

Centro de Investigación en Matemáticas, A.C.

CIMAT

Análisis Espectral de Grafos
Una Aplicación Biométrica

Tesis presentada para la obtención de grado de

Maestro en Ciencias

con Especialidad en Computación
y Matemáticas Industriales

P R E S E N T A:

Ulises Tirado Zatarain

Comité de evaluación:

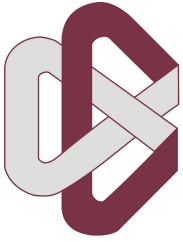
Dr. Jean-Bernard Hayet

Dr. Alonso Ramírez Manzanares

Director de tesis:

Dr. Mariano J.J. Rivera Meraz

Guanajuato, Gto., México. Febrero, 2015.



Centro de Investigación en Matemáticas, A.C.

CIMAT

Análisis Espectral de Grafos
Una Aplicación Biométrica

Tesis presentada para la obtención de grado de

Maestro en Ciencias

con Especialidad en Computación
y Matemáticas Industriales

P R E S E N T A:

Ulises Tirado Zatarain

Director de tesis:

Dr. Mariano J.J. Rivera Meraz

Dedico este trabajo a mi familia y amigos que nunca dejaron de apoyarme.

RESUMEN

La teoría de grafos es un área de las matemáticas que se encarga del estudio de ciertas propiedades de estructuras llamadas grafos o gráficas. Estas estructuras se utilizan para modelar diversos problemas en áreas como Ingeniería, Física, Economía y por supuesto Ciencias de la Computación.

La biometría es el estudio de medidas para distinguir diversas características anatómicas o de comportamiento para identificar de forma automática a los seres humanos. Estas características son conocidas como identificadores biométricos y pueden ser: huellas dactilares, el rostro (en dos y tres dimensiones), el iris del ojo, la retina del ojo, la voz, entre otras.

Una huella dactilar es la marca que se produce al contacto de algún dedo de la mano o el pie con alguna superficie. Las distintas formas que se presentan en estas marcas, son generadas por fuerzas de elasticidad en la piel desde el desarrollo embrionario. Las huellas dactilares poseen ciertas características o rasgos que las hacen únicas en cada individuo.

AGRADECIMIENTOS

Quiero agradecer a CIMAT, AC por brindarme la oportunidad de realizar mis estudios de maestría, así como a CONACyT por otorgar por segunda ocasión beneficios económicos para poder sustentar mis estudios profesionales. Agradezco a todos mis profesores que me acompañaron en el andar, pero principalmente a Jean-Bernard Hayet, Claudia Esteves Jaramillo, Johan van Horebeek y José Luis Marroquín Zaleta, quienes no sólo me transmitieron gran parte de los conocimientos básicos que sustentan este trabajo sino que además me brindaron su amistad. Aprovechando este espacio, quiero agradecer también a Rafael Ayala, ya que sin él no me hubiera enterado de la existencia de esta gran institución académica.

De mis profesores, quiero dar un agradecimiento muy especial a mi asesor Dr. Mariano José Juan Rivera Meraz por apoyarme no solo, por darme la oportunidad de trabajar con él sino por contagiarme de su ingenio para la resolución de problemas e impartirme dos de los cursos más importantes para el desarrollo de este trabajo, además de su paciencia y determinación para explicarme a detalle cada uno de los temas involucrados y no involucrados en este trabajo.

Muchos fueron los amigos que conocí en este caminar, a los cuales les quiero agradecer por los momentos de distracción y recreación que aliviaron muchos momentos de estrés, así como lo importante y difícil que es el desarrollo de una tesis, por mencionar algunos David Quiroz, Paulina Salcedo, Marcel Stöckli, Alberto Ramírez, Alfonso Cabrera, David Reynoso, Verónica Suaste, Caleb Méndez, Luis Vargas, Jonathan Patiño, Alejandro Bustos, Teresa Rodríguez, Mariana Ramírez, Josué Vázquez, Francisco Arellano y muchos otros más. Agradezco también a quienes fueron mis compañeros de cubículo y ayudantes: David Jacobo, Rafael Robledo, Carlos Zubieta, Guillermo Laguna, Mauricio García, Iván Márquez, Cristóbal García, Francisco Madrigal, Ramón Aranda, Roberto Cruz y Hugo Carlos Martínez.

A mis compañeros de generación con los que compartí muchos desvelos, viajes, experiencias y retos: Ángel Martínez, Karen Loya, Miguel Covarrubias, Roberto Guzmán, Arturo Suárez, Noé Faurrieta, Gilberto Chávez, Fernando Fimbres, Marcos Calderón, pero de una forma muy particular a Guillermo Rangel por compartir departamento conmigo, su apoyo y motivaciones durante todo este tiempo.

Por su apoyo incondicional en momentos muy difíciles quiero agradecer muy especialmente a Sandra Alfaro y su familia, así como a Aldo Ortiz, Salvador Meneses, Román Aranda, Hugo Dueñas y Asiel Corpus.

Finalmente, quiero agradecer a mis padres Aristeo y Delia, así como a mis hermanos Alejandra, María de los Ángeles, Karla, Alma Graciela, Germán, Aristeo y sus respectivas familias que han formado. A todos quiero decir, simplemente, muchas gracias.

ÍNDICE GENERAL

i	PRELIMINARES	1
1	INTRODUCCIÓN	3
1.1	Conceptos básicos	3
1.1.1	Huellas dactilares	4
1.1.2	Procesamiento digital de señales	4
1.1.3	Reconocimiento estadístico de patrones	6
1.1.4	Teoría de grafos	7
1.1.5	Notación adicional	9
1.2	Definición del problema	10
1.2.1	Análisis espectral de grafos	11
1.2.2	Análisis de Fourier	11
ii	MARCO TEÓRICO	13
2	TRABAJO PREVIO	15
2.1	Método elemental basado en minucias	15
2.2	Banco de filtros de Gabor	18
2.2.1	Localización del punto de referencia	20
2.2.2	Proceso de filtrado	22
2.2.3	Comparación	23
3	MÉTRICAS PARA LA COMPARACIÓN DE HUELLAS DACTILARES	25
3.1	Métodos basados en grafos	25
3.1.1	Método de la matriz de sociabilidad	29
3.1.2	Método del esquema espectral	32
3.2	Método basado en Análisis de Fourier	34
iii	MARCO EXPERIMENTAL	39
4	IMPLEMENTACIÓN Y EXPERIMENTOS	41
4.1	Pruebas con datos artificiales	41
4.2	Pruebas con datos reales	43
4.3	Resultados	44
5	CONCLUSIONES	49
5.1	Trabajo futuro	50
5.2	Recomendaciones	51
	BIBLIOGRAFÍA	53

ÍNDICE DE FIGURAS

Figura 1.1.1	Huella dactilar. En esta imagen se pueden observar las crestas en color negro, mientras que los valles son los espacios en blanco.	4
Figura 1.1.2	Minucias de una huella dactilar	5
Figura 1.1.3	Representación gráfica de un grafos. Los vértices son representados mediante círculos y las aristas son las conexiones entre ellos.	8
Figura 2.2.1	Localización del punto de referencia de la imagen de una huella dactilar para determinar su punto de referencia.	18
Figura 2.2.2	Regiones de integración de las intensidades de \mathcal{E} para calcular $\mathcal{A}(i, j)$	20
Figura 2.2.3	Kernels para los filtros de Gabor entonados a una frecuencia $\omega = \frac{1}{10}$ en 8 diferentes orientaciones.	23
Figura 3.1.1	Un grafo generado por un conjunto de minucias con su proyección de los datos de la matriz de sociabilidad sobre los primeros dos componentes principal.	32
Figura 3.1.2	Grafo bipartito generado por dos huellas dactilares. Los vértices de cada conjunto de minucias se muestran en color diferente.	33
Figura 3.2.1	Banco de filtros de Gabor en 8 orientaciones y entonado a 3 frecuencias	35
Figura 3.2.2	Extracción de características por medio de magnitud y fase	36
Figura 4.1.1	Datos simulados de minucias	42
Figura 4.2.1	Huellas dactilares de dos sensores diferentes	43
Figura 4.3.1	Resultados del Algoritmo basado en la Matriz de Sociabilidad	45
Figura 4.3.2	Resultados del Algoritmo basado en el esquema espectral	46
Figura 4.3.3	Resultado del Algoritmo basado en bancos de filtros	47
Figura 5.1.1	Huellas de la base de datos NIST-SD27	50

ÍNDICE DE CUADROS

Cuadro 4.3.1	Estadísticas sobre el comportamiento del Algoritmo 8	45
Cuadro 4.3.2	Estadísticas sobre el comportamiento del Algoritmo 9	47

Parte I

PRELIMINARES

INTRODUCCIÓN

La **biometría** es el estudio de métricas para distinguir diversas características anatómicas o de comportamiento con el fin de identificar de forma automática a los seres humanos. Estas características son conocidas como identificadores biométricos y pueden ser: las huellas dactilares, el rostro (en dos y tres dimensiones), el iris del ojo, la retina del ojo, la voz, entre otras. Debido a que los identificadores biométricos no pueden ser extraviados, falsificados o compartidos fácilmente, en los últimos años se han considerado como un método de identificación confiable y el uso de sistemas biométricos ha ido creciendo notablemente en diversas áreas de aplicación como pueden ser: autenticación en sistemas electrónicos, acceso a instalaciones, identificación criminal, etcétera. Para esto, es necesario desarrollar algoritmos para la comparación de dichas características por medio de la computadora que sean eficientes y robustos dependiendo de la aplicación. Esto se logra a través de la definición de métricas de similitud y de distancia que permiten obtener valores simples para ser comparados.

Durante el verano del año 1891, el croata naturalizado argentino Iván Vučetić (Juan Vucetich) desarrolló un sistema de identificación de personas mediante huellas dactilares con la finalidad de esclarecer un crimen. El 1 de Septiembre de 1891 se estableció como el Día Mundial de la Dactiloscopía debido a que Vucetich realizó las primeras fichas dactilares con las huellas de 23 procesados por el departamento de policía de Buenos Aires, Argentina. Jain *et al.* (2002) plantea la unicidad de la huella dactilar desde un punto de vista criminológico, mientras que Kücken (2007) da un enfoque más biológico fundamentado en la teoría de formación de patrones durante el desarrollo embrionario de los individuos. Las distintas formas de las huellas dactilares son el resultado de fuerzas elásticas no lineales en reacción a células en la dermis y la epidermis. Según Kücken y Newell (2005) las huellas digitales son únicas debido a que una pequeña variación en las fuerzas elásticas antes mencionadas puede ocasionar serios cambios en el resultado final del patrón ya que estas fuerzas son no lineales. Esta teoría tiende a ser matemáticamente razonable y físicamente aceptable.

1.1 CONCEPTOS BÁSICOS

Con la finalidad de familiarizar al lector, primeramente daremos a una introducción a conceptos técnicos de distintas áreas del conocimiento que se manejarán a lo largo de este documento.



Figura 1.1.1: Huella dactilar. En esta imagen se pueden observar las crestas en color negro, mientras que los valles son los espacios en blanco.

1.1.1 *Huellas dactilares*

Una **huella dactilar** es la marca que se produce al contacto de algún dedo de la mano o del pie con alguna superficie. Las distintas formas que se presentan en estas marcas, son generadas por fuerzas de elasticidad en la piel desde el desarrollo embrionario.

Las huellas dactilares poseen ciertas características o rasgos que las hacen únicas en cada individuo. Estas características están determinadas por las curvas que la forman, de las cuales hay de dos tipos. Las **crestas** son las partes altas del relieve de la piel en el dedo y las que ocasionan la impresión de la huella dactilar en las superficies, mientras que las partes bajas del relieve de la piel en el dedo se denominan **valles**.

A cada punto de intersección o cambio de dirección entre crestas o valles se le conoce como **minucia**. De estos puntos nos interesarán particularmente dos cosas, su **localización** en coordenadas bidimensionales y la dirección en la que se dio el cruce o cambio, a la cual llamaremos **orientación**, es decir, una medida angular (como se puede ver en la Figura 1.1.2).

Entenderemos por **usuario** al individuo a la cual pertenece alguna huella dactilar. Al momento de comparar un par de huellas, obtendremos un **puntaje** que nos permitirá determinar que tan distintas son las huellas y posteriormente decidir si pertenecen al mismo usuario.

1.1.2 *Procesamiento digital de señales*

En la comparación de huellas dactilares resulta natural el proceso de imágenes o señales digitales, por lo que primeramente necesitaremos introducir algunos conceptos básicos de esta área.

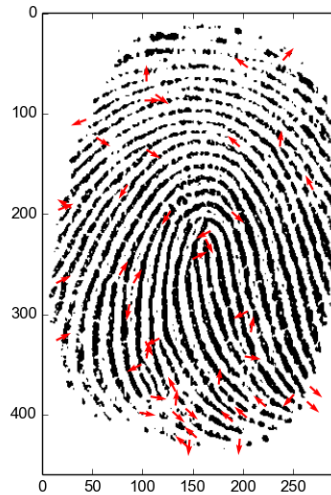


Figura 1.1.2: Minucias de una huella dactilar

Las minucias son los puntos de cruce o cambios de dirección de estas curvas. En esta imagen se puede observar su posición y orientación que está definida por las flechas en color rojo.

Una **señal** es la medición por medio de un sensor de alguna magnitud física en el tiempo o el espacio y existen básicamente de dos tipos. Las **señales analógicas** son representadas por una función con dominio continuo, es decir, una función del tipo $f : \mathbb{R}^m \mapsto \mathbb{R}^n$; mientras que las **señales digitales** se representan por funciones en dominio discreto, es decir, en estas últimas se tiene un muestreo de los valores de la magnitud física en tiempos o posiciones usualmente con valores enteros, por lo que la función que las representa es del tipo $f : \mathbb{Z}^m \mapsto \mathbb{Z}^n$. En áreas como electricidad y electrónica es muy común realizar procesamiento de señales reales en una sola dimensión, dicho de otra forma, con funciones del tipo $f : \mathbb{Z} \mapsto \mathbb{Z}$.

Entenderemos por **imagen digital** o simplemente imagen a aquella señal digital que dados valores de posición espacial provee valores que representan la intensidad de luz para dichas posiciones. En el caso de imágenes en escala de grises, se obtiene un valor escalar entero en el intervalo $[0, 255]$ para cada posición, mientras que para las imágenes en color se obtienen tres valores en el mismo intervalo, de manera que cada valor representa la intensidad de cada color primario (rojo, verde y azul). En este trabajo se manipularán **imágenes escalares** en dos dimensiones, es decir, imágenes en escala de grises que representaremos por funciones del tipo $I : \mathbb{Z}^2 \mapsto [0, 255]$. En estas imágenes cada una de estas muestras de intensidad se conoce como **pixel**.

En el análisis de señales digitales existen perturbaciones o datos de alguna manera incoherentes en las señales, las cuales son conocidas como **ruido**. Esto puede ser ocasio-

nado por diversos factores como puede ser la precisión del sensor o simplemente algunos factores naturales.

Para eliminar o mitigar el ruido se utilizan procedimientos denominados **filtros** que manipulan la señal. Usualmente consisten en analizar una vecindad de cada pixel, de tal forma que se determina el valor real (sin ruido) del pixel por medio de las muestras alrededor de él. A esta vecindad se le conoce comúnmente como **ventana** (Rivera *et al.*, 2012). Para aplicar un filtro se utiliza la convolución entre una función h , denominada **kernel** y definida en la ventana y la función que representa la señal; para una imagen I y en el dominio espacial discreto esto está dado por:

$$[I * h](\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n I(\mathbf{x}) h(\mathbf{x} - \mathbf{y}) \quad (1.1.1)$$

donde $\mathbf{x} = [x_1 \ x_2]^\top$, $\mathbf{y} = [i \ j]^\top$ y h representa un filtro invariante espacialmente.

Una característica muy importante de las señales es su **frecuencia**, es decir, su cantidad de repeticiones por unidad espacial o de tiempo. En el caso de las imágenes escalares, es la cantidad de veces que una intensidad en grises se repite por cada pixel. La importancia característica de las señales radica en que es posible hacer un procesamiento más eficiente en el **dominio de la frecuencia** en lugar del dominio espacial o temporal por medio de la **Transformada Discreta de Fourier** (DFT, por sus siglas en inglés), la cual está dada por:

$$\mathcal{F}(\boldsymbol{\omega}) = \sum_{x_1=1}^m \sum_{x_2=1}^n f(\mathbf{x}) e^{\frac{2\pi i}{mn} \boldsymbol{\omega}^\top \mathbf{x}} \quad (1.1.2)$$

donde $\mathbf{x} = [x_1 \ x_2]^\top$ y f es la función de la cual se toma la transformada.

Por medio de la DFT, es posible construir muchos tipos de filtros gracias al siguiente teorema importante:

Teorema 1.1 (Teorema de Convolución Discreta). Sean $f, h : \mathbb{Z}^m \mapsto \mathbb{R}$ y \mathcal{F} y \mathcal{H} sus transformadas discretas de Fourier respectivamente, entonces:

$$g(\mathbf{x}) = [f * h](\mathbf{x}) \Rightarrow \mathcal{G}(\boldsymbol{\omega}) = \mathcal{F}(\boldsymbol{\omega}) \mathcal{H}(\boldsymbol{\omega}) \quad (1.1.3)$$

En nuestro desarrollo experimental en particular utilizaremos los llamados **Filtros de Gabor**, los cuales consisten en un **filtro pasa banda**, es decir, filtros que permiten el paso de un rango de frecuencias en una señal, mientras que las intensidades que están fuera de ese rango son suavizadas o anuladas.

1.1.3 Reconocimiento estadístico de patrones

Tomaremos un conjunto de n **observaciones** $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ para formar una **matriz de datos** $\mathbf{X} \in \mathbb{R}^{n \times d}$ donde la i -ésima fila corresponderá con la observación \mathbf{x}_i^\top y denominaremos **característica** a cada una de las d entradas de cada observación, de forma que la i -ésima observación será $\mathbf{x}_i = [x_{i1} \ \dots \ x_{id}]^\top$.

Un aspecto importante de estas características es su **escala**, por lo que recurriremos frecuentemente a **normalizar** nuestras matrices de datos por columnas, esto es, que nuestros datos estén de alguna manera acotados. Típicamente lo haremos para que estén en el intervalo $[0, 1]$.

La técnica estadística más utilizada en nuestros experimentos será el **Análisis de Componentes Principales (PCA)**, esto es, una descomposición espectral a partir de la matriz de covarianza de un conjunto de observaciones, la cual está dada por:

$$\text{cov}(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \quad (1.1.4)$$

donde $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ es la media de las observaciones. Posteriormente, se busca la dirección en la que la variabilidad de los datos es máxima, es decir, un vector unitario $\boldsymbol{\ell}^*$ que maximice la variabilidad. Para esto, es necesario resolver el siguiente problema de optimización cuadrática con restricciones:

$$\begin{aligned} \max_{\boldsymbol{\ell}} f(\boldsymbol{\ell}) &= \boldsymbol{\ell}^\top \text{cov}(\mathbf{X}) \boldsymbol{\ell} \\ \text{s. a: } \|\boldsymbol{\ell}\|_2 &= 1 \end{aligned} \quad (1.1.5)$$

Este problema es bastante complicado, sin embargo, debido a que $\text{cov}(\mathbf{X})$ es una matriz simétrica, es equivalente a maximizar un Cociente de Rayleigh sin restricciones:

$$\max_{\boldsymbol{\ell}} f(\boldsymbol{\ell}) = \frac{\boldsymbol{\ell}^\top \text{cov}(\mathbf{X}) \boldsymbol{\ell}}{\boldsymbol{\ell}^\top \boldsymbol{\ell}} \quad (1.1.6)$$

La solución al modelo (1.1.6) ya es conocida debido a que es un cociente de Rayleigh, y $\boldsymbol{\ell}^* = \boldsymbol{\varphi}_1$, es decir, es el eigenvector correspondiente al eigenvalor más grande en valor absoluto λ_1 , lo cual puede ser demostrado mediante la desigualdad de Cauchy-Schwarz. Más aun, por construcción $\lambda_1 \geq 0$, es decir, $\text{cov}(\mathbf{X})$ es semidefinida positiva.

Entonces, si tomamos las proyecciones de los datos sobre las k direcciones de máxima variabilidad, esto es las correspondientes a los eigenvalores más grandes en valor absoluto de la matriz de covarianza (Bishop, 2006), obtendremos una versión más reducida de las observaciones sin perder información relevante.

1.1.4 Teoría de grafos

La teoría de grafos es un área de las matemáticas que se encarga del estudio de ciertas propiedades de estructuras llamadas grafos o gráficas. Estas estructuras se utilizan para modelar diversos problemas en áreas como Ingeniería, Física, Economía y por supuesto Ciencias de la Computación.

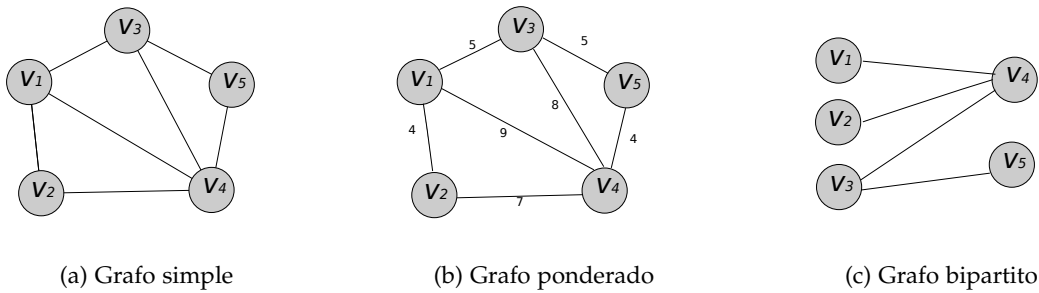


Figura 1.1.3: Representación gráfica de un grafos. Los vértices son representados mediante círculos y las aristas son las conexiones entre ellos.

Un **grafo** o gráfica (a veces también conocidos como redes) es una colección de objetos llamados **vértices** o nodos los cuales están relacionados por medio de **aristas** o arcos.

Formalmente y de la forma más simple, se pueden definir los grafos como un par de conjuntos, el primero V , contiene a un conjunto de vértices, mientras que el segundo E , contiene las aristas, es decir:

$$G = (V, E)$$

donde:

- V es un conjunto de etiquetas de los nodos. Es habitual usar números naturales $\{1, 2, \dots, n\}$, letras minúsculas $\{a, b, \dots\}$ o letras con subíndice $\{v_1, v_2, \dots, v_n\}$, siendo n la cantidad de nodos en el grafo.
- $E \subseteq V \times V$ es un conjunto de pares que denota la relación entre los vértices. Por ejemplo, para el grafo de la figura 1.1.3a el conjunto de aristas está dado por:

$$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4), (v_3, v_4), (v_3, v_5), (v_4, v_5)\}$$

Otra forma matemática de representar los grafos es por medio de una **matriz de adyacencia**, esto es, una matriz binaria $\mathbf{A} \in \{0, 1\}^{n \times n}$, donde:

$$a_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{en otro caso} \end{cases}$$

En la teoría de grafos existe toda una taxonomía, de tal forma que se describan algunas características generales de los mismos. A continuación definiremos algunas de los distintos tipos de grafos.

Un grafo **dirigido** es aquel en el que las aristas tienen una orientación, es decir, en este tipo de grafos puede existir una relación entre el vértice v_i y el vértice v_j pero no

necesariamente de forma recíproca. Dicho de otra forma, (v_i, v_j) es diferente de (v_j, v_i) . Un grafo que no tiene orientaciones en sus aristas se conoce como **no dirigido**. Es fácil notar que la matriz de adyacencia correspondiente a un grafo no dirigido es simétrica.

Se conoce como **grafo ponderado** a aquél en el que las aristas tienen asociado un peso w_{ij} . En este caso, en lugar de una matriz de adyacencia, el grafo tiene asociada una **matriz de pesos W** , la cual puede contener entradas reales, enteras e incluso complejas dependiendo de la aplicación que se esté modelando. De igual forma, dependiendo de la aplicación, el grafo puede ser dirigido y además ponderado, es decir, que el peso asociado a la arista (v_i, v_j) no es necesariamente el mismo que el asociado a la arista (v_j, v_i) , o bien, $W \neq W^T$.

Un grafo cuya matriz de adyacencia o de pesos contiene muchas entradas nulas se conoce como **grafo disperso** o ralo mientras que si su matriz asociada contiene pocas entradas nulas se conoce como **grafo denso**.

Finalmente, un **grafo bipartito** es un grafo en el que los vértices se pueden separar en dos conjuntos mutuamente excluyentes y sus aristas van de un conjunto a otro, es decir, $V = V_A \cup V_B$, $V_A \cap V_B = \emptyset$ y $E \subseteq V_A \times V_B$. En la figura 1.1.3c se muestra un ejemplo sencillo de este tipo de grafos. Para más información sobre teoría de grafos consultar (Estrada, 2012).

1.1.5 Notación adicional

Debido a que la literatura consultada para desarrollar este trabajo es muy diversa en cuanto a autores y distintas áreas de la ciencia, es necesario uniformizar la notación, de manera que se puedan reducir la cantidad de ambigüedades de forma significativa. Representaremos los vectores con letras minúsculas en negritas, mientras que las matrices y conjuntos de vectores aparecerán escritos con letras mayúsculas en negritas. Resultará muy común utilizar un conjunto de números naturales $\{1, \dots, n\}$, por lo que de forma práctica nos referiremos a este tipo de conjuntos por medio de $[n]$ (Sussman *et al.*, 2012).

En el contexto de las minucias, comúnmente trabajaremos con dos tipos de normas, $\|\cdot\|_s$ denotará una que toma en cuenta sólo la parte espacial de dicha minucia, mientras que $\|\cdot\|_D$ se utilizará para la parte angular o de dirección.

Cuando una variable cambia de valor en un algoritmo iterativo se utilizarán subíndices entre paréntesis para el valor en la r -ésima iteración de dicha variable, por ejemplo $z_{(r)}$.

Se utilizará el símbolo $\lfloor x \rfloor$ para denotar la parte entera por abajo de un valor x . Además de los operadores \bmod y div para el residuo y la división entera respectivamente. Es importante resaltar, que aunque *a priori* \bmod y div se utiliza en el contexto de los números enteros, en los capítulos subsecuentes también lo utilizaremos en contexto de números reales (Jain *et al.*, 1999), por ejemplo, $12.5 \text{ div } 5.5 = 2$, de manera similar $\frac{7\pi}{8} \bmod \frac{\pi}{4} = \frac{\pi}{8}$.

En el proceso de binarización y construcción de matrices de adyacencia se utilizará el símbolo $\mathbb{1}_S$ para denotar la función indicadora, de manera que:

$$\mathbb{1}_S(x) = \begin{cases} 1 & x \in S \\ 0 & \text{en otro caso} \end{cases}$$

No debe confundirse este símbolo con $\mathbf{1}$, el cual indica un vector columna con todas sus entradas en valor 1. La dimensión de este vector será determinada con el contexto, es decir, de forma que la operación en la que se esté usando esté correctamente definida.

1.2 DEFINICIÓN DEL PROBLEMA

De la forma más simple, el problema consiste en definir una función de distancia que tome como entrada dos huellas dactilares y arroje de salida un valor por medio del cual sea posible determinar si ambas huellas son del mismo individuo o no.

Al momento de realizar la comparación de dos huellas dactilares es necesario considerar varios factores (principalmente geométricos) que pueden provocar que el resultado obtenido sea incorrecto, por lo cual los métodos propuestos deben de ser invariantes a ellos. Estos factores son los siguientes:

- Escala
- Traslación
- Rotación
- Cizallamiento
- Ruido

Como se mostrará en el Capítulo 3 de este trabajo se abordará el problema de comparación de huellas dactilares con enfoques distintos, dos de ellos basados en **análisis espectral de grafos** y uno más basado en procesamiento de imágenes por medio de **análisis de Fourier**. Con ambos enfoques y de forma general, nuestro problema se podrá atacar definiendo lo siguiente:

- **Criterio de normalización.** Debido a que las técnicas utilizadas en este trabajo involucrarán métodos estadísticos, es importante definir criterios de normalización de datos, los cuales nos permitirán acotar el problema y conservar los datos en una determinada escala.
- **Función de distancia.** Es necesario definir una función escalar real que arroje un valor escalar con el cual podamos medir la diferencia entre dos huellas dactilares.

- **Umbral.** Finalmente, para realizar la comparación de la distancia obtenida es necesario un valor real, por medio del cual podamos discernir si ambas huellas corresponden o no al mismo sujeto.

1.2.1 *Análisis espectral de grafos*

En estos enfoques, se utilizarán las minucias extraídas de las imágenes de las huellas dactilares. Sin embargo, no se tratará el proceso de extracción de estos puntos de interés, es decir, como entrada tendremos una matriz de datos en la que cada renglón contendrá la información espacial, angular y confiabilidad de las minucias. Estos puntos de interés tienen las siguientes componentes:

- **Espacial.** Describe la localización del punto en dos dimensiones, es decir, las coordenadas $(x, y) \in \mathbb{Z}^{+2}$ en el dominio espacial de la imagen original de la huella dactilar.
- **Angular.** Describe la orientación de las minucias, la cual será denotada por $\alpha \in [0, 360)$.
- **Calidad.** Es el nivel de confianza que tiene el punto como observación, lo denotaremos por medio de $q \in (0, 100]$.

En este caso, nuestros algoritmos recibirán como entrada dos matrices de datos con dimensiones $m \times 4$ y $n \times 4$ respectivamente, en las cuales cada renglón contendrá la información de una minucia u observación con las componentes descritas anteriormente.

1.2.2 *Análisis de Fourier*

En este enfoque, utilizaremos bancos de Filtro de Gabor para extraer las características de las huellas digitales en el dominio de la frecuencia.

A este vector de características lo conoceremos como **FingerCode** (Jain *et al.*, 2000), o dicho de otra forma un conjunto de características que nos describirán una observación. Para esto tendremos como entrada las imágenes escalares de las dos huellas como tal, es decir, no se le realizará un preprocesamiento previo a las imágenes como en el caso de los dos enfoques anteriores.

Veremos como el uso de bancos de Filtros de Gabor garantiza un algoritmo más robusto con respecto a las transformaciones geométricas mencionadas al inicio de esta sección. Sin embargo, también notaremos que el procesamiento de las imágenes resulta más exhaustivo y por lo tanto más costoso computacionalmente.

Parte II

MARCO TEÓRICO

TRABAJO PREVIO

Existen un gran número de métodos para realizar comparaciones entre huellas dactilares. Estos tienen algunas diferencias que se deben considerar, por ejemplo: algunos son más veloces que otros, pero menos robustos; otros son más robustos pero más lentos.

Durante el desarrollo de este trabajo se utilizaron tres tipos de métricas, basados en dos enfoques distintos: el primero basado en puntos de interés existentes en cada una de las huellas y el segundo basado en características en el espacio de Fourier de las imágenes de las huellas.

Para el primer enfoque se consideran características espaciales de las huellas dactilares, de forma que se extraen las minucias de las imágenes de las huellas y posteriormente se utilizarán conceptos de teoría de grafos con las minucias y sus relaciones entre sí.

El segundo enfoque es tomar características de la imagen de la huella en el dominio de la frecuencia, es decir, después de aplicar la transformada de Fourier.

2.1 MÉTODO ELEMENTAL BASADO EN MINUCIAS

Básicamente el problema se plantea como una especie de registro de imágenes ya que se tiene que encontrar una correspondencia entre pares de puntos de dos conjuntos, de tal forma que se minimice una métrica de error y se obtenga el mejor ajuste posible. A continuación abordaremos el método más elemental (Maltoni *et al.*, 2009), el cual de momento no considera la componente de calidad o confianza de las minucias extraídas.

Dados dos conjuntos \mathcal{T} y \mathcal{U} que contienen las minucias de dos huellas dactilares, representadas mediante tercias de la forma $[x \ y \ \alpha]$, en las que x y y denotan la localización de la minucia en la imagen de la huella dactilar y α su orientación, de esta manera:

$$\mathcal{T} = \{t_1, t_2, \dots, t_M\} \quad t_i = \begin{bmatrix} x_i & y_i & \alpha_i \end{bmatrix}^T \quad i \in [M] \quad (2.1.1a)$$

$$\mathcal{U} = \{u_1, u_2, \dots, u_N\} \quad u_j = \begin{bmatrix} x'_j & y'_j & \alpha'_j \end{bmatrix}^T \quad j \in [N] \quad (2.1.1b)$$

donde M y N representan la cantidad de minucias en \mathcal{T} y \mathcal{U} , respectivamente.

Así, podemos hacer corresponder una minucia $t_i \in \mathcal{T}$ con una $u_j \in \mathcal{U}$ definiendo métricas o normas $\|\cdot\|_S$ y $\|\cdot\|_D$ para sus componentes espaciales y angulares, respectiva-

mente. Dados dos umbrales ρ_0 y α_0 podemos decir que \mathbf{t}_i y \mathbf{u}_j son correspondientes si se cumplen los siguientes criterios:

$$\|\mathbf{u}_j - \mathbf{t}_i\|_S = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq \rho_0 \quad (2.1.2a)$$

$$\|\mathbf{u}_j - \mathbf{t}_i\|_D = \min(|\alpha'_j - \alpha_i|, 360 - |\alpha'_j - \alpha_i|) \leq \alpha_0 \quad (2.1.2b)$$

En el criterio de la distancia angular definido en (2.1.2b), podemos observar que tomamos un mínimo de dos valores, esto es para obtener la verdadera diferencia que hay entre dos ángulos en forma circular. Por ejemplo, entre 350° y 5° en realidad sólo hay una diferencia de 15° y no de 345° como comúnmente se pensaría.

Entonces, para alinear dos huellas dactilares, el problema consiste realmente en maximizar la cantidad de pares de minucias correspondientes, es decir, encontrar una transformación geométrica que considere los factores mencionados en la Sección (1.2).

Por lo anterior, necesitamos una función $v(\cdot)$, la cual mapea una minucia $\mathbf{u}_j \in \mathcal{U}$ en una \mathbf{u}'_j , de acuerdo a una transformación geométrica dada; por ejemplo, considerando un desplazamiento $[\Delta x \ \Delta y]^\top$ y una rotación φ , entonces ¹:

$$v\left(\mathbf{u}_j = \begin{bmatrix} x'_j & y'_j & \alpha'_j \end{bmatrix}^\top\right) = \mathbf{u}'_j$$

donde:

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

Consideremos una función indicadora $\mu(\cdot, \cdot)$, que regrese 1 cuando la minucia \mathbf{u}_j y la minucia \mathbf{t}_i (asumiendo que que \mathbf{t}_i está alineado con \mathbf{u}_j) sean minucias correspondientes de acuerdo a los criterios descritos en (2.1.2a) y (2.1.2b):

$$\mu(\mathbf{u}_j, \mathbf{t}_i) = \begin{cases} 1 & \|\mathbf{u}_j - \mathbf{t}_i\|_S \leq \rho_0 \text{ y } \|\mathbf{u}_j - \mathbf{t}_i\|_D \leq \alpha_0 \\ 0 & \text{en otro caso} \end{cases}$$

Para esta situación el problema de alineación de huellas dactilares puede modelarse como:

$$\max_{\Delta x, \Delta y, \varphi, p} \sum_{i=1}^M \mu(v_{\Delta x, \Delta y, \varphi}(\mathbf{u}_{p(i)}), \mathbf{t}_i) \quad (2.1.3)$$

donde $p: [M] \mapsto [N]$ es una función desconocida que determina los pares entre minucias de \mathcal{T} y \mathcal{U} ; en particular, cada minucia tiene exactamente una minucia correspondiente en la otra huella dactilar o no tiene ninguna, es decir:

¹ Los datos de las minucias deben estar centrados antes del proceso de alineación.

1. $p(i) = j$, indica que la minucia $t_i \in \mathcal{T}$ corresponde con $u_j \in \mathcal{U}$.
2. $p(i) = \text{null}$, indica que la minucia $t_i \in \mathcal{T}$ no tiene correspondencia en \mathcal{U} . Las minucias sin correspondencia no son consideradas para (2.1.3).
3. Una minucia $u_j \in \mathcal{U}$, no tiene correspondencia en \mathcal{T} si $p(i) \neq j \forall i \in [M]$.
4. $\forall i \neq k \in [M] \Rightarrow p(i) \neq p(k) \vee p(i) = p(k) = \text{null}$, es decir, cada minucia en \mathcal{U} está asociada con a lo más una minucia en \mathcal{T} , esto es p es inyectiva.

Por otro lado, el problema de correspondencia de las minucias puede ser resuelto mediante varios enfoques que implican técnicas de reconocimiento de patrones y visión computacional, entre los cuales podemos mencionar los siguientes:

- **Geometría algebraica**. Varios métodos han sido propuestos en la literatura para diferentes versiones del problema: $M = N$ ó $N \leq M$, con correspondencia de puntos exacta o inexacta. (Bishnu *et al.*, 2006) propone un algoritmo para correspondencia parcialmente inexacta de puntos con una complejidad $\mathcal{O}(M^2 N^2 \log M)$. Sin embargo, este algoritmo se puede simplificar tomando en cuenta ciertos supuestos:
 - Todos los puntos en \mathcal{U} tienen una correspondencia en \mathcal{T} , aunque algunos puntos en \mathcal{T} pueden no tener correspondencia en \mathcal{U} .
 - Los niveles de tolerancia alrededor de los puntos no se intersectan.
- **Transformada de Hough**. Si pasamos las huellas al espacio generalizado de Hough, el problema de correspondencia de puntos se convierte en un problema de detección de picos, esto es, mediante un algoritmo basado en votación (Ballard, 1981). Además, existen enfoques basados en una transformada de Hough jerárquica que reducen la dimensión del arreglo de memoria utilizado en estos algoritmos.
- **Relajación**. El enfoque de relajación de (Rosenfeld y Kak, 1976) ajusta iterativamente, bajo cierto nivel de confianza, cada par de puntos correspondientes de forma consistente con otros pares hasta llegar a cierto criterio de convergencia. En cada iteración r , el método calcula $M \times N$ valores $p_{ij} \in [0, 1]$ que representan la probabilidad de que el punto $i \in [M]$ corresponda con punto $j \in [N]$:

$$p_{ij(r+1)} = \frac{1}{M} \sum_{h=1}^M \left[\max_{k \in [N]} \{c(i, j; h, k) \cdot p_{ij(r)}\} \right] \quad (2.1.4)$$

donde $c(i, j; h, k)$ es una métrica de compatibilidad entre los pares (i, j) y (h, k) , la cual puede ser definida de acuerdo a la consistencia del alineamiento necesario para mapear el punto j en el i y el punto k en el h . Un criterio de convergencia para determinar la correspondencia de cada punto i con el punto j puede ser $p_{ij} = \max_s \{p_{is}\}$ donde s es cualquier punto del otro conjunto.



Figura 2.2.1: Localización del punto de referencia de la imagen de una huella dactilar para determinar su punto de referencia.

- **Minimización de energía.** En este tipo de métodos se define una función que representa una “energía” (o ajuste) que da la solución al problema, de tal manera que ésta debe ser minimizada (o maximizada si se trata de un ajuste). En el uso de estas técnicas predominan los algoritmos genéticos basados en enfoques estocásticos como los propuestos por [Ansari *et al.* \(1992\)](#) o aplicaciones de métodos de recocido simulado propuestas por [Starink y Backer \(1995\)](#).

2.2 BANCO DE FILTROS DE GABOR

La sección 2.1 describe un método elemental basado en minucias, es decir, en puntos de interés extraídos de las imágenes de las huellas. A estos puntos también podemos llamarles características espaciales y angulares. A continuación describiremos otro método que se basa en otro tipo de características que no son propiamente espaciales, sino mediante la transformada de Fourier y bancos de filtros de Gabor, es decir, las características extraídas en este caso estarán en el dominio de la frecuencia. Para esto trabajaremos con las imágenes de las huellas dactilares directamente, es decir, no con datos preprocesados como en el caso de las minucias.

Un filtro de Gabor, es un filtro “pasa banda” complejo, esto es, un filtro que sólo permite pasar un rango de frecuencias de una señal en el espacio de Fourier. Entonces, la idea general es ver que datos pasan por varios filtros de Gabor entonados a distintas frecuencias y estos datos serán nuestras características.

El método que describirémos a continuación fue propuesto por [Jain et al. \(2000\)](#). Básicamente consiste en cuatro pasos descritos en el Algoritmo 1:

Algoritmo 1: Banco de Filtros de Gabor para Extracción de Características en Huellas Dactilares

1. Determinar un punto de referencia y una región de interés para la imagen de la huella dactilar.
2. Dividir de forma radial la región de interés alrededor del punto de referencia (ver Figura 2.2.1a).
3. Filtrar la región de interés en ocho direcciones diferentes usando un banco de filtros de Gabor (ocho direcciones son necesarias para obtener completamente las características locales de las crestas en una huella dactilar, mientras que sólo cuatro son suficientes para obtener la configuración global ([Jain et al., 1999](#))).
4. Para cada una de las ocho imágenes filtradas, calcular la desviación absoluta promedio (DAP) de los valores en escala de grises en cada sector para definir un vector de características que llamaremos FingerCode.

Sea $I(\mathbf{x})$, el valor en nivel de gris en el pixel $\mathbf{x} = [x_1 \ x_2]^T$ en una imagen I de una huella dactilar de dimensiones $m \times n$ y $\mathbf{c} = [c_1 \ c_2]^T$ su punto referencia. Llamamos **región de interés** a la colección de todos los sectores S_i , donde el i -ésimo sector S_i es calculado en términos de los parámetros (r, θ) (correspondiente a las coordenadas polares de \mathbf{x} con respecto a \mathbf{c}) de la siguiente forma:

$$S_i = \{ \mathbf{x} : b \lfloor i/k \rfloor \leq r < b (\lfloor (i+1)/k \rfloor + 1), \\ \theta_i \text{ mód } k+1 \leq \theta < \theta_{(i+1)} \text{ mód } k+1 \\ 1 \leq x_1 \leq n, 1 \leq x_2 \leq m \} \quad (2.2.1)$$

donde:

$$\theta_i = \frac{2\pi}{k} [(i-1) \text{ mód } k] \\ r = \|\mathbf{x} - \mathbf{c}\|_2 \\ \theta = \tan^{-1} \left(\frac{x_2 - c_2}{x_1 - c_1} \right)$$

b es el ancho de cada banda.

k es el número de sectores considerados en cada banda.

$i \in [Bk]$ con B como el número de bandas concéntricas alrededor de $\mathbf{c} = [c_1 \ c_2]^T$.

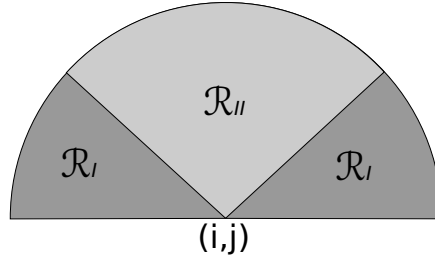


Figura 2.2.2: Regiones de integración de las intensidades de \mathcal{E} para calcular $\mathcal{A}(i, j)$.

2.2.1 Localización del punto de referencia

En (Jain *et al.*, 2000) se define un campo de orientación \mathcal{O} para la imagen de una huella dactilar como una imagen de dimensiones $p \times q$, donde $\mathcal{O}(i, j)$ representa la orientación local de las crestas en el pixel (i, j) .

Es posible estimar la orientación cuadrática media por medio de los pasos descritos en el Algoritmo 2:

Algoritmo 2: Estimación de la Orientación Cuadrática Media

1. Dividir la imagen de entrada I en bloques no superpuestos de tamaño $w \times w$.
2. Calcular el gradiente ∇I en cada pixel (i, j) .
3. Estimar la orientación local en cada bloque centrado en (i, j) por medio de:

$$\mathcal{V}_x(i, j) = 2 \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} [\nabla_x I(u, v) \nabla_y I(u, v)] \quad (2.2.2a)$$

$$\mathcal{V}_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} [\nabla_x I(u, v)^2 - \nabla_y I(u, v)^2] \quad (2.2.2b)$$

$$\mathcal{O}(i, j) = \frac{1}{2} \tan^{-1} \left[\frac{\mathcal{V}_y(i, j)}{\mathcal{V}_x(i, j)} \right] \quad (2.2.2c)$$

donde $\mathcal{O}(i, j)$ es la orientación cuadrática media local. Matemáticamente, esto representa la dirección ortogonal dominante en el **Espectro de Fourier** de una ventana de dimensiones $w \times w$.

A continuación se describe el algoritmo para determinar el punto de referencia de la imagen de una huella dactilar.

Algoritmo 3: Estimación del punto de referencia

1. Estimar el campo de orientación \mathcal{O} mediante el Algoritmo 2 con un tamaño de ventana $w \times w$.
2. Suavizar el campo de orientación en una vecindad local. Para esto, es necesario lo siguiente:

a) Un campo vectorial continuo definido como sigue:

$$\Psi_x(i, j) = \cos [2\mathcal{O}(i, j)] \quad (2.2.3a)$$

$$\Psi_y(i, j) = \sin [2\mathcal{O}(i, j)] \quad (2.2.3b)$$

donde Ψ_x y Ψ_y son las componentes en x y y del campo vectorial, respectivamente.

b) Con el campo vectorial resultante podemos aplicar un filtro pasa bajas de la siguiente manera:

$$\tilde{\Psi}_x(i, j) = \sum_{u=-\frac{w_\Psi}{2}}^{\frac{w_\Psi}{2}} \sum_{v=-\frac{w_\Psi}{2}}^{\frac{w_\Psi}{2}} W(u, v) \Psi_x(i - uw, j - vw) \quad (2.2.4a)$$

$$\tilde{\Psi}_y(i, j) = \sum_{u=-\frac{w_\Psi}{2}}^{\frac{w_\Psi}{2}} \sum_{v=-\frac{w_\Psi}{2}}^{\frac{w_\Psi}{2}} W(u, v) \Psi_y(i - uw, j - vw) \quad (2.2.4b)$$

donde W es un kernel de convolución de un filtro pasa bajas que integra la unidad y de tamaño $w_\Psi \times w_\Psi$. Un filtro de caja de 5×5 puede ser suficiente. Es importante notar que este filtro es a nivel de bloque y no de pixel.

c) El campo de orientación suavizado puede ser calculado como:

$$\tilde{\mathcal{O}}(i, j) = \frac{1}{2} \tan^{-1} \left[\frac{\tilde{\Psi}_y(i, j)}{\tilde{\Psi}_x(i, j)} \right] \quad (2.2.5)$$

3. Calcular una imagen \mathcal{E} de orientación periódica, que contenga la componente seno de $\tilde{\mathcal{O}}$ como sigue:

$$\mathcal{E}(i, j) = \sin [\tilde{\mathcal{O}}(i, j)] \quad (2.2.6)$$

4. Calcular una imagen de etiquetas \mathcal{A} que se usará para indicar el punto de referencia, de la siguiente forma (ver Figura 2.2.2):

$$\mathcal{A}(i, j) = \sum_{\mathcal{R}_I} \mathcal{E}(i, j) - \sum_{\mathcal{R}_{II}} \mathcal{E}(i, j) \quad (2.2.7)$$

5. Buscar el valor máximo en \mathcal{A} y asignar sus coordenadas al punto de referencia.
 6. Repetir los cinco pasos anteriores con distintos tamaños de ventana $\tilde{w} \times \tilde{w}$, donde $\tilde{w} < w$ y limitando la búsqueda del paso 5 a una vecindad local en cada iteración.
-

2.2.2 Proceso de filtrado

Un banco de filtros de Gabor correctamente entonado puede proveer información sobre la estructura básica de crestas y valles además de eliminar el ruido, en particular sobre la orientación (Daugmann, 1993, 1985). El primer paso es normalizar la región de interés en cada sector. Esta normalización ayuda a remover los efectos de ruido del sensor y la deformación en niveles de gris debido a las diferencias de presión en las huellas. Sea $\mathbf{I}(\mathbf{x})$, el valor en nivel de gris en el pixel $\mathbf{x} = [x_1 \ x_2]^\top$ en una imagen de una huella dactilar de dimensiones $m \times n$, μ_i y σ_i^2 la media y varianza en el sector S_i , respectivamente, y $\mathbf{N}_i(\mathbf{x})$ la intensidad en gris normalizada en el pixel $\mathbf{x} = [x_1 \ x_2]^\top$ con respecto a ese sector. Para todo los pixeles en el sector S_i , la imagen normalizada está dada por:

$$\mathbf{N}_i(\mathbf{x}) = \begin{cases} \mu_0 + \frac{\sigma_0[\mathbf{I}(\mathbf{x}) - \mu_i]}{\sigma_i} & \text{si } \mathbf{I}(\mathbf{x}) > \mu_i \\ \mu_0 - \frac{\sigma_0[\mathbf{I}(\mathbf{x}) - \mu_i]}{\sigma_i} & \text{en otro caso} \end{cases} \quad (2.2.8)$$

donde μ_0 y σ_0^2 son los valores de la media y la varianza deseadas, respectivamente. Debido a que la normalización es una operación a nivel de pixel, esto no cambia la claridad en la estructura de crestas y valles.

Un filtro de Gabor simétrico en el dominio espacial está dado por la siguiente forma:

$$\mathbf{G}_{\omega, \varphi}(\mathbf{x}) = e^{-\frac{1}{2}\tilde{\mathbf{x}}^\top \boldsymbol{\Sigma}^{-1} \tilde{\mathbf{x}}} (\cos 2\pi\omega\tilde{x}_1 + i \sin 2\pi\omega\tilde{x}_1) \quad (2.2.9)$$

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} \sin \varphi & \cos \varphi \\ \cos \varphi & -\sin \varphi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.2.10)$$

donde ω es la frecuencia es de una onda sinusoidal en la dirección φ desde el eje x_1 , y $\boldsymbol{\Sigma} = \text{diag}(\sigma_{\tilde{x}_1}^2, \sigma_{\tilde{x}_2}^2)$ son los parámetros de la envolvente Gaussiana a lo largo de los ejes \tilde{x}_1 y \tilde{x}_2 .

Para realizar el proceso de filtrado en el dominio espacial, basta con una máscara de 33×33 . Sin embargo, para acelerar el proceso, se realizará la convolución en un pixel solamente si los valores en la máscara son mayores que 0.05. Esto acelera el proceso de convolución significativamente, pero manteniendo la estructura de la información ya que se eliminarán sólo los valores que no contribuyen de manera significativa en la convolución. Para el caso de la frecuencia, se debe considerar que si el valor de ω es muy grande se crearán crestas falsas en la imagen filtrada. En cambio, si el valor de ω es muy pequeño las crestas que se encuentren muy cercanas se mezclarán en una sola. Un valor recomendado para ω es tomar el inverso de la frecuencia promedio en el relieve, es decir, $\omega = \frac{1}{K}$, donde K es la frecuencia promedio entre las crestas (de forma experimental, $K = 10$ pixeles en imágenes de huellas dactilares a 500 ppp (pixeles por pulgada).

Se procede a filtrar en ocho diferentes direcciones, es decir, ocho diferentes valores para φ ($0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}$ y $\frac{7\pi}{8}$) con respecto al eje x_1 (los kernels de convolución de

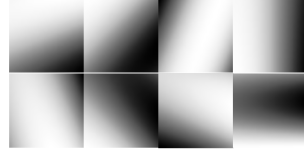


Figura 2.2.3: Kernels para los filtros de Gabor entonados a una frecuencia $\omega = \frac{1}{10}$ en 8 diferentes orientaciones.

los filtros se muestran en la Figura 2.2.3). La imagen de la región de interés normalizada convolucionada con los ocho filtros produce un conjunto de ocho imágenes filtradas. El efecto producido por el filtro radica en que las crestas paralelas a la dirección φ son acentuadas mientras que las otras son suavizadas.

2.2.3 Comparación

Una vez realizado el proceso de filtrado en las ocho direcciones antes mencionadas, procedemos a calcular el vector de características de cada huella, al cual lo denominamos FingerCode (Jain *et al.*, 1999).

Dado $F_{i\varphi}(\mathbf{x})$, el valor de la imagen filtrada en el pixel $\mathbf{x} = [x_1 \ x_2]^T$ para la dirección φ y el sector S_i , $\forall i \in [80]$ y $\varphi \in \{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}\}$, definimos la desviación absoluta promedio $v_{i\varphi}$ de la siguiente manera:

$$v_{i\varphi} = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} |F_{i\varphi}(\mathbf{x}) - \rho_{i\varphi}| \quad (2.2.11)$$

donde $|S_i|$ es la cantidad de pixeles en el sector S_i y $\rho_{i\varphi}$ es el promedio de los valores $F_{i\varphi}(\mathbf{x})$ en el sector S_i . Esto significa que al tener 80 sectores y 8 direcciones obtendremos un FingerCode de 640 características, es decir, un vector $\mathbf{v} \in \mathbb{R}^{640}$.

Con esto, el proceso de comparación en principio es bastante simple, ya que sólo consiste en calcular la norma Euclidiana de la diferencia de los FingerCode. Sin embargo, esto no hace que el método sea invariante bajo rotación. Una buena aproximación a la invarianza bajo rotación se puede lograr aprovechando las direcciones de filtrado y tomando las características del FingerCode de manera cíclica como se describe en (2.2.12a), (2.2.12b) y (2.2.12c), es decir, tomando las 80 características de cada dirección y trasladándolas a la siguiente dirección. A esto le llamaremos paso de rotación, y de esta forma una rotación de $r \in \mathbb{Z}$ pasos corresponderá a una rotación de $\frac{r\pi}{4}$ radianes en la imagen. De esta forma, un número de pasos de rotación implicará una rotación en la imagen en el sentido de las manecillas del reloj, mientras que un valor negativo una rotación en sentido contrario al

de las manecillas del reloj. El FingerCode que se obtiene después de r pasos de rotación está dado por:

$$\mathbf{v}_{i\varphi r} = \mathbf{v}_{\tilde{i}\tilde{\varphi}} \quad (2.2.12a)$$

$$\tilde{i} = (i + k + r) \text{ mód } k + k(i \text{ div } k) \quad (2.2.12b)$$

$$\tilde{\varphi} = \left(\varphi + \frac{r\pi}{4} + \pi \right) \text{ mód } \pi \quad (2.2.12c)$$

recordando que $k = 16$ es el número de sectores angulares en una banda. Dado esto, tomaremos 8 diferentes FingerCode, uno para cada dirección, es decir, $r = 0, 1, 2, \dots, 7$, por lo que obtendremos 8 normas Euclidianas de las cuales tomaremos la menor y ese será el puntaje o distancia entre las dos huellas dactilares. Finalmente, dado un umbral $\tau \in \mathbb{R}$, decidiremos si ambas huellas son lo suficientemente similares y por ende pertenecen o no al mismo sujeto. El resumen de este procedimiento se muestra en el Algoritmo 4.

Algoritmo 4: Comparación por medio de Banco de Filtros de Gabor

1. Dividir las imágenes $I_{\mathcal{J}}$ e $I_{\mathcal{U}}$ en sectores de acuerdo a lo descrito en (2.2.1).
2. Encontrar el punto de referencia en ambas huellas utilizando el Algoritmo 3.
3. Normalizar las imágenes por medio de (2.2.8).
4. Filtrar las imágenes normalizadas en cada sector y en cada dirección, por medio de la envolvente Gaussiana descrita en (2.2.9).
5. Calcular los FingerCode de ambas huellas $\mathbf{v}_{\mathcal{J}}$ y $\mathbf{v}_{\mathcal{U}}$.
6. Realizar pasos de rotación sobre $\mathbf{v}_{\mathcal{U}}$, para obtener $\mathbf{v}_{\mathcal{U}0}, \dots, \mathbf{v}_{\mathcal{U}7}$.
7. Calcular el puntaje de la comparación por medio de:

$$\tilde{s}(I_{\mathcal{J}}, I_{\mathcal{U}}) = \min_{r \in \{0, \dots, 7\}} \|\mathbf{v}_{\mathcal{J}} - \mathbf{v}_{\mathcal{U}r}\|_2 \quad (2.2.13)$$

8. Si $\tilde{s}(I_{\mathcal{J}}, I_{\mathcal{U}}) \leq \tau$ entonces:

las huellas son similares

de lo contrario:

las huellas NO son similares

MÉTRICAS PARA LA COMPARACIÓN DE HUELLAS DACTILARES

Como se mencionó en los capítulos anteriores, durante el desarrollo de este trabajo se utilizaron tres enfoques diferentes para resolver el problema de comparación de huellas dactilares. En este capítulo detallaremos los métodos que se utilizaron, los cuales están basados en el estado del arte que describimos en el capítulo anterior.

3.1 MÉTODOS BÁSADOS EN GRAFOS

En estos enfoques, de manera general utilizaremos el esquema definido a continuación para realizar la comparación de dos huellas dactilares:

1. Definir una norma $\|\cdot\|_z : \mathbb{R}^3, \mathbb{R}^3 \rightarrow \mathbb{R}$ que nos indique la relación que existe entre dos minucias.
2. Tomar pares de minucias en ambas huellas dactilares y calcular la norma definida.
3. Construir un grafo basado en las normas obtenidas.
4. Definir una medida de similitud que tome el grafo como entrada y un umbral $\tau \in \mathbb{R}$.
5. Calcular y comparar la medida de similitud obtenida con el umbral.

Por lo que no utilizaremos las imágenes de las huellas en este momento sino los puntos de interés, los cuales asumiremos que han sido previamente extraídos. Esto es, no se tratará aquí el proceso de extracción de estos puntos de interés y como entrada tendremos una matriz de datos en la que cada renglón contendrá la información espacial, angular y confiabilidad de las minucias.

En este caso, nuestros algoritmos recibirán como entrada dos conjuntos de observaciones \mathcal{T} y \mathcal{U} como los descritos en (2.1.1a) y (2.1.1b), con los que formaremos las matrices de datos $\tilde{\mathbf{T}} \in \mathbb{R}^{n \times 3}$ y $\tilde{\mathbf{U}} \in \mathbb{R}^{m \times 3}$, en las cuales cada renglón contendrá la información de una minucia con las componentes descritas anteriormente. De manera que el objetivo es definir una función $\|\cdot\|_g : \mathbb{R}^{m \times 3}, \mathbb{R}^{n \times 3} \rightarrow \mathbb{R}$, es decir, una función que nos de como salida un valor real que nos represente la similitud entre dos huellas dactilares tomando sus minucias como entrada.

Dado que las minucias de las huellas a tratar pueden venir de distintas fuentes y condiciones (como el sensor o la resolución de las imágenes originales), es recomendable primeramente realizar una selección y normalización. El proceso de selección consistirá en tomar las n minucias más confiables de un conjunto de N en total, es decir, las

de mayor calidad. Esta selección debe hacerse con cuidado, ya que es importante tener un número considerable de minucias por cuestiones estadísticas. Es común que los algoritmos de extracción de puntos de interés den como resultado 70 y 80 minucias para sensores ópticos de buena calidad y en nuestros experimentos consideramos que de las N minucias, podremos contar con al menos n minucias de buena calidad, es decir, de cada conjunto de minucias tomamos las n mejores, por ejemplo $n = 50$. Por otro lado, el proceso de normalización también podrá contribuir a que el problema se mantenga hasta cierto punto acotado. De acuerdo a la descripción de los componentes de una minucia al inicio de la sección, podemos observar que el componente espacial es el único del que no tenemos cota superior, es decir, sólo sabemos que las coordenadas de la posición de la minucia son enteros positivos. Por esta razón la normalización radicaré en este componente. Si llamamos \mathbf{x} y \mathbf{y} a las dos primeras columnas de la matriz de datos que contienen al componente espacial de cada minucia respectivamente y \mathbf{p} a la componente espacial completa, entonces podemos ver que una normalización adecuada es la siguiente:

$$\tilde{\mathbf{p}}_i = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{x_i - \bar{x}}{\|\mathbf{x} - \bar{x}\mathbf{1}\|_\infty} & \frac{y_i - \bar{y}}{\|\mathbf{y} - \bar{y}\mathbf{1}\|_\infty} \end{bmatrix} \in \mathbb{D}^2 \quad (3.1.1)$$

Es decir, centramos los datos de posición y dividimos sus componentes entre sus respectivos máximos y además multiplicamos por un factor que nos garantice mantener nuestros datos en el disco unitario de dimensión 2. Esto último nos ayudará a que la distancia euclidiana entre las componentes espaciales de cualquier par de minucias sea a lo más 2 y esto nos será útil un poco más adelante.

Por otro lado, para la componente angular bastará con tenerla en radianes, es decir, en el intervalo $[0, 2\pi)$, esto simplemente para facilitar los cálculos por medio de la computadora.

Con esto, manejamos dos conjuntos \mathcal{T} y \mathcal{U} que contienen las minucias de dos huellas dactilares, representadas mediante tercias de la forma $[x \ y \ \alpha]$, en las que x y y denotan la localización de la minucia en la imagen de la huella dactilar y α su orientación, de esta manera:

$$\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_M\} \quad \mathbf{t}_i = \begin{bmatrix} x_i & y_i & \alpha_i \end{bmatrix}^\top \quad i \in [M] \quad (3.1.2a)$$

$$\mathcal{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_N\} \quad \mathbf{u}_j = \begin{bmatrix} x'_j & y'_j & \alpha'_j \end{bmatrix}^\top \quad j \in [N] \quad (3.1.2b)$$

donde M y N representan la cantidad de minucias en \mathcal{T} y \mathcal{U} , respectivamente. Nuestros conjuntos de entrada serán:

$$\tilde{\mathcal{T}} = \{\tilde{\mathbf{t}}_1, \dots, \tilde{\mathbf{t}}_m\} \quad \tilde{\mathbf{t}}_i = \begin{bmatrix} \tilde{x}_i & \tilde{y}_i & \tilde{\alpha}_i \end{bmatrix}^\top \quad i \in [m] \quad (3.1.3a)$$

$$\tilde{\mathcal{U}} = \{\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_n\} \quad \tilde{\mathbf{u}}_j = \begin{bmatrix} \tilde{x}'_j & \tilde{y}'_j & \tilde{\alpha}'_j \end{bmatrix}^\top \quad j \in [n] \quad (3.1.3b)$$

donde m y n representan la cantidad de minucias que fueron seleccionadas de los conjuntos originales \mathcal{F} y \mathcal{U} , respectivamente.

Algoritmo 5: Normalizar y centrar minucias

Datos : \mathcal{F} : conjunto de minucias; n : cantidad de minucias
Resultados : $\tilde{\mathcal{F}} \subseteq \mathbb{D}^2 \times [-\pi, \pi)$: minucias normalizadas y centradas

```

1 función normalizar-y-centrar( $\mathcal{F} \subseteq \mathbb{Z}^{+2} \times [0, 360)$ ;  $n \in \mathbb{N}$ ) inicio
2    $x_{\text{máx}}, y_{\text{máx}}, \alpha_{\text{máx}} \leftarrow \text{máx}(\mathcal{F});$  //  $\alpha_{\text{máx}}$  no es realmente necesaria.
3   para cada  $m_i \in \mathcal{F}$  hacer
4      $\hat{m}_i \leftarrow \left[ \begin{array}{ccc} x_i & y_i & 2\pi \frac{\alpha_i}{180} \end{array} \right];$ 
5   fin
6    $\hat{m} \leftarrow \text{media}(\hat{\mathcal{F}});$ 
7   para cada  $\hat{m}_i \in \hat{\mathcal{F}}$  hacer
8      $\tilde{m}_i \leftarrow \hat{m}_i - \hat{m};$ 
9   fin
10  regresar  $\tilde{\mathcal{F}}$ 
11 fin

```

Hasta aquí, ambos conjuntos los tenemos en la misma escala y con la misma media espacial, por lo que podemos proceder al primer paso de nuestro esquema general, es decir, definir una norma que nos sirva como indicador para comparar la minucia i con la minucia j . Para esto podemos proponer varias ideas:

1. Una idea inicial puede ser tomar una **componente de proyección** de una minucia sobre la otra, de forma que, tomamos la dirección de la diferencia en la parte espacial por medio de:

$$\vec{s}_{ij} = \frac{(\tilde{p}_i - \tilde{p}_j)}{\|\tilde{p}_i - \tilde{p}_j\|_2} \quad (3.1.4)$$

Posteriormente, definimos las direcciones de la parte angular como:

$$\vec{\alpha}_i = [\cos \tilde{\alpha}_i \quad \sin \tilde{\alpha}_i]^\top \quad (3.1.5a)$$

$$\vec{\alpha}_j = [\cos \tilde{\alpha}_j \quad \sin \tilde{\alpha}_j]^\top \quad (3.1.5b)$$

Luego, calculamos la dirección de la diferencia entre ambas con:

$$\vec{\varphi}_{ij} = \frac{(\vec{\alpha}_i - \vec{\alpha}_j)}{\|\vec{\alpha}_i - \vec{\alpha}_j\|_2} \quad (3.1.6)$$

Finalmente, nuestra medida de relación entre ambas minucias es la siguiente:

$$d_{ij} = \frac{1}{2} \left(1 - \vec{s}_{ij}^\top \vec{\varphi}_{ij} \right) \in [0, 1] \quad (3.1.7)$$

2. Otra idea es tomar la distancia euclidiana en la parte espacial:

$$s_{ij} = \frac{1}{2} \|\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_j\|_2 \quad (3.1.8)$$

posteriormente definimos $\beta_{ij} \in [0, \pi)$ como el ángulo de inclinación de la recta que une a la minucia i con la minucia j , es decir:

$$\tan \beta_{ij} = \frac{y_i - y_j}{x_i - x_j} \quad (3.1.9)$$

seguido de esto, restamos β_{ij} a cada una de las orientaciones y calculamos la distancia entre los cosenos de ambas diferencias, es decir:

$$\varphi_{ij} = \frac{1}{2} |\cos(\tilde{\alpha}_i - \beta_{ij}) - \cos(\tilde{\alpha}_j - \beta_{ij})| \quad (3.1.10)$$

finalmente nuestra **medida de relación** entre ambas minucias puede ser:

$$d_{ij}^2 = s_{ij}^2 + \varphi_{ij}^2 \in [0, 1] \quad (3.1.11)$$

3. Una idea más simple, pero funcional, es tomar la distancia euclidiana de la parte espacial y la distancia coseno en la parte angular, es decir:

$$\begin{aligned} s_{ij} &= \frac{1}{2} \|\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_j\|_2 \\ \varphi_{ij} &= \frac{1}{2} [1 - \cos(\tilde{\alpha}_i - \tilde{\alpha}_j)] \end{aligned}$$

y la medida de relación queda dada por la suma acotada:

$$d_{ij} = s_{ij} + \varphi_{ij} \in [0, 1] \quad (3.1.13)$$

4. Una mejora las opciones 2 y 3, consiste en ponderar la parte espacial y la parte angular por medio de un escalar $\mu \in [0, 1]$ que de más importancia a una componente u otra, de forma que podemos escribir (3.1.11) y (3.1.13) como (3.1.14) y (3.1.15) respectivamente.:

$$\tilde{d}_{ij}^2 = \mu s_{ij}^2 + (1 - \mu) \varphi_{ij}^2 \in [0, 1] \quad (3.1.14)$$

$$\tilde{d}_{ij} = \mu s_{ij} + (1 - \mu) \varphi_{ij} \in [0, 1] \quad (3.1.15)$$

5. Finalmente, podemos tomar la **distancia de Mahalanobis** normalizada. Esto es, si tenemos un conjunto de minucias \mathcal{F} el cual convertiremos a una matriz de datos $\mathbf{F} \in \mathbb{R}^{n \times 3}$, la distancia de Mahalanobis entre dos minucias \mathbf{m}_i y \mathbf{m}_j en el conjunto está dada por:

$$\|\mathbf{m}_i - \mathbf{m}_j\|_{\text{Mahalanobis}}^2 = (\mathbf{m}_i - \mathbf{m}_j)^\top [\text{cov}(\mathbf{F})]^{-1} (\mathbf{m}_i - \mathbf{m}_j) \quad (3.1.16)$$

Ahora bien, suponiendo que el conjunto de minucias está normalizado y centrado, el máximo de la distancia de Mahalanobis será el eigenvalor más grande de la inversa de la matriz de covarianza de los datos, al que llamaremos λ_1 ; por lo que para asegurar que la distancia esté en el intervalo $[0, 1]$, bastará con dividir entre λ_1 como se muestra a continuación:

$$d_{ij}^2 = \frac{1}{\lambda_1} (\mathbf{m}_i - \mathbf{m}_j)^\top [\text{cov}(\mathbf{F})]^{-1} (\mathbf{m}_i - \mathbf{m}_j) \in [0, 1] \quad (3.1.17)$$

Una vez elejida una de las métricas de distancia definidas anteriormente podemos pasar al tercer paso de nuestro esquema general. Sin embargo, a partir de aquí proponemos dos enfoques distintos, el primero de ellos construye un grafo con cada conjunto de entrada, mientras que el segundo construye un solo grafo combinando los dos conjuntos de entrada.

3.1.1 Método de la matriz de sociabilidad

Dado un grafo no dirigido G de n vértices y con matriz de adyacencia $\mathbf{A} \in \{0, 1\}^{n \times n}$, definimos su matriz de sociabilidad de grado r , como una matriz $\mathbf{S}^r \in \mathbb{R}^{n \times r}$ que contiene en la entrada s_{ij} el promedio normalizado del número de caminos de tamaño j que salen del nodo v_i , es decir, la matriz de sociabilidad está dada por:

$$\mathbf{S}^r = \left[\tilde{\mathbf{A}}\mathbf{1} \mid \tilde{\mathbf{A}}^2\mathbf{1} \mid \dots \mid \tilde{\mathbf{A}}^r\mathbf{1} \right] \quad (3.1.18)$$

donde $\tilde{\mathbf{A}}^j = \left[\text{diag}(\mathbf{1}^\top \mathbf{A}^j) \right]^{-1} \mathbf{A}^j$ con $j \in [r]$ y r representa el tamaño máximo de camino que deseamos considerar.

Algoritmo 6: Cálculo de la Matriz de Sociabilidad

Datos : \mathbf{A} : matriz de adyacencia; n : cantidad de nodos del grafo; r : grado de sociabilidad

Resultados : $\mathbf{S}^r \in \mathbb{R}^{n \times r}$: matriz de sociabilidad

```

1 función matriz-de-sociabilidad( $\mathbf{A} \in \{0, 1\}^{n \times n}$ ;  $n, r \in \mathbb{N}$ ) inicio
2    $\mathbf{S}^r \leftarrow \mathbf{0}_{n \times r}$ ;
3   para  $j \leftarrow 2$  a  $r$  hacer
4      $[\mathbf{S}^r]_{:,j} \leftarrow \left[ \text{diag}(\mathbf{1}^\top \mathbf{A}^j) \right]^{-1} \mathbf{A}^j \mathbf{1}$ ;
5   fin
6   regresar  $\mathbf{S}^r$ ;
7 fin

```

Ahora bien, eligiendo una de las relaciones $\|\cdot\|_{Z_{\#}}$ entre minucias mencionadas anteriormente podemos formar una matriz de distancias entre todos los pares de minucias de los conjuntos $\tilde{\mathcal{F}}$ y $\tilde{\mathcal{U}}$. Es decir, las formamos las matrices de distancias siguientes:

$$[\mathbf{D}_{\tilde{\mathcal{F}}}]_{ij} = \|\tilde{\mathbf{t}}_i - \tilde{\mathbf{t}}_j\|_{Z_{\#}} \quad i, j \in [m] \quad (3.1.19)$$

$$[\mathbf{D}_{\tilde{\mathcal{U}}}]_{ij} = \|\tilde{\mathbf{u}}_i - \tilde{\mathbf{u}}_j\|_{Z_{\#}} \quad i, j \in [n] \quad (3.1.20)$$

Posteriormente, binarizamos estas matrices mediante un umbral $\delta \in [0, 1]$, de forma que las minucias representarán los vértices de un grafo y decimos que dos minucias conectadas por una arista si la norma $Z_{\#}$ de su diferencia es menor a δ , es decir, las matrices de adyacencia del grafo estarán dadas por:

$$[\mathbf{A}_{\tilde{\mathcal{F}}}]_{ij} = \mathbb{1}_{[0, \delta]}([\mathbf{D}_{\tilde{\mathcal{F}}}]_{ij}) \quad i \neq j \in [m] \quad (3.1.21)$$

$$[\mathbf{A}_{\tilde{\mathcal{U}}}]_{ij} = \mathbb{1}_{[0, \delta]}([\mathbf{D}_{\tilde{\mathcal{U}}}]_{ij}) \quad i \neq j \in [n] \quad (3.1.22)$$

Algoritmo 7: Cálculo de la Matriz de Adyacencia

Datos : $\tilde{\mathcal{F}}$: conjunto de minucias; n : cantidad de minucias; δ : umbral de distancia; $\|\cdot\|_{Z_{\#}}$: función de distancia
Resultados : $\mathbf{A} \in \{0, 1\}^{n \times n}$: matriz de adyacencia
1 función matriz-de-adyacencia($\tilde{\mathcal{F}} \subseteq \mathbb{D}^2 \times [-\pi, \pi)$; $n \in \mathbb{N}$; $\delta \in [0, 1]$;
 $\|\cdot\|_{Z_{\#}} : \mathbb{R}^3, \mathbb{R}^3 \mapsto \mathbb{R}$)**inicio**
2 $\mathbf{A} \leftarrow \mathbf{0}_{n \times n}$;
3 **para cada** $\tilde{\mathbf{m}}_i \in \tilde{\mathcal{F}}$ **hacer**
4 **para cada** $\tilde{\mathbf{m}}_j \in \tilde{\mathcal{F}}$ **hacer**
5 **si** $i \neq j$ **entonces:**
6 $[\mathbf{A}]_{ij} \leftarrow \mathbb{1}_{[0, \delta]}(\|\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}}_j\|_{Z_{\#}})$;
7 **fin**
8 **fin**
9 **fin**
10 **regresar** \mathbf{A} ;
11 fin

Ahora bien, podemos tomar las correspondientes matrices de sociabilidad de grado r para cada grafo $\mathbf{S}_{\tilde{\mathcal{F}}}^r$ y $\mathbf{S}_{\tilde{\mathcal{U}}}^r$, respectivamente, como si fueran matrices de datos en las que cada renglón es una observación y con esta realizar Análisis de Componentes Principales Bishop (2006, PCA). Con esto obtendremos las direcciones de máxima variabilidad de los promedios de caminos en cada grafo y su correspondientes latencias. Si tomamos

K componentes principales podemos relacionar las dos huellas dactilares por medio del siguiente valor:

$$\|\tilde{\mathcal{J}} - \tilde{\mathcal{U}}\|_{\mathcal{R}}^2 = \frac{1}{K} \sum_{k=1}^K [(\lambda_{\tilde{\mathcal{J}}_k} - \lambda_{\tilde{\mathcal{U}}_k}) \|\psi_{\tilde{\mathcal{J}}_k} - \psi_{\tilde{\mathcal{U}}_k}\|_2^2] \quad (3.1.23)$$

donde $\lambda_{\tilde{\mathcal{J}}_k}$ y $\lambda_{\tilde{\mathcal{U}}_k}$ son los eigenvalores de las matrices $\text{cov}(\mathbf{S}_{\tilde{\mathcal{J}}}^r)$ y $\text{cov}(\mathbf{S}_{\tilde{\mathcal{U}}}^r)$, respectivamente, ordenados de mayor a menor, mientras que $\psi_{\tilde{\mathcal{J}}_k}$ y $\psi_{\tilde{\mathcal{U}}_k}$ son los eigenvectores correspondientes a dichos eigenvalores (ver Figura 3.1.1). Finalmente, podemos observar este procedimiento resumido en el Algoritmo 8.

Algoritmo 8: Diferencia en Variabilidad Cuadrática Promedio de Caminos

Datos : \mathcal{J} : conjunto de minucias de prueba; m: cantidad de minucias de prueba; \mathcal{U} : conjunto de minucias del usuario; n: cantidad de minucias del usuario; δ : umbral de distancia; $\|\cdot\|_{\mathcal{Z}_{\#}}$: función de distancia entre dos minucias; τ : umbral de distancia entre los grafos formados con las minucias; r: grado de las matrices de sociabilidad a utilizar

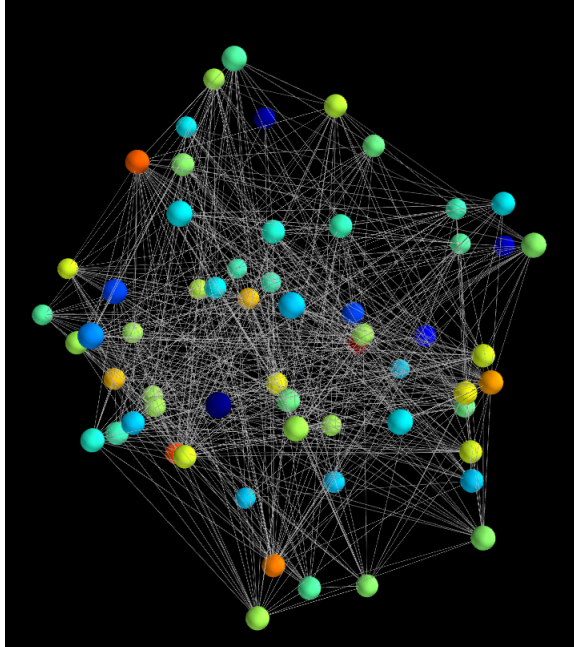
Resultados : Valor lógico que indica si las dos huellas son suficientemente similares para ser consideradas del mismo usuario.

1 **función** distancia-de-sociabilidad($\mathcal{J}, \mathcal{U} \subseteq \mathbb{D}^2 \times [-\pi, \pi)$; m, n, r $\in \mathbb{N}$; $\delta \in [0, 1]$;
 $\|\cdot\|_{\mathcal{Z}_{\#}} : \mathbb{R}^3, \mathbb{R}^3 \mapsto \mathbb{R}$; $\tau \in \mathbb{R}$) **inicio**
2 $\tilde{\mathcal{J}} \leftarrow \text{normalizar-y-centrar}(\mathcal{J}, m)$;
3 $\tilde{\mathcal{U}} \leftarrow \text{normalizar-y-centrar}(\mathcal{U}, n)$;
4 $\mathbf{A}_{\tilde{\mathcal{J}}} \leftarrow \text{matriz-de-adyacencia}(\tilde{\mathcal{J}}, m, \delta)$;
5 $\mathbf{A}_{\tilde{\mathcal{U}}} \leftarrow \text{matriz-de-adyacencia}(\tilde{\mathcal{U}}, n, \delta)$;
6 $\mathbf{S}_{\tilde{\mathcal{J}}}^r \leftarrow \text{matriz-de-sociabilidad}(\tilde{\mathcal{J}}, m, r)$;
7 $\mathbf{S}_{\tilde{\mathcal{U}}}^r \leftarrow \text{matriz-de-sociabilidad}(\tilde{\mathcal{U}}, n, r)$;
8 pca($\mathbf{S}_{\tilde{\mathcal{J}}}^r$);
9 pca($\mathbf{S}_{\tilde{\mathcal{U}}}^r$);
10 **si** $\|\tilde{\mathcal{J}} - \tilde{\mathcal{U}}\|_{\mathcal{R}}^2 \leq \tau$ **entonces**:
11 | **regresar verdadero**;
12 **de lo contrario**:
13 | **regresar falso**;
14 **fin**
15 **fin**

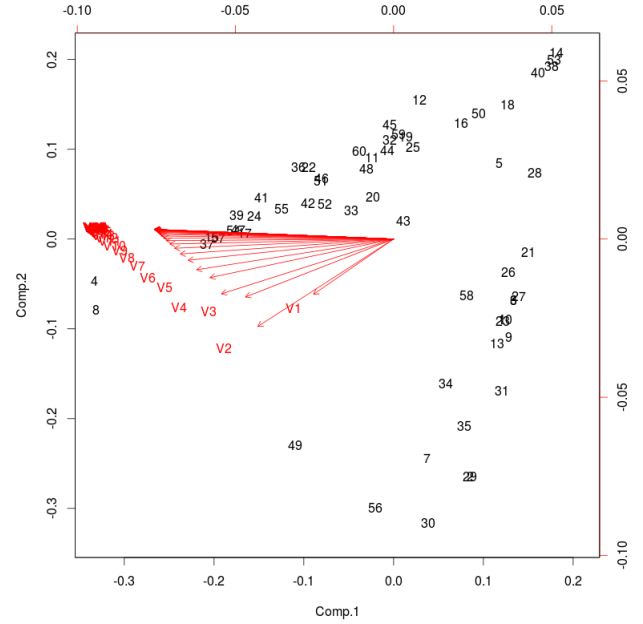
El paso 10 del Algoritmo 8 puede ser modificado para devolver un grado de similitud $\tilde{s}(\tilde{\mathcal{J}}, \tilde{\mathcal{U}})$, entre huellas usando:

$$\tilde{s}(\tilde{\mathcal{J}}, \tilde{\mathcal{U}}) = e^{-\frac{\|\tilde{\mathcal{J}} - \tilde{\mathcal{U}}\|_{\mathcal{R}}^2}{\sigma^2}} \quad (3.1.24)$$

Una ventaja de usar similitud en lugar de la distancia, es que, por construcción está perfectamente acotada en el intervalo $[0, 1]$, por lo que es un poco más fácil decidir el umbral τ a utilizar.



(a) Grafo generado



(b) Proyección sobre componentes principales

Figura 3.1.1: Un grafo generado por un conjunto de minucias con su proyección de los datos de la matriz de sociabilidad sobre los primeros dos componentes principal.

3.1.2 Método del esquema espectral

Dados los conjuntos de minucias $\tilde{\mathcal{T}}$ y $\tilde{\mathcal{U}}$, búscamos los $k \in \mathbb{N}$ vecinos más cercanos en $\tilde{\mathcal{U}}$ de cada minucia en $\tilde{\mathcal{T}}$, para esto utilizamos la norma euclídeana solo en las componentes espaciales de las minucias. Posteriormente, calculamos la matriz de homografía \mathbf{H} para registrar las minucias de $\tilde{\mathcal{U}}$ con las de $\tilde{\mathcal{T}}$ y obtener $\hat{\mathcal{U}}$, esto es una aproximación inicial de la correspondencia de las minucias entre ambos conjuntos. Seguido de esto y tomando como base el enfoque de [Luo et al. \(2003\)](#), formamos un grafo, de manera que calculamos las distancias desde todas las minucias en $\tilde{\mathcal{T}}$ hacia todas las de $\hat{\mathcal{U}}$ y construimos una matriz de distancias $\mathbf{D} \in \mathbb{R}^{m \times n}$, de manera que cada entrada estará dada por:

$$[\mathbf{D}]_{ij} = \|\tilde{\mathbf{t}}_i - \hat{\mathbf{u}}_j\|_{z_{\#}} \quad (i, j) \in [m] \times [n] \tag{3.1.25}$$

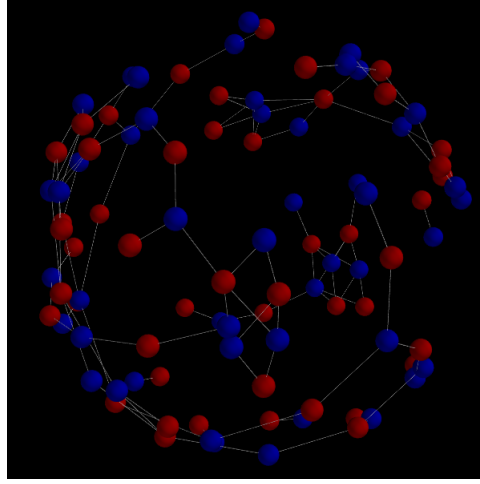


Figura 3.1.2: Grafo bipartito generado por dos huellas dactilares. Los vértices de cada conjunto de minucias se muestran en color diferente.

Posteriormente, binarizando por medio de con un umbral $\delta \in [0, 1]$, obtendremos una matriz $\mathbf{A} \in \{0, 1\}^{m \times n}$, en la que cada entrada estará dada por:

$$[\mathbf{A}]_{ij} = \mathbb{1}_{[0, \delta]}([\mathbf{D}]_{ij}) \quad (i, j) \in [m] \times [n] \quad (3.1.26)$$

En este punto, es importante resaltar que si se decide usar la distancia de Mahalanobis normalizada se debe considerar formar una sola matriz de covarianza con las minucias de ambas huellas una vez ya registradas, es decir, considerar una sola matriz de datos a partir de los conjuntos $\tilde{\mathcal{T}}$ y $\tilde{\mathcal{U}}$.

Seguido de esto, construimos una matriz de adyacencia $\tilde{\mathbf{A}} \in \{0, 1\}^{(m+n) \times (m+n)}$ para un grafo bipartito, es decir, una matriz a bloques que tendrá la siguiente forma:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{m \times m} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0}_{n \times n} \end{bmatrix} \quad (3.1.27)$$

Posteriormente, contaremos la cantidad de caminos de tamaño r en promedio que hay del conjunto $\tilde{\mathcal{T}}$ al conjunto $\tilde{\mathcal{U}}$, para algún $r \in \mathbb{N}$, por medio de:

$$s_r(\tilde{\mathcal{T}}, \tilde{\mathcal{U}}) = \frac{1}{kr(m+n)} \sum_{i=1}^{m+n} \lambda_i \quad (3.1.28)$$

donde $\lambda_1, \dots, \lambda_{m+n}$ son los eigenvalores de $\tilde{\mathbf{A}}^\top$.

Ahora bien, para que esta medida sea simétrica se realiza el mismo procedimiento para obtener $s_r(\tilde{\mathcal{U}}, \tilde{\mathcal{T}})$ y se promedian ambos resultados, es decir, nuestra medida está dada por:

$$\tilde{s}_r(\tilde{\mathcal{T}}, \tilde{\mathcal{U}}) = \frac{1}{2} [s_r(\tilde{\mathcal{T}}, \tilde{\mathcal{U}}) + s_r(\tilde{\mathcal{U}}, \tilde{\mathcal{T}})] \quad (3.1.29)$$

Finalmente, mediante un umbral $\tau \in \mathbb{R}$ decidiremos si ambas huellas son lo suficientemente similares y por ende pertenecen o no al mismo sujeto. Sin embargo, debemos tomar en cuenta que ésta es una medida de similitud y no de distancia. Este procedimiento se resume en el Algoritmo 9.

Algoritmo 9: Promedio de Número de Caminos

Datos : \mathcal{T} : conjunto de minucias de prueba; m : cantidad de minucias de prueba; \mathcal{U} : conjunto de minucias del usuario; n : cantidad de minucias del usuario; δ : umbral de distancia; $\|\cdot\|_{z_{\#}}$: función de distancia entre dos minucias; τ : umbral de similitud entre las huellas dactilares; r : longitud de caminos; k : cantidad de vecinos más cercanos a buscar

Resultados : Valor lógico que indica si las dos huellas son suficientemente similares para ser consideradas del mismo usuario.

1 **función** similitud-promedio-en-caminos($\mathcal{T}, \mathcal{U} \subseteq \mathbb{D}^2 \times [-\pi, \pi]$; $m, n, r, k \in \mathbb{N}$;
 $\delta \in [0, 1]$; $\|\cdot\|_{z_{\#}} : \mathbb{R}^3, \mathbb{R}^3 \mapsto \mathbb{R}$; $\tau \in \mathbb{R}$) **inicio**

2 $\tilde{\mathcal{T}} \leftarrow \text{normalizar-y-centrar}(\mathcal{T}, m)$;

3 $\tilde{\mathcal{U}} \leftarrow \text{normalizar-y-centrar}(\mathcal{U}, n)$;

4 $\mathcal{P} \leftarrow \text{k-vecinos-más-cercanos}(\tilde{\mathcal{T}}, \tilde{\mathcal{U}})$;

5 $\mathbf{H} \leftarrow \text{matriz-de-homografía}(\tilde{\mathcal{T}}, \tilde{\mathcal{U}})$;

6 $\tilde{\mathcal{U}} \leftarrow \text{transformación-de-perspectiva}(\tilde{\mathcal{T}}, \mathbf{H})$;

7 **para cada** $\mathbf{p} \in \mathcal{P}$ **hacer**

8 **para cada** $(i, j) \in \mathbf{p}$ **hacer**

9 $[\mathbf{A}]_{ij} \leftarrow \mathbb{1}_{[0, \delta]}(\|\tilde{\mathbf{t}}_i - \tilde{\mathbf{u}}_j\|_{z_{\#}})$;

10 $[\mathbf{A}]_{ij} \leftarrow [\mathbf{A}]_{ji}$;

11 **fin**

12 **fin**

13 $\lambda \leftarrow \text{eigenvalores}(\mathbf{A}^r)$;

14 **si** $\frac{1}{2} [s_r(\tilde{\mathcal{T}}, \tilde{\mathcal{U}}) + s_r(\tilde{\mathcal{U}}, \tilde{\mathcal{T}})] \geq \tau$ **entonces:**

15 **regresar verdadero;**

16 **de lo contrario:**

17 **regresar falso;**

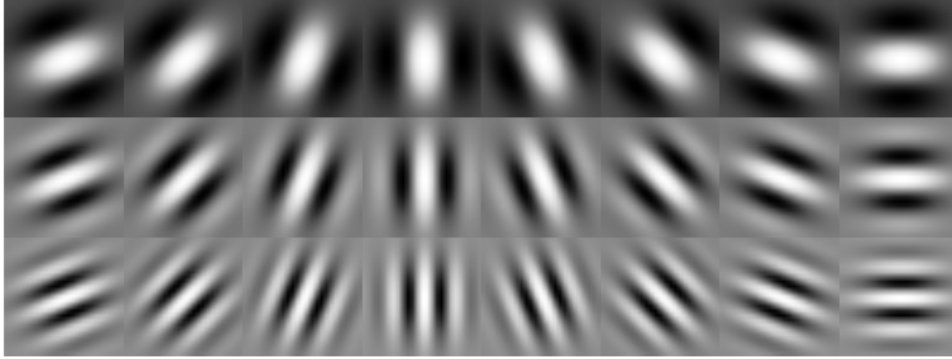
18 **fin**

19 **fin**

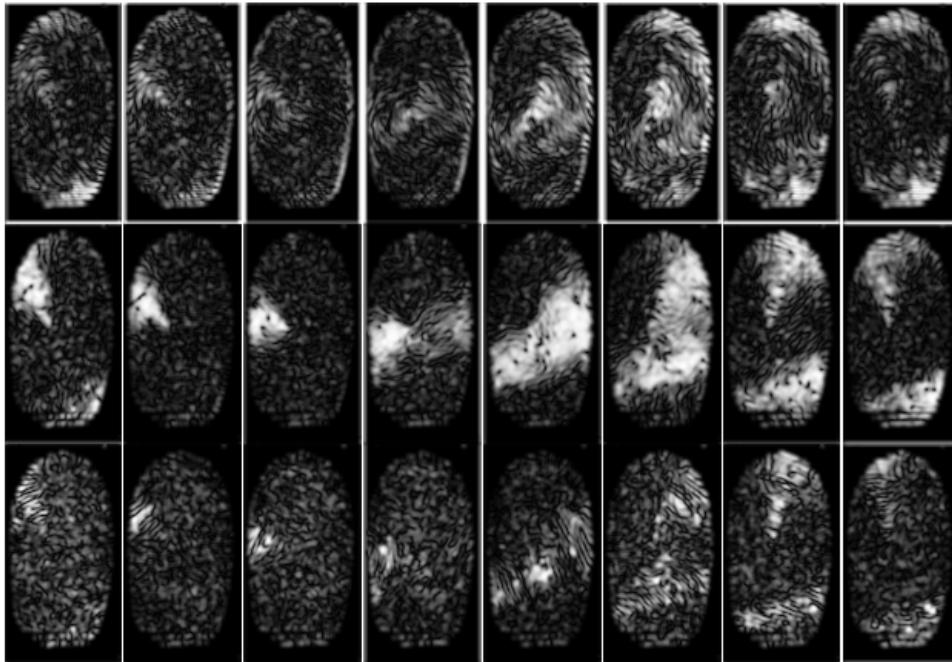
3.2 MÉTODO BASADO EN ANÁLISIS DE FOURIER

En la sección 2.2 vimos como podemos realizar la comparación de dos huellas dactilares mediante bancos de filtros de Gabor. De acuerdo al estado del arte este algoritmo muestra ser bastante robusto, sin embargo, es un poco costoso al momento de realizar el proceso

para la extracción de características, por lo que se propone una alternativa un poco más simple. Básicamente existen dos diferencias entre el proceso descrito anteriormente y el propuesto en este trabajo.



(a) Kernels para los filtros de Gabor



(b) Magnitud de las imágenes filtradas

Figura 3.2.1: Banco de filtros de Gabor en 8 orientaciones y entonado a 3 frecuencias

Primeramente, de igual forma que el algoritmo descrito anteriormente procedemos a dividir la imagen I en sectores de acuerdo a con (2.2.1) y buscamos el punto de referencia por medio del Algoritmo 3. Ahora bien, podemos entonar los kernels de los filtros a h distintas frecuencias. Estos valores de ω producirán $8h$ kernels para el banco de filtros y por

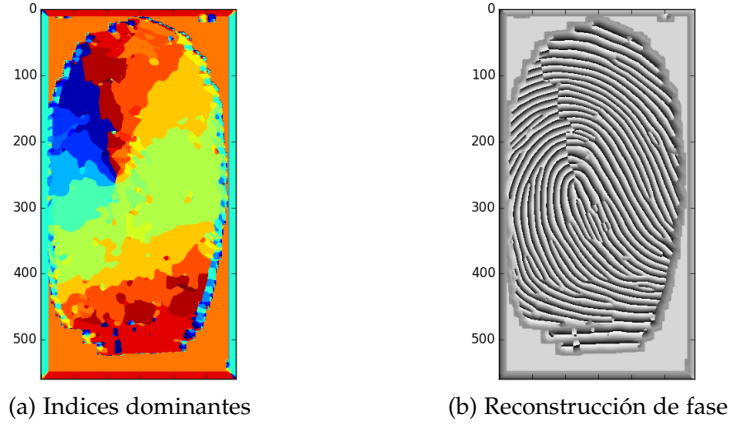


Figura 3.2.2: Extracción de características por medio de magnitud y fase

consecuente $8h$ imágenes en valores complejos en lugar de sólo las 8 que se mencionaron anteriormente. La figura 3.2.1a muestra la visualización de 24 kernels ($h = 3$) generados por frecuencias de 0.3, 0.6 y 0.9; estos valores fueron tomados de forma experimental.

Una vez realizado el proceso de filtrado con estos kernels, se procede a tomar la orientación dominante entre las 24 imágenes en cada pixel $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$, esto es, para cada pixel se toma índice de la imagen en la que la magnitud sea la máxima. Para esto podemos nombrar y enumerar las imágenes F_j con $j \in [8h]$; es decir, para $j = 1$, se toman los valores de ω_1 y φ_1 , mientras que para $j = 8h$, se toman ω_h y φ_8 . Con esto podemos crear una imagen J que contenga los índices de las orientaciones dominantes (ver Figura 3.2.2a).

$$J(\mathbf{x}) = \arg \max_j \|F_j(\mathbf{x})\| \quad (3.2.1a)$$

$$F_j(\mathbf{x}) = [\tilde{\mathbf{I}} * \mathbf{G}_{\omega, \varphi}](\mathbf{x}) \quad (3.2.1b)$$

De tal forma que nuestras características consistirán en una imagen \mathbf{C} , que contiene el valor de la fase en las imágenes correspondientes al índice de la orientación dominante (ver Figura 3.2.2b), es decir:

$$\mathbf{C}(\mathbf{x}) = \arg F_{J(\mathbf{x})}(\mathbf{x}) \quad (3.2.2)$$

Finalmente, para realizar la comparación bastará con calcular la distancia cuadrática promedio entre dos imágenes de características y calcular la similitud para compararla con un umbral $\tau \in \mathbb{R}$, esto es:

$$\tilde{s}(\tilde{\mathcal{J}}, \tilde{\mathcal{U}}) = e^{-\frac{1}{mn} \sum_x [C_{\mathcal{J}}(\mathbf{x}) - C_{\mathcal{U}}(\mathbf{x})]^2} \quad (3.2.3)$$

Este procedimiento puede ser resumido en el Algoritmo 10:

Algoritmo 10: Simitud por medio de Banco de Filtros de Gabor

Datos : $I_{\mathcal{T}}$: imagen de la huella de prueba; U : imagen de la huella del usuario; τ : umbral de similitud entre las huellas dactilares

Resultados : Valor lógico que indica si las dos huellas son suficientemente similares para ser consideradas del mismo usuario.

1 **función** similitud-con-banco-de-filtros($I_{\mathcal{T}}, I_{\mathcal{U}} : \mathbb{Z}^2 \mapsto \mathbb{R}; \tau \in \mathbb{R}$) **inicio**

2 | inicialización-de-sectores(\mathcal{S});

3 | $c_{\mathcal{T}} \leftarrow$ punto-de-referencia($I_{\mathcal{T}}$);

4 | $c_{\mathcal{U}} \leftarrow$ punto-de-referencia($I_{\mathcal{U}}$);

5 | $G_{\mathcal{T}} \leftarrow$ inicialización-de-kernels($c_{\mathcal{T}}$);

6 | $G_{\mathcal{U}} \leftarrow$ inicialización-de-kernels($c_{\mathcal{U}}$);

7 | **para cada** $\omega \in \{\frac{3}{10}, \frac{6}{10}, \frac{9}{10}\}$ **hacer**

8 | | **para cada** $\varphi \in \{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}\}$ **hacer**

9 | | | $F_{\mathcal{T}j} \leftarrow$ convolución($I_{\mathcal{T}}, G_{\mathcal{T}(\omega, \varphi)}$);

10 | | | $F_{\mathcal{U}j} \leftarrow$ convolución($I_{\mathcal{U}}, G_{\mathcal{U}(\omega, \varphi)}$);

11 | | **fin**

12 | **fin**

13 | **para cada** $x \in \Omega$ **hacer**

14 | | $J_{\mathcal{T}}(x) \leftarrow$ arg máx $_j \|F_{\mathcal{T}j}(x)\|$;

15 | | $J_{\mathcal{U}}(x) \leftarrow$ arg máx $_j \|F_{\mathcal{U}j}(x)\|$;

16 | | $C_{\mathcal{T}}(x) \leftarrow \tan^{-1} \left(\frac{\Im[F_{\mathcal{T}J_{\mathcal{T}}(x)}(x)]}{\Re[F_{\mathcal{T}J_{\mathcal{T}}(x)}(x)]} \right)$;

17 | | $C_{\mathcal{U}}(x) \leftarrow \tan^{-1} \left(\frac{\Im[F_{\mathcal{U}J_{\mathcal{U}}(x)}(x)]}{\Re[F_{\mathcal{U}J_{\mathcal{U}}(x)}(x)]} \right)$;

18 | **fin**

19 | **si** $e^{-\frac{1}{mn} \sum_x [C_{\mathcal{T}}(x) - C_{\mathcal{U}}(x)]^2} \geq \tau$ **entonces:**

20 | | **regresar verdadero;**

21 | **de lo contrario:**

22 | | **regresar falso;**

23 | **fin**

24 **fin**

Parte III

MARCO EXPERIMENTAL

IMPLEMENTACIÓN Y EXPERIMENTOS

Durante la implementación y experimentación se realizaron pruebas tanto con bases de datos de huellas dactilares artificiales como de datos naturales.

Los algoritmos descritos en los capítulos anteriores fueron implementados en Python 2.6 para prototipar su comportamiento. Además, se utilizaron librerías como OpenCV 2.4.1 para los requerimientos de visión computacional como lo son la matriz de homografía y los algoritmos para máximo ajuste de puntos FlannBasedMatcher y KNN (k-nearest neighbour, por sus siglas en inglés), mientras que para las operaciones de algebra matricial y estadística se utilizó numpy y scipy. Por otro lado, para las visualizaciones se utilizó R con ggplot2, así como matplotlib, mayavi y networkx.

4.1 PRUEBAS CON DATOS ARTIFICIALES

Los datos generados artificialmente consisten en minucias, por lo cual estos datos sólo se usaron para experimentar con los algoritmos que consisten en análisis espectral de grafos. A continuación describiremos el procedimiento mediante el cual fueron generados estos datos.

Básicamente, el procedimiento consiste en simular los puntos de interés de las huellas dactilares, es decir, las minucias. Para esto, primeramente generaremos el número N de minucias. Como se mencionó en capítulos anteriores un sensor de buena calidad produce una imagen de la cual se pueden extraer entre 70 y 80 minucias, además de acuerdo con [Ortega-Garcia et al. \(2010\)](#) basta con tener 50 minucias para poder realizar una comparación entre dos huellas, de manera que $N \sim \mathcal{U}(50, 80)$, es decir, el número de minucias se distribuye uniformemente entre el mínimo necesario para realizar la comparación y el máximo que puede dar un sensor de buena calidad.

Posteriormente, supondremos que cada una de las componentes de las minucias son independientes, es decir, una minucia \mathbf{m}_i , con $i \in [N]$, estará formada por componentes $X_i \sim \mathcal{U}(0, W)$, $Y_i \sim \mathcal{U}(0, H)$ y $\alpha_i \sim \mathcal{U}(0, 360)$, de tal forma que $W \times H$ representa la dimensión en píxeles de la imagen de la que supuestamente fueron extraídas las minucias. Para nuestros experimentos, supondremos imágenes de 512×512 píxeles. Nótese, que por la posterior normalización de los datos estas dimensiones no tienen mucha relevancia. Dado esto, procedemos a generar muestra de N minucias para simular un conjunto \mathcal{F} .

Lo siguiente es simular las huellas contra las cuales vamos a comparar el conjunto de minucias, es decir, se requiere de una huella con la cual la comparación deba dar un resultado negativo y una más con la cual el resultado de la comparación deba ser positivo.

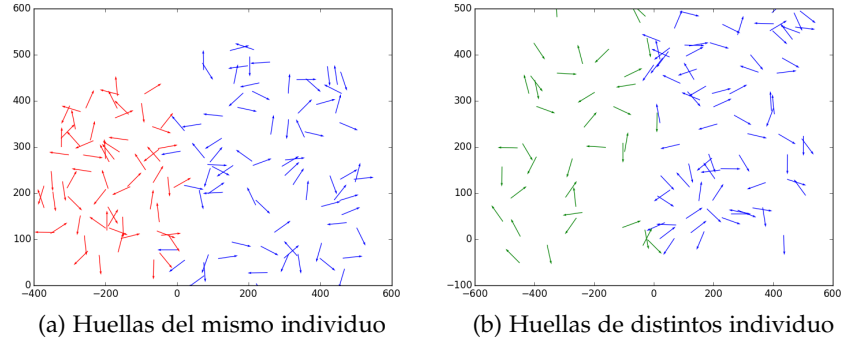


Figura 4.1.1: Datos simulados de minucias

Para el primer caso basta con generar otra huella \mathcal{H} mediante el mismo procedimiento que se generó \mathcal{F} , solamente que en este caso se generarán $M \sim \mathcal{U}(50, 80)$ minucias. En el segundo caso, el procedimiento es ligeramente más sofisticado puesto que se deben simular algunas transformaciones geométricas como traslación, rotación y escala, además de un factor de ruido (ver Figura); para simular estos factores, procederemos de la siguiente forma:

- **Traslación.** Generamos un vector aleatorio $\mathbf{t} = \begin{bmatrix} t_1 & t_2 \end{bmatrix}^\top$, de manera que $t_1 \sim \mathcal{U}(0, T_x)$ y $t_2 \sim \mathcal{U}(0, T_y)$. Posteriormente, trasladamos las componentes espaciales de cada minucia en el conjunto. En este caso T_x y T_y son los valores máximos de traslación en cada uno de los ejes.
- **Rotación.** Generamos un valor angular aleatorio $\alpha \sim \mathcal{U}(-\pi, \pi)$ para construir una matriz de rotación como la siguiente:

$$\mathbf{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

y rotamos cada una de las minucias del conjunto simplemente multiplicando por la izquierda esta matriz. Sin embargo, debemos tener precaución de sumar el ángulo α a la componente angular de la minucia.

- **Escala.** Generamos un escalar aleatorio $\kappa \sim \mathcal{U}(0, \kappa_{\text{máx}})$, para posteriormente escalar las componentes espaciales de cada minucia en el conjunto.

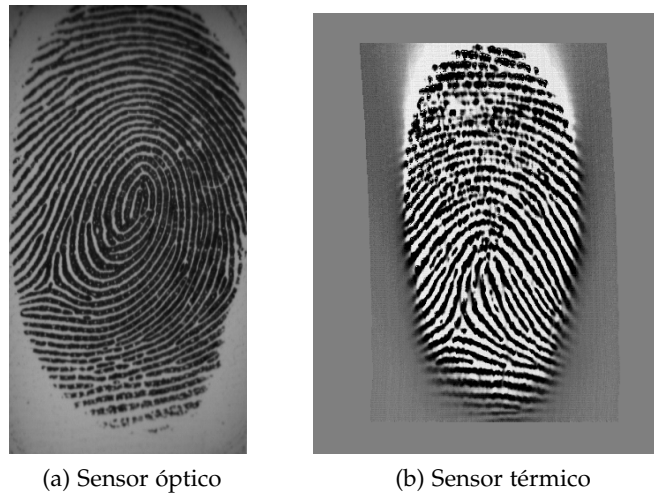


Figura 4.2.1: Huellas dactilares de dos sensores diferentes

- **Ruido.** Generamos un factor de ruido $\sigma \sim \mathcal{U}(0, 1)$ y posteriormente ruido gaussiano, es decir, $\eta \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma \mathbf{I}_{3 \times 3})$ para sumar este ruido a cada minucia del conjunto.

Finalmente, este tipo de pruebas consistieron en generar un conjuntos de entre 50 y 80 minucias para posteriormente aplicarle las transformaciones geométricas antes mencionadas y compararlos entre sí, esperando un resultado positivo, es decir, que los algoritmos basados en minucias nos arrojen que efectivamente las dos huellas simuladas pertenecen al mismo sujeto; mientras que, al generar pares de conjuntos de forma independiente, se esperaban resultados negativos, es decir, que las huellas simuladas no pertenecen al mismo individuo.

4.2 PRUEBAS CON DATOS REALES

Se trabajó con datos reales provistos por la Universidad Autónoma de Madrid a través de (Fiérrez *et al.*, 2013) de una base de datos es denominada BioSecure. Los datos contienen información obtenida a través de sensores térmicos y ópticos (ver Figura 4.2.1). Los datos de cada tipo de sensor se dividen en tres conjuntos de datos de 70 usuarios cada uno. De cada usuario se tienen 4 impresiones de un solo dedo de la mano.

En los experimentos se utilizaron sólo las huellas de sensores ópticos. Cada impresión de una huella dactilar de estos sensores está formado por dos archivos con el mismo nombre pero distinta extensión:

- **Imagen.** Es un archivo en formato **PGM** que contiene la información en bruto leída por el sensor. Las dimensiones de estas imágenes son de 296×560 píxeles.

- **Minucias.** Es un archivo en formato de texto plano con extensión **xyt** sin ningún tipo de encabezado. En cada línea de los archivos hay 4 valores enteros separados por un espacio. Estos valores representan las coordenadas (x, y) de la minucia, su orientación α y la calidad o confianza q .

En este caso se realizaron pruebas tanto con los algoritmos basados en análisis espectral, como los basados en análisis de Fourier. Las pruebas con ambos tipos de algoritmos fueron esencialmente idénticas, la única diferencia fueron los archivos de entrada que tomaba cada algoritmo. En el caso de los algoritmos de análisis espectral de grafos tomamos los archivos de texto con las minucias; y en el caso de los bancos de filtros de Gabor tomamos las imágenes.

De manera general, estas pruebas consistieron en tomar un conjunto de 70 usuarios con sus 4 impresiones, es decir, los datos de 280 huellas dactilares y comparar cada huella con el resto. Posteriormente, se miden los resultados mediante una comparación de las distribuciones de las medidas entre pares de huellas del mismo individuo y de individuos distintos.

4.3 RESULTADOS

Una vez descritos los datos utilizados y el proceso de pruebas, se procedió con el ajuste de parámetros para los distintos métodos. En el caso de los métodos basados en minucias, se utilizó el mismo umbral de distancia $\delta = 0.25$ para decidir si dos minucias están suficientemente cercanas.

Para el caso del algoritmo basado en la matriz de sociabilidad se utilizó un grado de sociabilidad $r = 40$, mientras que se escogieron los primeros $K = 25$ componentes principales. Se utilizaron cuatro de las métricas de distancia mencionadas en la Sección 3.1 y los resultados de este enfoque se muestran en la Figura 4.3.1. En color azul se muestra la distribución de los usuarios genuinos mientras que en color rojo la distribución de los impostores. Observamos que, el algoritmo resulta ser muy duro, puesto que los pares de huellas que son del mismo individuo pero con transformaciones fuertes, son tomados como impostores, es decir, existe mucho traslape entre la distribución de los impostores y los genuinos. La dureza del algoritmo la podemos controlar mediante los parámetros r y K , ya que cuanto mayor sean los valores de éstos, se obtienen mayores detalles acerca del comportamiento del grafo generado. Sin embargo, subir el valor de r provoca que se pueda causar un desbordamiento fácilmente ya que el número de caminos de tamaño r que salen del vértice i crece de forma muy acelerada. En estas pruebas se utilizó un umbral $\tau = 0.7$ para la similitud que fue determinado de forma experimental. En la Tabla 4.3.1 se muestran algunos datos para resumir los resultados con las distintas normas utilizadas. Los datos mostrados son el tiempo promedio de comparación, el porcentaje de falsos positivos y negativos, y finalmente el porcentaje de efectividad. Estos datos fueron tomados después de 39,340 comparaciones.

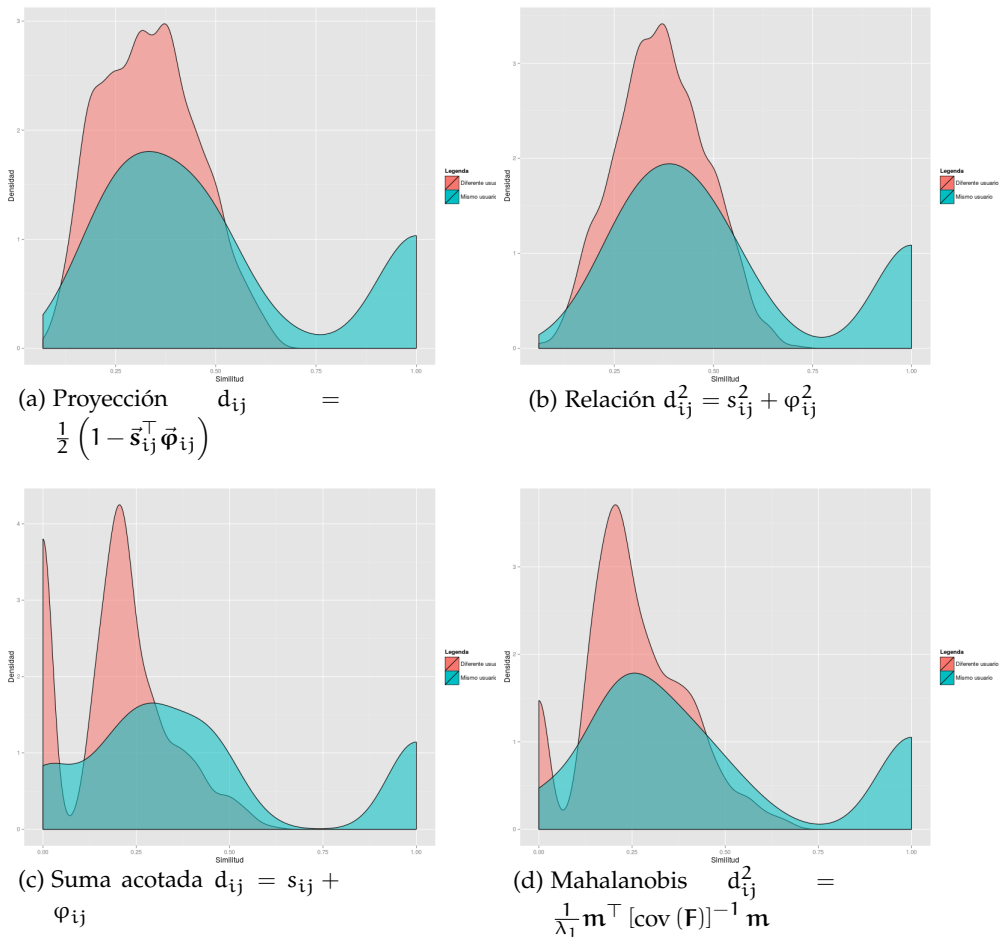


Figura 4.3.1: Resultados del Algoritmo basado en la Matriz de Sociabilidad

Cuadro 4.3.1: Estadísticas sobre el comportamiento del Algoritmo 8

Norma utilizada	Tiempo	Falsos positivos	Falsos negativos	Efectividad	Sensibilidad
Proyección	0.042seg	5.25 %	6.25 %	88.50 %	37.50 %
Relación	0.037seg	6.12 %	6.62 %	87.25 %	33.80 %
Mahalanobis	0.059seg	5.87 %	6.87 %	87.25 %	31.30 %
Norma acotada	0.027seg	5.25 %	6.06 %	88.68 %	39.40 %

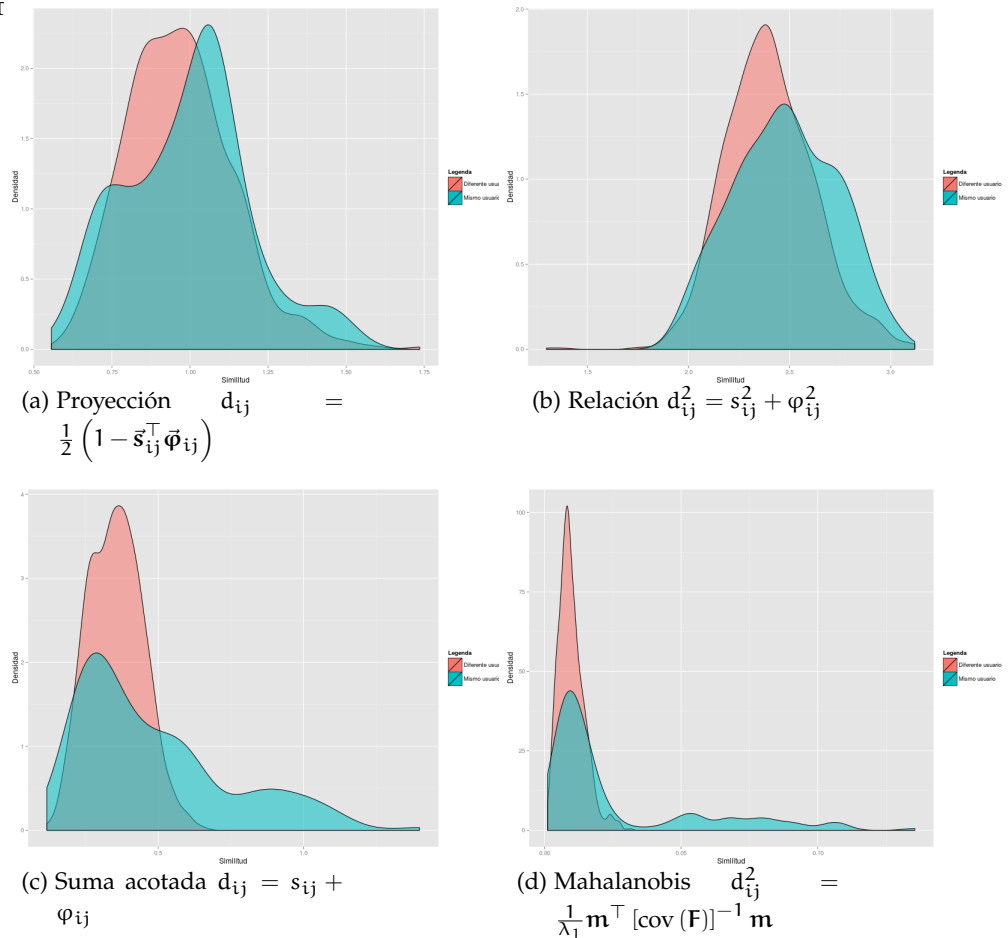


Figura 4.3.2: Resultados del Algoritmo basado en el esquema espectral

Adicionalmente se realizaron experimentos con el método del esquema espectral. Para lo cual se tomaron de forma empírica los $k = 15$ vecinos más cercanos de cada minucia y una longitud de caminos $r = 3$, con los cuales se obtuvieron los resultados mostrados en la Figura 4.3.2. A diferencia del método anterior las medidas de similitud de este no están bien acotadas, por lo que se utilizaron distintos umbrales τ dependiendo de la distancia entre minucias utilizada.

Las estadísticas de este algoritmo se presentan en la Tabla 4.3.2. De igual forma que con el método anterior, estos datos fueron tomados después de 39,340 comparaciones.

Cuadro 4.3.2: Estadísticas sobre el comportamiento del Algoritmo 9

Norma utilizada	Tiempo	Falsos positivos	Falsos negativos	Efectividad	Sensibilidad
Proyección	0.037seg	45.00 %	3.93 %	51.06 %	60.70 %
Relación	0.039seg	42.62 %	3.75 %	53.62 %	62.50 %
Mahalanobis	0.053seg	21.62 %	5.00 %	73.37 %	50.00 %
Norma acotada	0.035seg	27.25 %	5.00 %	67.75 %	50.00 %

Por otro lado, con el método basado en bancos de filtros se obtuvieron se utilizaron (además de las 8 orientaciones ya mencionadas en capítulos anteriores) 3 frecuencias ω , que fueron 0.3, 0.6 y 0.9; dando un total de 24 filtros. El umbral que se utilizó fue $\tau = 0.7$. Los resultados mostrados en la Figura 4.3.3, mientras que sus estadísticas se listan a continuación:

- Tiempo de ejecución: 28.403 seg en promedio para un par de huellas.
- Falsos positivos: 3.18 %
- Falsos negativos: 4.83 %
- Efectividad: 91.98 %
- Sensibilidad: 51.70 %

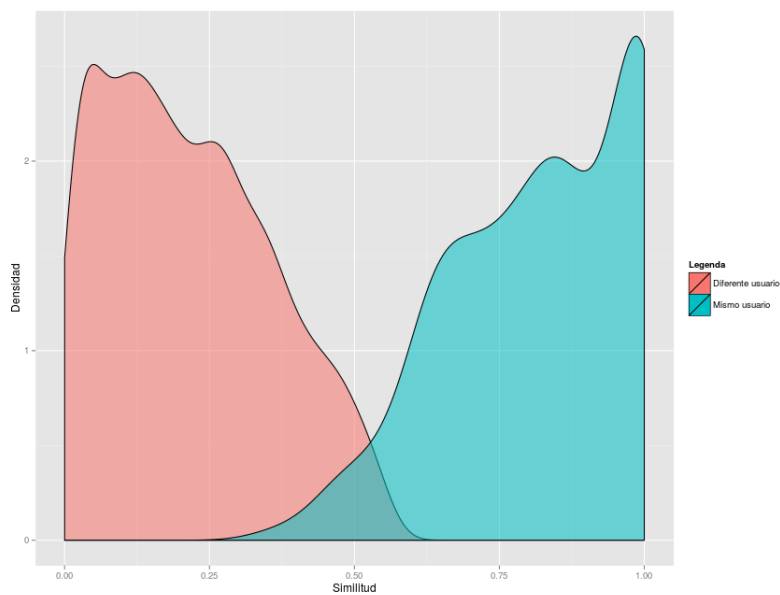


Figura 4.3.3: Resultado del Algoritmo basado en bancos de filtros

CONCLUSIONES

En los resultados mostrados podemos observar algunos detalles importantes, por ejemplo la distancia de Mahalanobis normalizada es bastante costosa debido al cálculo de la matriz de covarianza de las minucias de ambas huellas en cada comparación, sin embargo de manera general ofrece un buen resultado. Aunque, aún es mejor el resultado y la velocidad del uso de la suma acotada.

Los resultados mostrados por el método del esquema espectral en general son bastante pobres, aunque el método es considerablemente velóz, es necesario considerar algún factor para poder hacerlo más robusto. Podemos destacar que se comporta de mejor forma con el uso de la norma de Mahalanobis y la suma acotada. Es importante destacar, que en este caso el cálculo de la matriz de covarianza implicada no resulta tan costoso debido a que se procesan las minucias como un solo conjunto, a diferencia de los otros dos métodos que tratan los datos de ambas huellas por separado durante gran parte del proceso. También podemos destacar, que con tanto con la distancia de Mahalanobis como con la suma acotada, el método da como resultado puntajes muy cercanos a cero cuando se intentan comparar huellas de distintos usuarios. En este método los parámetros r (número de caminos) y k (cantidad de vecinos más cercanos o potenciales para registro) provocan que el puntaje de similitud crezca considerablemente, además de que pueden bajar el rendimiento del algoritmo debido a que habrá que calcular más potencias de la matriz de adyacencia y el grafo se vuelve más denso por al aumentar la cantidad de vecinos.

El método propuesto utilizando bancos de filtros de Gabor es mucho más costoso computacionalmente, sin embargo, es mucho más robusto y preciso que los de análisis espectral, y en consecuencia puede ser más seguro en sistemas de seguridad.

Por otro lado, a pesar de que el método propuesto usando filtros de Gabor es más costoso, una buena idea puede ser utilizarlo en conjunto con los métodos de análisis espectral de grafos, por ejemplo, supongamos que tenemos una base de datos con N huellas (con N suficientemente grande, es decir, del orden de 10^4 o superior) y queremos agrupar las más similares, es decir, sabemos previamente que existen repeticiones de huellas del mismo individuo; en este caso podemos utilizar los algoritmos basados en análisis espectral para realizar la comparación de las huellas tomando todos los pares posibles y descartar rápidamente las huellas que son muy diferentes, para posteriormente utilizar el método de banco de filtros de Gabor con conjuntos bastante reducidos de huellas y obtener resultados más precisos.

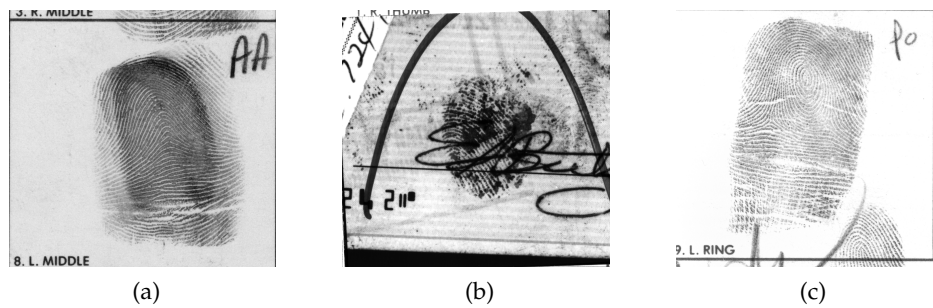


Figura 5.1.1: Huellas de la base de datos NIST-SD27

5.1 TRABAJO FUTURO

Las pruebas que se realizaron fueron hasta cierto punto muy básicas, sería interesante manejar otro tipo de pruebas más robustas además de usar otros conjuntos de datos. Por ejemplo, en las pruebas descritas en el capítulo anterior sólo se experimentó con los datos provenientes de sensores ópticos. Los datos de sensores térmicos tienen la misma estructura, la diferencia quizá solo sea que tienen distinta resolución (400×500 píxeles); sin embargo, son más engañosos ya que el sensor captura imágenes con distorsiones bastante fuertes y en consecuencia la extracción de minucias es mucho menos precisa que en el caso de las imágenes capturadas por sensores ópticos, tan es así que no basta que un simple ajuste de los parámetros en los algoritmos para realizar pruebas efectivas.

Otro conjunto de datos provisto por la Universidad Autónoma de Madrid (Fierrez *et al.*, 2013) corresponde a una base de datos denominada NIST-SD27. Esta base de datos contiene muestras forenses de huellas dactilares que son bastante peculiares e interesantes (ver Figura 5.1.1), ya que las muestras son tomadas en un ambiente que podemos denominar no controlado debido a diversas situaciones:

- El usuario no intentó proporcionarlas deliberadamente.
- No se utilizan sensores convencionales para este fin.
- Algunas son tomadas de documentos que no sólo contienen la huella sino otras características como letras manuscritas o a computadora.

En esta base de datos también contiene información sobre las minucias extraídas de las imágenes. Sin embargo, la cantidad de minucias extraídas es considerablemente menor (entre 10 y 25 minucias por huella dactilar). Además, los archivos de texto tienen una estructura ligeramente diferente; por ejemplo, cada línea contiene un 5 valores enteros y en ellos no está incluido el componente de confianza.

Por cuestiones de sólo realizar un prototipo, los algoritmos fueron implementados en un lenguaje interpretado, lo cual no es óptimo en cuestiones de cómputo, por lo que si son

implementados en un lenguaje compilado como C o C++ se deberían tener tiempos de ejecución considerablemente mejores. Aún más, existen varias operaciones en los algoritmos que son posible ejecutarlas en paralelo, entonces al implementar estos algoritmos en C o C++ se pudiera utilizar OpenMP o alguna tecnología similar para paralelizar tareas. algunas de las tareas que se pueden paralelizar son las siguientes:

- En todos los tres enfoques mostrados se realiza procesamiento en ambas huellas a comparar de manera independiente como:
 - Los procesos de normalización.
 - El cálculo de distancias entre minucias.
 - El cálculo de matrices de sociabilidad.
 - La búsqueda del punto de referencia.
- El proceso de filtrado en las distintas orientaciones y frecuencias.
- Operaciones matriciales en general.

Una aplicación más elaborada se puede realizar al observar que con los métodos de bancos de filtros se puede obtener un número considerablemente grande de características para una sola huella dactilar; en particular 640 para el método mostrado en el Sección 2.2 propuesto en el estado del arte. Entonces, si se tiene una base de datos muy grande de huellas y se sabe que existen huellas del mismo individuo, es posible extraer las características, reducir la dimensionalidad por medio de PCA y posteriormente agrupar las huellas similares en un espacio de dimensión mucho más bajo.

Otra aplicación de los algoritmos espectrales pudiera en la construcción de la conectividad cerebral del ser humano mediante difusión en imágenes de resonancia magnética (MRI por sus siglas en inglés), quizá puede ser posible identificar materia gris, materia blanca, entre otras cosas.

También resulta interesante el poder hacer más robustos los métodos basados en análisis espectral. Por ejemplo, a la matriz de sociabilidad se le podrían asociar otras características del grafo como una medida de centralidad de los nodos. Mientras que al método del esquema espectral se le puede combinar con alguna medida que indique, que tan bipartito es el grafo, es decir, que tanto se pueden alienar los vértices de un conjunto y otro.

5.2 RECOMENDACIONES

En este documento se utilizaron como normas de distancia las mencionadas en el Sección 3.1, quizá sea posible experimentar con algunas otras (por ejemplo Chebyshev o correlación). Sin embargo, se recomienda normalizar esta distancia como en el caso de uso de la distancia Mahalanobis.

El ajuste de los parámetros puede ser muy complicado dependiendo de la calidad de los datos de entrada. No se recomienda el uso de estos métodos si los datos de entrada no cumplen los requerimientos mencionados en capítulos anteriores. Por ejemplo en el caso de los métodos basados en minucias, el tamaño de la muestra es muy importante, ya que si la muestra fuera menor al grado de sociabilidad, es decir $n < r$, no sería posible usar PCA, esto debido a que tendríamos menos observaciones que variables, por lo que nuestra matriz de datos no podría ser de rango completo y en consecuencia no podríamos construir un espacio vectorial de r vectores linealmente independientes de dimensión r que correspondería con el espacio de los componentes principales; una alternativa para poder abordar este problema es usar Descomposición en Valores Singulares (SVD) de $\mathbf{X}\mathbf{X}^T$ y aprovechar la relación con los valores singulares de $\mathbf{X}^T\mathbf{X}$.

BIBLIOGRAFÍA

- Ansari, N., Chen, M., y Hou, E. (1992). A Genetic Algorithm for Point Pattern Matching. *Dynamic, Genetic and Chaotic Programming*. (Cita en la página 18)
- Ballard, D. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 3(2):110–122. (Cita en la página 17)
- Bishnu, A., Das, S., Nandy, S., y Bhattacharya, B. (2006). Simple Algorithms for Partial Point Set Pattern Matching Under Rigid Motion. *Pattern Recognition*, 39(9):1662–1671. (Cita en la página 17)
- Bishop, C. M. (2006). *Pattern Recognition in Machine Learning*. Springer. (Citas en las páginas 7 y 30)
- Daugmann, J. (1985). Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1169. (Cita en la página 22)
- Daugmann, J. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1148–1161. (Cita en la página 22)
- Estrada, E. (2012). *The structure of complex networks : theory and applications*. Oxford University Press, New York. (Cita en la página 9)
- Fierrez, J., Krish, R. P., Ramos, D., Veldhuis, R., y Wang, R. (2013). Evaluation of afis-ranked latent fingerprint matched templates. *6th Pacific-Rim Symposium on Image and Video Technology*, pp. 230–241. (Citas en las páginas 43 y 50)
- Jain, A., Prabhakar, S., y Hong, L. (1999). A Multichannel Approach to Fingerprint Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):348–359. (Citas en las páginas 9, 19 y 23)
- Jain, A., Prabhakar, S., Hong, L., y Pankanti, S. (2000). Filterbank-Based Fingerprint Matching. *IEEE Transactions on Image Processing*, 9(5):846–859. (Citas en las páginas 11, 19 y 20)
- Jain, A., Prabhakar, S., y Pankanti, S. (2002). On the Similarity of Identical Twin Fingerprints. *Pattern Recognition*, 35(11):2653–2663. (Cita en la página 3)
- Kücken, M. (2007). Models for Fingerprint Pattern Formation. *Forensic Science International*, 171(2–3):85–96. (Cita en la página 3)

- Kücken, M. y Newell, A. (2005). Fingerprint Formation. *Journal of Theoretical Biology*, 235(1):71–83. (Cita en la página 3)
- Luo, B., Wilson, R. C., y Hancock, E. R. (2003). Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213–2230. (Cita en la página 32)
- Maltoni, D., Maio, D., Jain, A., y Prabhakar, S. (2009). *Handbook of Fingerprint Recognition*. Springer, Biometric Systems Lab (DEIS), Universita di Bologna. (Cita en la página 15)
- Ortega-Garcia, J., Fierrez, J., Alonso-Fernandez, F., Galbally, J., Freire, M. R., González-Rodríguez, J., y García-Mateo (2010). The multiscenario multienvironment biosecure multimodal database (bmdb). *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1097–1111. (Cita en la página 41)
- Rivera, M., Cedeño, O. D., Mio, W., y Ramirez-Manzanares, A. (2012). Spatial sampling for image segmentation. *Comput. J.*, 55(3):313–324. (Cita en la página 6)
- Rosenfeld, A. y Kak, A. (1976). *Digital Picture Processing*. Academic, New York. (Cita en la página 17)
- Starink, J. P. P. y Backer, E. (1995). Finding point correspondences using simulated annealing. *Pattern Recognition*, 28(2):231–240. (Cita en la página 18)
- Sussman, D. L., Tang, M., Fishkind, D. E., y Priebe, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *ArXiv e-prints*. (Cita en la página 9)

