experimental framework in C # for fingerprint recognition, *Lecture Notes in Computer Science* **8495**, 132–141.
- Mikaelyan, A. and Bigun, J.: 2012, Ground truth and evaluation for latent fingerprint matching, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, Providence, RI, pp. 83–88.
- Neurotechnology Inc.: 2014, Verifinger, <http://www.neurotechnology.com/>. [Online; accessed May 8, 2014].
- Nguyen, T. H., Wang, Y. and Li, R.: 2013, An improved ridge features extraction algorithm for distorted fingerprints matching, *Journal of Information Security and Applications* **18**(4), 206–214.
- Pankanti, S., Prabhakar, S. and Jain, A. K.: 2002, On the individuality of fingerprints, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(8), 1010–1025.
- Parziale, G. and Niel, A.: 2004, A fingerprint matching using minutiae triangulation, *1st International Conference on Biometric Authentication*, Vol. LNCS 3072, Hong Kong, China, pp. 241–248.
- Paulino, A. A., Feng, J. and Jain, A. K.: 2011, Latent fingerprint matching using descriptor-based Hough Transform, *International Joint Conference on Biometrics (IJCB 2011)*, Washington, DC, USA, pp. 1–7.
- Paulino, A., Feng, J. and Jain, A.: 2013, Latent fingerprint matching using descriptor-based hough transform, *IEEE Transactions on Information Forensics and Security* **8**(1), 31–45.
- Paulino, A., Liu, E., Cao, K. and Jain, A.: 2013, Latent fingerprint indexing: Fusion of level 1 and level 2 features, *IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2013)*, Washington, DC, USA, pp. 1–8.
- Prabhakar, S., Ivanisov, A. and Jain, A. K.: 2011, Biometric recognition: Sensor characteristics and image quality, *IEEE Instrumentation & Measurement Magazine* **14**(3), 10–16.

- Puertas, M., Ramos, D., Fierrez, J., Ortega-Garcia, J. and Exposito, N.: 2010, Towards a better understanding of the performance of latent fingerprint recognition in realistic forensic conditions, *20th International Conference on Pattern Recognition (ICPR2010)*, Istanbul, Turkey, pp. 1638–1641.
- Qi, J., Yang, S. and Wang, Y.: 2005, Fingerprint matching combining the global orientation field with minutia, *Pattern Recognition Letters* **26**(15), 2424–2430.
- Ramoser, H., Wachmann, B. and Bischof, H.: 2002, Efficient alignment of fingerprint images, *16th International Conference on Pattern Recognition (ICPR'02)*, Vol. 3, Quebec City, Canada, p. 30748.
- Reisman, J., Uludag, U. and Ross, A.: 2005, Secure Fingerprint Matching with External Registration, *5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005)*, Vol. LNCS 3546, NY, USA, pp. 720–729.
- Ross, A. A., Jain, A. K. and Reisman, J.: 2003, A hybrid fingerprint matcher, *Pattern Recognition* **36**(7), 1661–1673.
- Sankaran, A., Dhamecha, T. I., Vatsa, M. and Singh, R.: 2011, On matching latent to latent fingerprints, *International Joint Conference on Biometrics (IJCB 2011)*, Washington, DC, USA, pp. 1–6.
- Syarif, M. A., Ong, T. S. and Tee, C.: 2014, Fingerprint recognition based on multi-resolution histogram of gradient descriptors, *Lecture Notes in Electrical Engineering* **291**, 189–196.
- Tan, X. and Bhanu, B.: 2006, Fingerprint matching by genetic algorithms, *Pattern Recognition* **39**(3), 465–477.
- Tico, M. and Kuosmanen, P.: 2003, Fingerprint matching using an orientation-based minutia descriptor, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8), 1009–1014.
- Tong, X., Huang, J., Tang, X. and Shi, D.: 2005, Fingerprint minutiae matching using the adjacent feature vector, *Pattern Recognition Letters* **26**, 1337–1345.

- Udupa U, R., Garg, G. and Sharma, P.: 2001, Fast and accurate fingerprint Verification, *3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA 2001)*, Vol. LNCS 2091, Halmstad, Sweden, pp. 192–197.
- Vatsa, M.: 2014, Quality induced multiclassifier fingerprint verification using extended feature set, in J. Scharcanski, H. Proença and E. Du (eds), *Signal and Image Processing for Biometrics*, Vol. 292 of *Lecture Notes in Electrical Engineering*, Springer Berlin Heidelberg, pp. 231–254.
- Vatsa, M., Singh, R., Noore, A. and Morris, K.: 2011, Simultaneous latent fingerprint recognition, *Applied Soft Computing* **11**(7), 4260–4266.
- Wahby Shalaby, M. A. and Omair Ahmad, M.: 2013, A multilevel structural technique for fingerprint representation and matching, *Signal Processing* **93**(1), 56–69.
- Wang, R., Ramos, D. and Fierrez, J.: 2012, Improving Radial Triangulation-based Forensic Palmprint Recognition According to Point Pattern Comparison by Relaxation, *2012 5th IAPR International Conference on Biometrics (ICB)*, pp. 427–432.
- Watson, C. I.: 2001, NIST Special Database 14: Mated fingerprint cards pairs 2 version 2, *Technical report*, National Institute of Standards and Technology, Gaithersburg, MD, USA.
- Watson, C. I. and Wilson, C. L.: 1992, NIST special database 4, *Technical report*, National Institute of Standards and Technology, Gaithersburg, MD, USA.
- Wen, W., Qi, Z., Li, Z., Zhang, J., Gong, Y. and Cao, P.: 2013, A robust and efficient minutia-based fingerprint matching algorithm, *2nd IAPR Asian Conference on Pattern Recognition (ACPR 2013)*, Naha, Japan, pp. 201–205.
- Xu, W., Chen, X. and Feng, J.: 2007, A robust fingerprint matching approach: growing and fusing of local structures, *2nd International Conference on Biometrics (ICB 2007)*, Vol. LNCS 4642, Seoul, Korea, pp. 134–143.

- Yoon, S., Feng, J. and Jain, A.: 2011, Latent fingerprint enhancement via robust orientation field estimation, *2011 International Joint Conference on Biometrics (IJCB)*, Washington, DC, USA, pp. 1–8.
- Yoon, S., Feng, J. and Jain, A. K.: 2010, On latent fingerprint enhancement, in B. V. K. Vijaya Kumar, S. Prabhakar and A. A. Ross (eds), *Proceedings of SPIE, Biometric Technology for Human Identification VII*, Vol. 7667, Orlando, Florida, pp. 766707–766707–10.
- Yu, K. D., Na, S. and Choi, T. Y.: 2005, A fingerprint matching algorithm based on radial structure and a structure-rewarding scoring strategy, *5th International Conference on Audio-and Video-Based Biometric Person Authentication, AVBPA 2005*, Vol. LNCS 3546, NY, USA, pp. 656–664.
- Zhao, Q., Zhang, D., Zhanga, L. and Luoa, N.: 2010, High resolution partial fingerprint alignment using pore-valley descriptors, *Pattern Recognition* **43**(3), 1050–1061.
- Zheng, J.-D., Gao, Y. and Zhang, M.-Z.: 2009, Fingerprint matching algorithm based on similar vector triangle, *2nd International Congress on Image and Signal Processing, CISP '09*, Tianjin, China, pp. 1–6.
- Zhou, R., Zhong, D. and Han, J.: 2013, Fingerprint identification using SIFT-based minutia descriptors and improved all descriptor-pair matching., *Sensors* **13**(3), 3142–56.