

Reconocimiento de las personas por la huella dactilar

Francesc Serratosa

PID_00195450



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

Introducción	5
Objetivos	7
1. Dispositivos de captura de la huella dactilar	9
1.1. Sensores ópticos	10
1.2. Sensores de estado sólido	11
1.3. Sensores de ultrasonido	12
2. Análisis y representación de la huella dactilar	14
2.1. Nivel 1: patrón global de las cordilleras y valles	14
2.2. Nivel 2: <i>minutiae</i> de los dedos	16
2.3. Nivel 3: formas, poros y contornos de los dedos	18
3. Procesado de la imagen de la huella dactilar	19
3.1. Imágenes de la orientación y de la consistencia local	19
3.1.1. Método basado en el gradiente	20
3.2. Matriz de la coherencia	26
3.3. Imagen segmentada	27
3.3.1. Algoritmo de Bresenham	30
4. Detección de singularidades y del núcleo	31
4.1. Detección de singularidades	31
4.2. Detección del núcleo	34
5. Mejora de la imagen de la huella dactilar	36
6. Detección de las <i>minutiae</i>	39
6.1. Proceso de binarización	39
6.2. Proceso de esqueletonización	41
6.3. Proceso de extracción de las <i>minutiae</i>	43
7. Comparación de huellas dactilares	47
7.1. Técnicas basadas en la correlación de imágenes	51
7.2. Técnicas basadas en la comparación de <i>minutiae</i>	52
8. Clasificación de las huellas dactilares	57
8.1. Descripción formal de las clases de huellas dactilares	58
Resumen	60

Actividades	61
Abreviaturas	64
Bibliografía	65
Anexo	66

Introducción

En este módulo, vamos a estudiar la técnica de la identificación de las personas a través de la huella dactilar. Tal y como se ha enfocado el desglose de la asignatura en módulos, es imprescindible haber leído y comprendido antes el primer módulo, “La biometría para la identificación de las personas”, así como el segundo módulo, dedicado al estudio de los errores, “Evaluación de los sistemas biométricos en aplicaciones reales”. En estos primeros módulos, explicamos los conceptos generales de los sistemas biométricos que se necesitan en algunas partes de este módulo. Además, también hay algunos conocimientos específicos de la huella dactilar, como por ejemplo los orígenes, que se ha decidido tratar en el primer módulo en lugar de en este para que quedara una documentación más compacta. Así, es importante destacar que los orígenes de la biometría aplicada a la huella dactilar es un tema que se explica en el primer módulo.

El orden en el que se tratan las técnicas y procesos en este módulo tiene cierta relación con el flujo de la información en el proceso general de la comparación de dos huellas dactilares o la busca de una huella en una base de datos. Vamos comentando estos métodos de la manera siguiente:

- Antes de nada, nos encontramos con que tenemos que escanear el dedo y generar una imagen, por este motivo, en el apartado “Dispositivos de captura de la huella dactilar” explicamos los diversos dispositivos de escáner de huellas.
- Una vez obtenida la imagen de la huella, esta se transforma en un registro donde se han extraído las características principales, en el apartado “Análisis y representación de la huella dactilar” explicamos cómo se representa una huella en técnicas que no son estrictamente una imagen.
- Pero para convertir la imagen en un registro nos hace falta todo un proceso de visión por computador donde vamos calculando una serie de imágenes intermedias que nos servirán para ir obteniendo la información de la representación que deseamos, así en el apartado “Procesado de la imagen de la huella dactilar” explicamos estas técnicas de visión por computador aplicadas a la biometría.
- Con estas nuevas imágenes, ya somos capaces de obtener el núcleo de una huella. El núcleo representa el punto central de la huella dactilar y es fundamental para poder saber cómo se ha ubicado la huella en el sensor así como deducir con qué tipo de huella nos encontramos, por eso en el apar-

tado “Detección de singularidades y del núcleo” se explica la obtención del núcleo así como otras partes de la huella llamadas singularidades.

- Por otro lado, las *minutiae* son las partes de la huella que más la caracterizan; con el objetivo de estudiar cómo se extraen, primero tenemos un apartado que vuelve a tocar temas de visión por computador y está dedicado a los métodos de mejora de la calidad de la imagen de la huella, esto se explica en el apartado “Mejora de la imagen de la huella dactilar”.
- Después, nos centramos en las técnicas usadas para extraer las *minutiae*, estas técnicas se explican en el apartado “Detección de las *minutiae*”.
- Finalmente, tenemos dos apartados más dedicados estrictamente a la biometría del dedo: el apartado “Comparación de huellas dactilares” explica cómo comparar dos dedos cuando se han extraído las características principales y el apartado “Clasificación de las huellas dactilares”, cómo clasificarlas.

Objetivos

Los objetivos básicos de este módulo son los siguientes:

1. Conocer las etapas del proceso de comparación de dos huellas dactilares o de busca de una huella en una base de datos.
2. Conocer los tipos de dispositivos para la captura de la huella dactilar.
3. Conocer las técnicas de la visión por computador aplicadas a la extracción de las características principales de la huella dactilar.
4. Representar una huella dactilar en un registro de una base de datos.
5. Comparar dos huellas dactilares.
6. Clasificar los dedos en un número pequeño de clases.

1. Dispositivos de captura de la huella dactilar

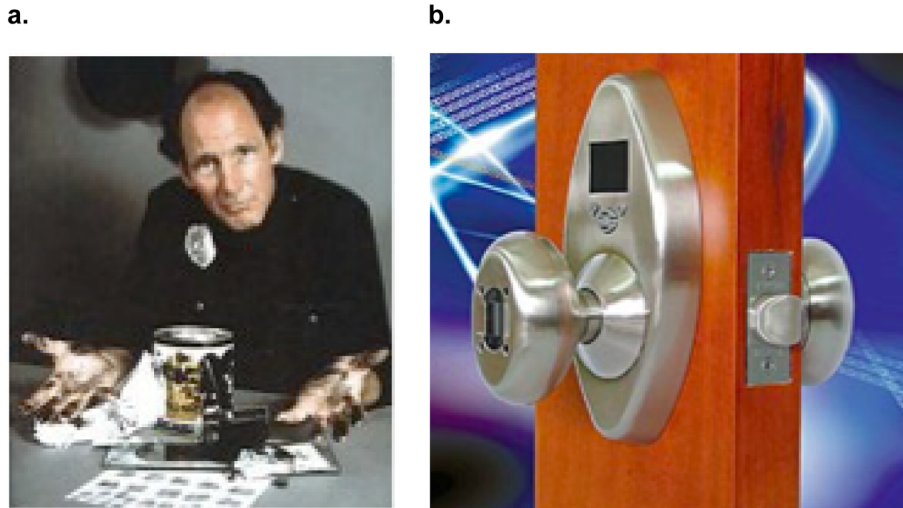
Históricamente, en las aplicaciones de las agencias del orden público, la adquisición de las imágenes de las huellas dactilares se llevaba a cabo con el **método de la tinta**. La huella dactilar de la persona que se tenía que matricular se manchaba con tinta negra y se pulsaba o se hacía rodar en una cartulina. La cartulina se escaneaba más tarde con un escáner de propósito general, que producía la imagen de la huella dactilar. Este método para adquirir las huellas se denomina **captura fuera de línea**. Hoy en día, la mayoría de las bases de datos de huellas aceptan imágenes *live-scan*, es decir, cuya imagen se ha adquirido directamente de la huella con un sensor de huellas. La tinta ya se ha eliminado y solo se requiere que el usuario ponga el dedo sobre la superficie del sensor. Aunque las bases de datos se han beneficiado de este sistema puesto que la adquisición es más rápida y las imágenes de mejor calidad, los grandes beneficiarios de la adquisición *live-scan* han sido las aplicaciones civiles y comerciales. Está claro que nadie aceptaría que se tuviera que manchar el dedo cada vez que entra en el gimnasio. No obstante, los sistemas fuera de línea siempre existirán en aplicaciones forenses donde se recogen huellas dactilares de escenas de crímenes.

En la figura 1, se muestra el sistema de adquisición de la huella dactilar con los problemas que comporta y un dispositivo actual para abrir una puerta. El dispositivo lee los patrones de cordilleras y valles que hay en la superficie del dedo y convierte la lectura analógica en digital a través de un convertidor de analógico a digital. El módulo de comunicación es el responsable de enviar la imagen así como algunos pedidos, como por ejemplo el fin del escaneo, y de recibir la orden de escanear desde un ordenador o dispositivo que incorpora un procesador. Lo que aquí denominamos sensor de huellas dactilares es solo el módulo que lee la superficie de la huella y genera una señal analógica.

Ved también

La estructura general de un dispositivo de lectura de huellas dactilares se muestra en la figura 2, subapartado 1.1 de este módulo.

Figura 1. Evolución de los sistemas de adquisición de una huella dactilar



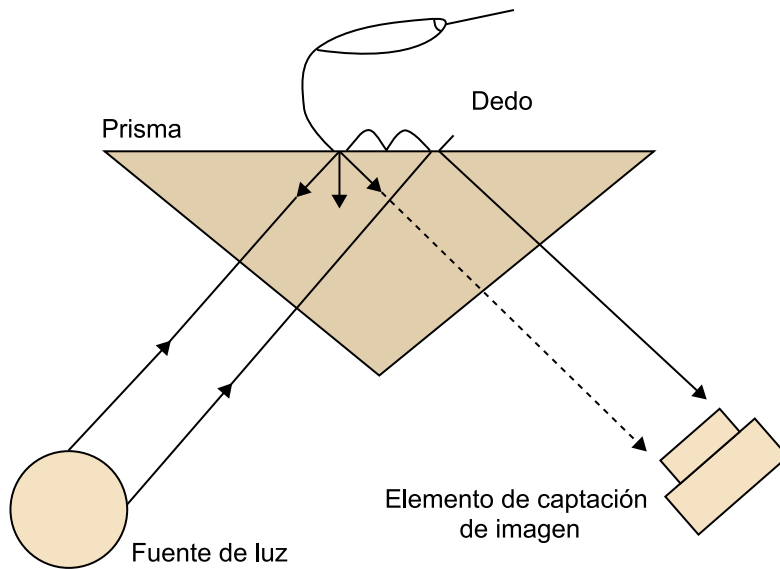
a. Sistema antiguo de adquisición de las huellas dactilares con tinta; b. dispositivo actual para leer la huella dactilar incorporado en una cerradura de una puerta

Se han diseñado muchas tecnologías diferentes para los sensores de las huellas dactilares que se pueden agrupar en familias. Algunas han llegado a comercializarse y otras no han pasado de prototipo. Aquí, vamos a comentar las tres familias más comunes (óptica, de estado sólido y ultrasonido) y, dentro de las familias, trataremos solo la tecnología más extendida.

1.1. Sensores ópticos

Los **sensores ópticos** utilizan la luz para adquirir la imagen. La tecnología más común y la más antigua es la llamada *frustrated total internal reflection* o *FTIR* (figura 2). Cuando el dedo toca la superficie de cristal o de plástico, las cordilleras están en contacto con la superficie del prisma pero los valles no. La parte izquierda del prisma se ilumina con una luz difusa. La luz que entra por el prisma se refleja en las cordilleras y la absorben las cordilleras. La carencia de reflejo permite a las cordilleras (que aparecen de color negro en la imagen) ser discriminadas de los valles (que aparecen iluminados). Los rayos de luz salen por el lado derecho del prisma y son captados a través de una lente hacia el sensor de imagen CCD. Debido a que los FTIR detectan la superficie tridimensional, es difícil que sean engañados al presentar una fotografía de un dedo. Tienen el inconveniente de que no funcionan muy bien con el dedo mojado.

Figura 2. Sensor de huella dactilar de tipo FTIR



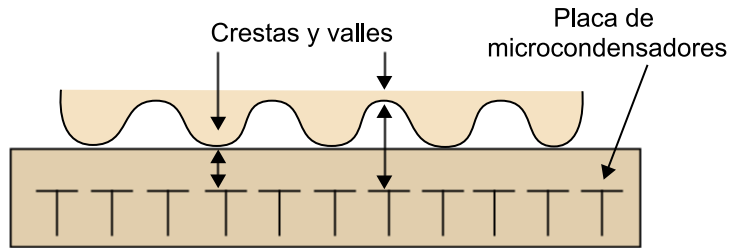
1.2. Sensores de estado sólido

Los **sensores de estado sólido** son estos sensores, también llamados sensores de silicio, que ya se propusieron en varias patentes desde la década de 1980, no obstante, hasta mediados de la década de 1990 no se comercializaron. El objetivo de estos sensores es solucionar los problemas de tamaño y coste de los sensores ópticos. Se considera que el gran despliegue de dispositivos para la captura de huellas no se hubiera llevado a cabo sin los sensores de estado sólido puesto que el precio es bastante más reducido y también el tamaño, entonces se pueden incorporar a portátiles y dispositivos de bajo coste y tamaño. Todos los sensores de estado sólido consisten en una matriz de píxeles donde cada píxel es un sensor pequeñito por sí mismo. El usuario toca directamente la superficie de silicio y no hay componentes ópticos ni sensores de captura de la imagen como CCD. Con el dedo mojado siguen funcionando bastante bien.

Las dos tecnologías principales para convertir el patrón de la huella dactilar en señales eléctricas son capacitivas y térmicas:

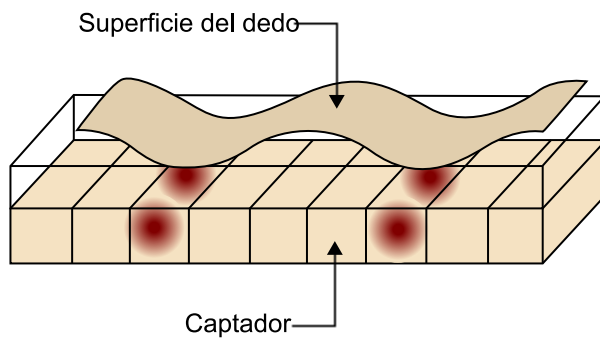
1) **Capacitivos:** Estos sensores (figura 3) detectan la diferencia de carga eléctrica entre las zonas que tocan el sensor (cordilleras) y las zonas que no tocan el sensor (valles). Son los sensores más comunes dentro de los dispositivos de estado sólido.

Figura 3. Sensor capacitivo



2) **Térmicos:** Estos sensores (figura 4) detectan la diferencia de temperatura entre cordillera y valle. Están formados por material piroeléctrico. Las cordilleras, que están en contacto con el sensor, transmiten más temperatura que los valles. Debido a que en pocos milisegundos toda la superficie se calienta, de hecho no captan los puntos calientes sino el aumento de temperatura en cada punto y en unos instantes de milisegundos, les afecta la temperatura ambiente.

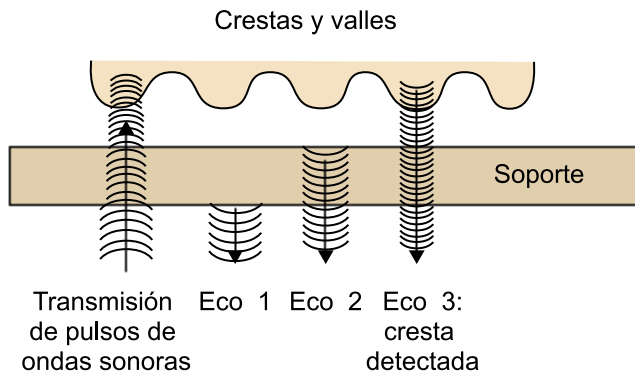
Figura 4. Sensor térmico



1.3. Sensores de ultrasonido

Los **sensores de ultrasonido** (figura 5) se pueden ver como un tipo de ecografía. La tecnología se basa en enviar señales acústicas hacia la superficie del dedo y captan el eco de la señal. El eco de la señal es el que nos aporta la información de la profundidad donde se encuentra la piel, las cordilleras muy cerca y los valles lejos. Por lo tanto, cada píxel del sensor está compuesto por dos componentes, un emisor de sonido y un micrófono. Son los dispositivos que más toleran el hecho de que los dedos estén sucios o mojados, pero tienen el problema de que son los más caros.

Figura 5. Sensor de ultrasonido



2. Análisis y representación de la huella dactilar

La **huella dactilar** es la reproducción del aspecto externo de la epidermis de los dedos. La característica más evidente de esta representación es un patrón donde destacan unas cordilleras y unos valles.

En una imagen de una huella, las cordilleras son oscuras mientras que los valles son claros (esta representación es herencia de la captura de la huella basada en el método de la tinta). La anchura de las cordilleras varía de 0,1 milímetros a 0,3 milímetros y, por lo general, el periodo entre cordilleras y valles es aproximadamente de medio milímetro. La mayoría de las heridas en los dedos, como son las quemaduras superficiales, cortes o desgaste, no afectan a la estructura que hay debajo de la epidermis y, además, el patrón original se renueva con la misma estructura cuando crece la nueva piel.

Los detalles de los patrones de las huellas se describen en tres niveles diferentes:

- **Nivel 1:** El flujo global del patrón de las cordilleras y valles que generan las **singularidades**.
- **Nivel 2:** Las terminaciones de las cordilleras, llamadas *minutiae*.
- **Nivel 3:** Los **poros de sudor**, formas pequeñas locales o los contornos de las cordilleras.

Cada uno de estos niveles genera después unos algoritmos y métodos diferentes para la comparación y clasificación de las huellas.

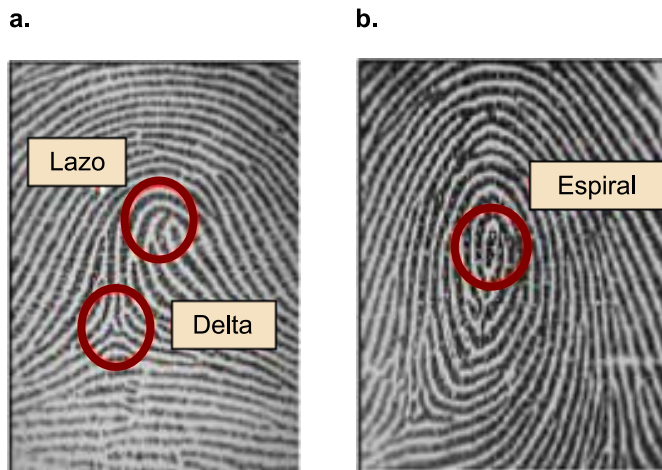
2.1. Nivel 1: patrón global de las cordilleras y valles

Normalmente, las cordilleras están ubicadas en una forma casi paralela pero hay unas pequeñas regiones donde describen unas formas distintivas puesto que generan una alta curvatura o frecuentes terminaciones de las cordilleras. Estas regiones se denominan **singularidades** o **regiones singulares** y se pueden clasificar en líneas generales en tres clases: **lazo (loop)**, **delta (delta)** y **espiral (whorl)**. Normalmente, cuando se representan sobre una imagen de una huella se caracterizan por los símbolos U, Δ y O, respectivamente.

En la singularidad de tipo lazo, las cordilleras de alrededor crean una forma que recuerda a la letra U. Por otro lado, en las singularidades de tipo delta, las cordilleras crean una forma parecida a un pico de una montaña o un triángulo. Finalmente, las cordilleras que crean un espiral tienen la forma de un círculo. Algunas veces, las singularidades de tipo espiral no se consideran puesto que

se supone que son dos singularidades de tipo lazo encaradas a muy poca distancia. Esta representación de las singularidades sin espirales tiene la ventaja de que no necesita discernir cuándo nos encontramos con una espiral o dos lazos suficientemente juntos. En la figura 6, mostramos dos huellas dactilares con las tres regiones singulares y cada una con su núcleo.

Figura 6. Dos huellas dactilares donde aparecen singularidades



a. lazo y delta; b. espiral.

Los algoritmos para comparar huellas (que describimos más adelante) pueden superponer y alinear los dedos según una marca o punto central llamado núcleo (*core*).

El **núcleo** sirve para poder deducir el desplazamiento en los dos ejes (vertical y horizontal) que se ha llevado a cabo al ubicar el dedo en el sensor, así podemos alinear correctamente las huellas dactilares a la hora de hacer la comparación.

Henry definió el núcleo de una huella dactilar en 1900 como “el lazo superior de la imagen de la cordillera más interna”. En la práctica, el núcleo se considera el centro de la singularidad más superior (sea cualquier singularidad de las tres), puesto que algunas veces no existe ningún lazo o espiral, como veremos más adelante. Además, y debido a la gran variabilidad de las huellas dactilares, algunas veces es realmente difícil localizar de forma metódica el núcleo de la huella.

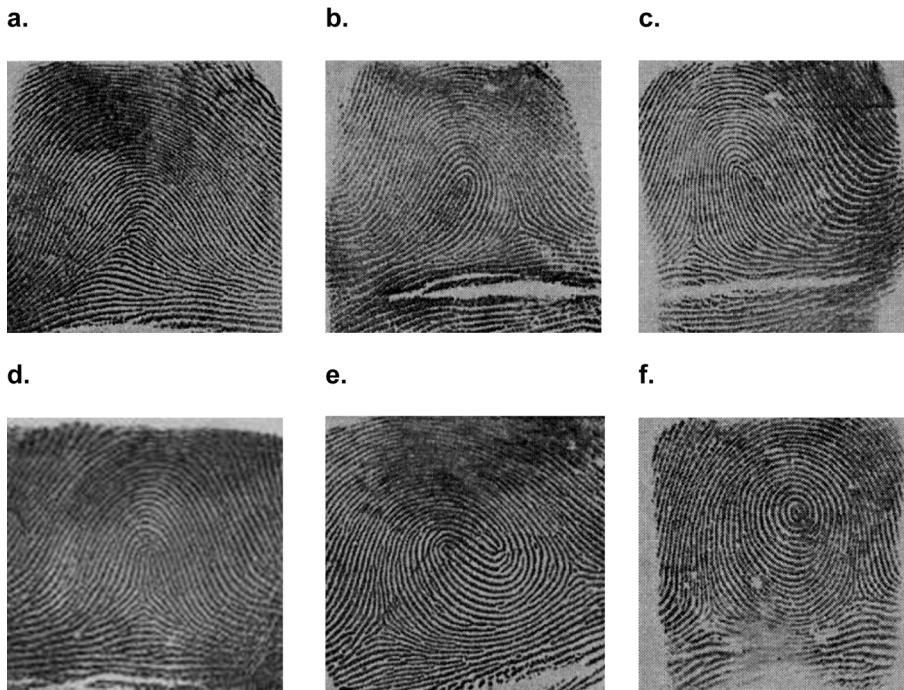
Las regiones singulares, además de ser útiles para la alineación de los dedos, se usan para clasificar los dedos, es decir, dada una huella dactilar, asignar una posible clase de un conjunto muy reducido (normalmente cuatro o cinco clases). La clasificación sirve para simplificar la busca de una huella en los sistemas de identificación y acelerar la obtención de resultados. Esto se debe a

Ved también

En el apartado “Clasificación de las huellas dactilares” del presente módulo haremos una descripción metódica de las cinco clases.

que, dada una huella dactilar, solo se compara con los dedos de la misma clase. La figura 7 muestra cinco imágenes con huellas de las cinco clases diferentes donde se puede ver que siguen patrones diferentes.

Figura 7.



Huellas dactilares de diferentes clases: a. arco; b. lazo izquierdo; c. lazo derecho; d. arco de tipo tienda; e. doble lazo; f. espiral

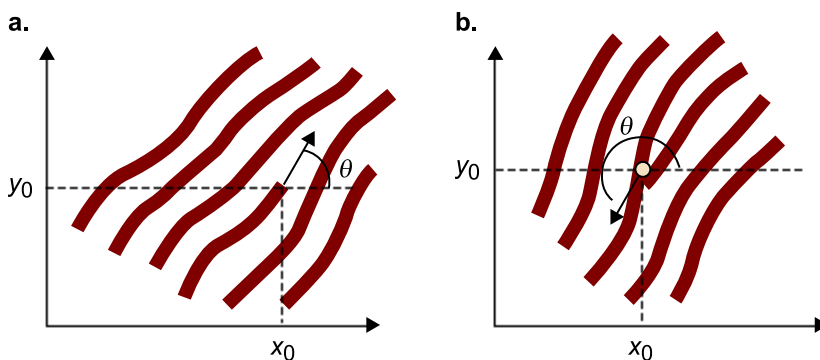
2.2. Nivel 2: *minutiae* de los dedos

Las *minutiae* son otras características muy importantes que se encuentran en los patrones de las huellas dactilares. En el contexto de las huellas, se refiere a las formas variadas como las cordilleras pueden crear discontinuidades. Por ejemplo, una cordillera se puede acabar de repente o se puede dividir en dos cordilleras. La figura 8 muestra los dos tipos de *minutiae* más comunes. También se muestra la información del ángulo de las *minutiae* así como su posición en las coordenadas (x, y) dentro de la imagen.

Etimología

Minutia quiere decir **detalle pequeño**, *minutiae* es el plural de *minutia*.

Figura 8.



Dos tipos diferentes de *minutiae*: a. terminal, y b. bifurcación

La *minutia* es la característica más usada hoy en día en el reconocimiento automático de huellas dactilares. Sir Francis Galton (1822-1911) fue la primera persona en categorizar las *minutiae* y observó, en 1892, que se mantenían invariables a lo largo de la vida de una persona. Por este motivo, hay bibliografía que llama a las *minutiae* detalles de Galton en su honor.

En una huella dactilar completa, el número de *minutiae* puede ser superior a cien. No obstante, se asume que con un número de doce a quince *minutiae* ya basta para establecer una buena comparación, si se demuestra que las impresiones son de la misma zona o el ángulo de ataque del dedo en la zona impresa es el mismo. Las estadísticas muestran que la densidad de las *minutiae* en las zonas singulares es de $0,49 \text{ minutiae/mm}^2$ y la densidad en las zonas fuera de la zona singular es de $0,18 \text{ minutiae/mm}^2$.

Aunque se pueden considerar muchos tipos de *minutiae* (como las mostradas en la figura 8), usualmente solo un grupo reducido de estas se considera para representar la huella dactilar, debido a la dificultad práctica de discernir entre clases de *minutiae* de forma automática. El **American National Standard Institute** propone una taxonomía de las *minutiae* basada en solo cuatro clases: **final de cordillera, bifurcación, compuesta (trifurcaciones o cruces) y tipo indeterminado**. Pero este estándar no es el más usado y el modelo de *minutiae* que establece el **FBI** considera solo la **final** y la **bifurcación** (figura 8). Los otros tipos no se detectan debido a la alta probabilidad de que sean clasificadas en otras clases. Nosotros usaremos a lo largo del módulo el sistema del FBI con solo dos clases de *minutiae*. En lo que sí están de acuerdo es en la forma de registrar las *minutiae*, que es la siguiente:

- **Clase:** *Minutia* final o *minutia* bifurcación (en el sistema FBI). Fijaos en que es importante que se haya estandarizado que la cordillera siempre genera valores oscuros en la imagen (valores pequeños en los píxeles) y que los valles generan valores claros en la imagen (valores altos en los píxeles). Si no fuera así, una *minutia* de tipo final se podría confundir con una *minutia* de tipo bifurcación. Este problema se denomina la dualidad de las *minutiae* final y bifurcación. Si generamos la imagen negativa de una huella dactilar, entonces las *minutiae* finales se convierten en bifurcación y viceversa.
- **Posición:** Coordenada bidimensional (x, y) absoluta dentro del área del sensor de la posición de la *minuta*. La posición $(0, 0)$ es el píxel inferior izquierdo de la imagen generada. Este dato depende de cómo se ha impreso el dedo en el sensor (o, en una huella latente, de cómo se ha fotografiado la huella). Si en la verificación el dedo se ha imprimido algo más a la izquierda que el dedo en la matriculación, entonces el valor del eje horizontal (normalmente se considera el eje x) tendrá un valor mayor en la verificación. Los algoritmos de comparación arreglan estas diferencias, que no dependen del dedo sino de cómo se ha imprimido.

Referencia bibliográfica

La taxonomía de las *minutiae* propuesta por el American National Standard Institute se recoge en el documento ANSI/NIST-ITL 1, 2007.

Una curiosidad

Es curioso que el FBI (institución pública de Estados Unidos) no haga caso de la estandarización impuesta por la ANSI (también es una institución pública de Estados Unidos).

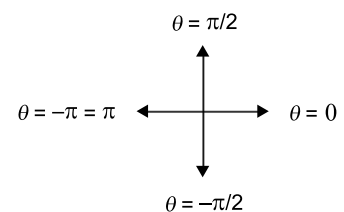


Figura 9. Orientación de los ángulos. Asumimos que el eje horizontal genera el ángulo 0.

- **Ángulo:** Ángulo generado entre la línea de la cordillera y el ángulo horizontal del sistema de coordenadas bidimensional. Este ángulo es de 360 grados $]-\pi, \pi]$ puesto que se considera que el vector director generado por la cordillera acaba en la *minutia*. De manera similar a la posición, si el usuario orienta el dedo al sensor de una manera diferente en la matriculación que en la verificación, entonces aparecerán ángulos diferentes para la misma *minutia*. Los algoritmos de comparación también arreglan estas diferencias. En la figura 9, mostramos las orientaciones de los ángulos.

La figura 10 muestra una *minutia* terminal. Fijaos en que la orientación del ángulo depende del vector director y no del tipo de *minutia*.

Con el objetivo de aumentar la operatividad a través de los algoritmos de detección y comparación de *minutiae*, se consideran otros convencionalismos sobre la orientación del ángulo y la posición de la *minutia*.

2.3. Nivel 3: formas, poros y contornos de los dedos

Las formas de las cordilleras, los poros y los contornos son los detalles más finos que se pueden extraer de las huellas dactilares. Incluyen todos los atributos dimensionales de las cordilleras, como la anchura, la forma, la largura o el contorno. También encontramos las cordilleras incipientes o pequeñas discontinuidades de estas (que no se detectan en el nivel 2, a la hora de deducir las *minutiae*). Además, también están los poros de sudor, que tienen un tamaño de 60 a 250 micras; se ha observado que el número de poros en un centímetro de una cordillera varía de 9 a 18. Los especialistas reivindican que de 20 a 40 poros de sudor pueden ser suficientes para determinar la identidad de una persona.

Aunque el nivel 3 parece que sea altamente discriminador y muy importante para los examinadores de huellas latentes, en la actualidad, muy pocos métodos para comparar huellas dactilares lo usan, ya que se necesitan escáneres de muy alta resolución (como mínimo 1.000 dpi) e imágenes de muy buena calidad. El estándar mencionado antes, el CDEFFS (2008), es el último esfuerzo para sacar adelante tecnologías con este nivel. La figura 11 muestra una huella dactilar donde son evidentes los poros.

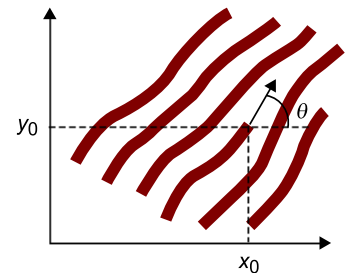


Figura 10. Ejemplo de *minutia* terminal. Se muestra el ángulo de la orientación de la cordillera.

Referencias bibliográficas

Otros convencionalismos sobre la orientación del ángulo y la posición de la *minutia* se recogen en los documentos ANSI/INCITS 378 (2004), ISO/IEC 19794-2 (2005) y CDEFFS (2008).



Figura 11. Imagen de un detalle de un dedo donde se pueden ver los poros de sudor. La imagen ha sido tomada a 1.000 dpi.

3. Procesado de la imagen de la huella dactilar

En el resto de este módulo, llamaremos **I** a la imagen original que contiene la huella dactilar. $I(x, y)$ representa el valor del píxel ubicado en la posición (x, y) de la imagen de la huella **I**. Aunque para la metodología y algoritmos tratados en este apartado es irrelevante, normalmente la posición $(0, 0)$ de la imagen se encuentra ubicada en el píxel inferior izquierdo de la imagen.

3.1. Imágenes de la orientación y de la consistencia local

La orientación local de la cordillera en el píxel (x, y) de la imagen de la huella dactilar **I** es el ángulo θ_{xy} , que forma la cordillera que hay en un entorno local a esta posición con el eje horizontal. Una posición concreta de la imagen de la orientación se describe con el símbolo $D(x, y)$:

$$D(x, y) = \theta_{xy}.$$

El primer investigador en introducir esta imagen fue **Grasselli** en 1969. Normalmente, se usa el término **orientación** para denominar un ángulo dentro del rango $]0, \pi]$ puesto que la cordillera no tiene una dirección específica (usamos solo dos cuadrantes en lugar de cuatro cuadrantes).

En el supuesto de que en la posición (x, y) no haya cordillera, $D(x, y)$ también toma un valor, aunque no tenga demasiado significado real. En el párrafo siguiente, vamos a tratar cómo se soluciona este problema. Además, fijaos en que en la caracterización de una *minutiae*, tenemos la información de la dirección, que es de 360 grados, $]-\pi, \pi]$ (aquí usamos los cuatro cuadrantes). Es decir, en las *minutiae* tenemos que concretar la dirección.

Asociado a la orientación de la cordillera θ_{xy} tenemos la consistencia r_{xy} .

La **consistencia** representa lo que nos fiamos de la información del ángulo o lo válido que es el valor del ángulo.

El valor del ángulo siempre está (es calculable) pero, en zonas donde no hay cordilleras, este valor no tiene sentido puesto que no se puede hablar de la orientación de la cordillera si esta no existe. La consistencia toma valores no negativos. También nos hace falta calcular la matriz de la consistencia **r**, que tiene exactamente las mismas dimensiones que la matriz de orientación y en cada píxel (x, y) encontramos el valor $r(x, y)$, que representa la consistencia del ángulo $D(x, y)$. En regiones de la imagen **I** donde hay cordillera pero son muy ruidosas o están seriamente corrompidas, la **r** toma valores positivos bajos. Por

Ved también

Para más información podéis ver el subapartado 7.2 del presente módulo.

otro lado, en las regiones de buena calidad con cordilleras, los valores de la consistencia son altos. La consistencia también puede servir para saber dónde se ha escaneado el dedo dentro del área del sensor. Donde no hay dedo, no hay cordilleras y entonces los valores de la consistencia son bajos.

Cada valor de $D(x, y)$ y de $r(x, y)$ se calcula como la media de la orientación y la consistencia de la imagen I en un entorno local cuadrado de $n \times n$ píxeles centrado en el píxel (x, y) . En los píxeles cercanos a los extremos de la imagen, se reduce el entorno donde se realiza la media. Hay varias técnicas para calcular estas dos matrices. Nosotros nos vamos a centrar en la más usada, que está basada en el **gradiente** de la imagen.

Otros métodos de cálculo

Otros métodos para el cálculo de las dos matrices pueden ser el método basado en las proyecciones o el método basado en el dominio frecuencial, que no se describen en este material.

3.1.1. Método basado en el gradiente

El método más simple y natural de extraer la información de la orientación y la consistencia de una huella digital se basa en calcular primero el gradiente de la imagen y después obtener las imágenes de orientación y consistencia a través del gradiente. El gradiente $\Delta(x, y)$ en el punto (x, y) de I es un vector bidimensional tal que:

$$\Delta(x, y) = [\Delta_x(x, y), \Delta_y(x, y)]$$

donde los componentes Δ_x y Δ_y son las derivadas horizontal y vertical de I en el punto (x, y) . Antes de nada, tenemos que explicar la técnica usada en la visión por computador para extraer las imágenes del gradiente horizontal $\Delta_x(x, y)$ y el gradiente vertical $\Delta_y(x, y)$. Después, pasamos a describir qué relación existe entre el gradiente de la imagen y la orientación de la cordillera. Finalmente, explicaremos la técnica para obtener la imagen de orientación y la imagen de consistencia.

Obtención de la imagen del gradiente horizontal Δ_x y vertical Δ_y

Para calcular estos gradientes, usaremos el conocido **método de Sobel**. El algoritmo de Sobel, presentado en 1968, se basa en convolucionar unas máscaras por toda la imagen. En este caso, una máscara W es una imagen cuadrada de 3×3 píxeles que contiene los pesos W_0, \dots, W_8 ubicados tal como se muestra en la figura 12.

Figura 12. Máscara de convolución genérica con la ubicación de los pesos

W_0	W_1	W_2
W_3	W_4	W_5
W_6	W_7	W_8

La convolución es una función c que, para cada píxel (x, y) de la imagen I , obtiene el valor siguiente:

$$c(x, y, W) = W_0 \cdot I[x-1, y+1] + W_1 \cdot I[x, y+1] + W_2 \cdot I[x+1, y+1] + W_3 \cdot I[x-1, y] + W_4 \cdot I[x, y] + W_5 \cdot I[x, y-1] + W_6 \cdot I[x+1, y-1] + W_7 \cdot I[x, y-1] + W_8 \cdot I[x, y-1].$$

Decimos que C es la imagen convolucionada de I y, según una máscara en concreto W , obtenemos:

$$c(x, y, W) = C[x, y].$$

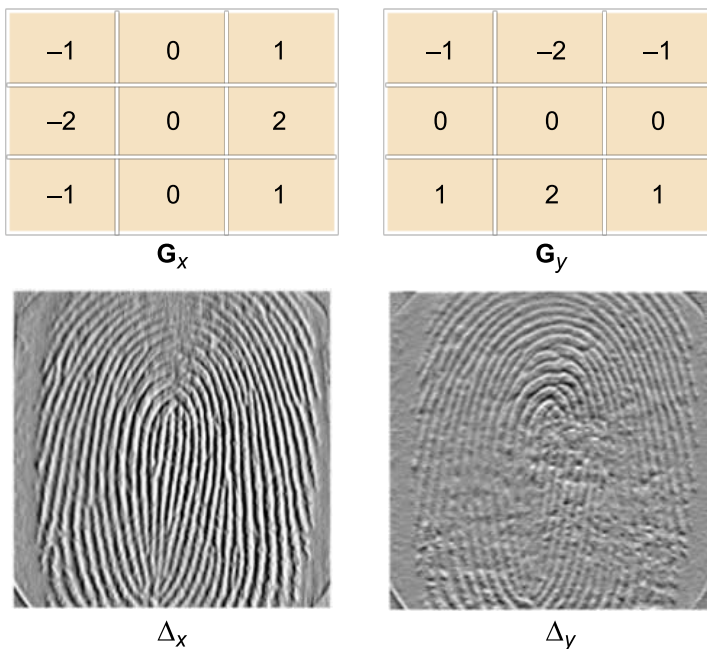
Notad que los valores de las imágenes resultado de la convolución C con cualquier tipo de máscara no tienen por qué estar dentro del rango $[0, 255]$ y por eso estas imágenes no se pueden visualizar directamente sin una normalización.

Ved también

En el anexo de este módulo, se explica el proceso de normalización de imágenes.

La figura 13 muestra las máscaras de 3×3 $W = G_x$ y $W = G_y$ usadas para obtener la imagen del gradiente horizontal Δ_x y la imagen del gradiente vertical Δ_y , junto con estas imágenes normalizadas. Fijaos en que en la imagen Δ_x se detectan las sierras verticales, en cambio, en la imagen Δ_y se detectan las sierras horizontales.

Figura 13. Máscaras para obtener el gradiente horizontal y vertical e imágenes resultantes normalizadas



Relación entre gradiente, orientación y consistencia

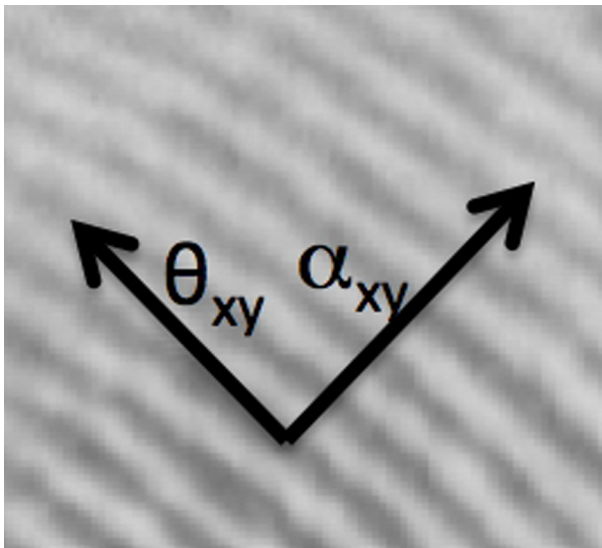
Es muy conocido que el ángulo del vector gradiente de una función denota la dirección donde está el máximo cambio de valor de la función y la magnitud del vector gradiente denota la cantidad de cambio del valor de la función. Por lo tanto, el ángulo en un punto (x, y) del vector gradiente:

$$\Delta(x, y) = [\Delta_x(x, y), \Delta_y(x, y)]$$

de la imagen I muestra dónde está el máximo cambio de intensidad de la imagen I en este punto. Por lo tanto, la magnitud del vector gradiente muestra si hay mucho cambio en los valores de los píxeles en esta orientación.

¿Dónde está el máximo cambio de intensidad en la imagen de la huella dactilar? Dada una huella, el máximo cambio de intensidad aparece en los lados de las cordilleras. Los píxeles de la cordillera tienen un valor bajo (gris tirando a negro) y los píxeles de los valles tienen un valor alto (gris muy claro, tirando a blanco). Por lo tanto, el máximo cambio de intensidad aparece en los contornos o lados de las cordilleras. La orientación θ_{xy} de una hipotética cordillera que cruza la región centrada en el punto (x, y) es ortogonal al ángulo donde se genera la máxima intensidad, es decir, es ortogonal al ángulo del gradiente en el punto (x, y) . La consistencia r_{xy} denota la magnitud (o también podríamos decir cantidad) de la variación. La figura 14 muestra una cordillera, el ángulo del gradiente α_{xy} en el punto (x, y) y la orientación de la cordillera θ_{xy} .

Figura 14. Ángulo del gradiente α_{xy} y orientación de la cordillera θ_{xy}



Con las descripciones anteriores, la imagen de orientación se define como:

$$\alpha_{xy} = \arctan\left(\frac{\Delta_y(x, y)}{\Delta_x(x, y)}\right) \quad (1)$$

donde el rango de la función es $]-\pi/2, \pi/2]$. En lenguaje de programación C o Java, esta operación se lleva a cabo con la función:

$$\alpha_{xy} = \text{atan2}(\Delta_y(x, y), \Delta_x(x, y)) \quad (2)$$

Así, se mantiene el signo del valor de las derivadas vertical y horizontal. Debido a que la orientación de la cordillera es siempre ortogonal al ángulo del gradiente, asumimos que:

$$\mathbf{D}(x, y) = \alpha_{xy} + \frac{\pi}{2} \quad (3)$$

Por lo tanto, tal como ya se ha indicado, el rango de los valores de la matriz \mathbf{D} es $]0, \pi]$.

La imagen de consistencia se define como la norma del vector gradiente de la forma siguiente:

$$\mathbf{r}(x, y) = \sqrt{\Delta_x(x, y)^2 + \Delta_y(x, y)^2} \quad (4)$$

donde el rango de la función es $[0, \infty[$.

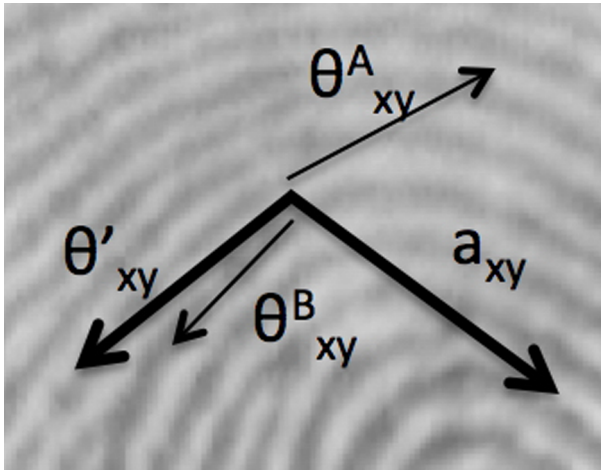
Obtención de las imágenes de orientación y consistencia

A la hora de deducir la orientación de las cordilleras a través del gradiente, no solo nos podemos centrar en el ángulo del gradiente en un píxel concreto, ya que estimaríamos la orientación a una escala demasiado fina y sería muy sensible al ruido de la imagen de la huella dactilar y la propia resolución del sensor. Este es el motivo por el que ya hemos dicho que calcularemos el valor de la orientación dada una ventana local (una zona alrededor del píxel que se va a analizar). Por otro lado, no podemos simplemente calcular la media de los ángulos dentro de esta ventana. La orientación media de dos cordilleras con orientaciones de 5° y 175° no es 90° , como resultaría de llevar a cabo una media aritmética, sino 0° . Además, el concepto de la media de la orientación no está siempre muy definido. Consideramos dos cordilleras con orientaciones de 0° y 90° . La orientación media puede ser de 45° o 135° . En la figura 15, se muestran dos cordilleras, el valor de la media de orientación angular θ'_{xy} y el valor de la media aritmética a_{xy} .

Reflexión

Debido de que el valor a_{xy} no es útil en estos métodos, no volveremos a hablar de él.

Figura 15. Dos cordilleras con sus ángulos de orientación θ_{xy}^A y θ_{xy}^B .



También se muestra la media de orientación angular θ'_{xy} y la media de orientación aritmética a_{xy} .

A continuación, describimos dos métodos para calcular la media de orientación angular. Con estas medias, se obtiene la **imagen de orientación filtrada** \hat{D} . Los algoritmos de detección de la huella dactilar, comparación y clasificación se basan en esta nueva imagen y la imagen no filtrada ya no se usa más, una vez se ha obtenido la filtrada.

El primer método fue presentado por **Kass y Witkin** en 1987. La idea básica es doblar los ángulos, así el estimador de la orientación se define por el vector siguiente:

$$\hat{\Delta} = [r \cdot \cos(2\theta), r \cdot \sin(2\theta),]$$

Fijaos en que:

$$\Delta_x = r \cdot \cos(\theta) \text{ y } \Delta_y = r \cdot \sin(\theta).$$

Por lo tanto, 2θ se usa en lugar de θ para no tener que tener en cuenta la circularidad de los ángulos. De este modo, se puede obtener un vector medio tal que:

$$\hat{\Delta}(x, y) = [\hat{\Delta}_x(x, y), \hat{\Delta}_y(x, y)] \quad (5)$$

simplemente con la media aritmética en un entorno E de la forma siguiente:

$$\hat{\Delta}_x(x, y) = \frac{1}{n^2} \sum_{(x', y') \in E(x, y)} r \cdot \cos(2D(x', y')) \quad (6)$$

$$\hat{\Delta}_y(x, y) = \frac{1}{n^2} \sum_{(x', y') \in E(x, y)} r \cdot \sin(2D(x', y')) \quad (7)$$

El entorno E representa a todos los píxeles de D dentro de una ventana local de $n \times n$ píxeles. Lo normal es que $n = 11$ pero este valor depende de la densidad de píxeles del sensor. Para obtener finalmente la matriz de orientación filtrada \hat{D} , bastará con aplicar las fórmulas expuestas antes pero considerando que el ángulo tiene que ser la mitad:

$$\hat{D}(x, y) = \frac{1}{2} \arctan\left(\frac{\hat{\Delta}_y}{\hat{\Delta}_x}\right) \quad (8)$$

El segundo método fue presentado por **Ratha, Chen y Jain** en 1995. La evaluación práctica muestra que es mucho más eficaz que el método presentado por Kass en 1987. Además, en este nuevo método se evita tener que calcular las matrices iniciales de orientación D y consistencia r puesto que trabaja directamente con las matrices de los gradientes horizontal y vertical:

$$\Delta(x, y) = [\Delta_x(x, y), \Delta_y(x, y)].$$

En la práctica, los resultados son bastante parecidos respecto al método de Kass y Witkin. Primero, se han de calcular los valores que se podrían interpretar como la covarianza y varianza entre los gradientes verticales y horizontales en un entorno E de $n \times n$ píxeles y centrado en el píxel que estamos tratando, (x, y) . La propuesta inicial de n es 9.

$$G_{xy}(x, y) = \sum_{(x', y') \in E(x, y)} \Delta_x(x', y') \cdot \Delta_y(x', y') \quad (9)$$

$$G_{xx}(x, y) = \sum_{(x', y') \in E(x, y)} \Delta_x(x', y')^2 \quad (10)$$

$$G_{yy}(x, y) = \sum_{(x', y') \in E(x, y)} \Delta_y(x', y')^2 \quad (11)$$

Después, se calcula el valor final de la media del ángulo:

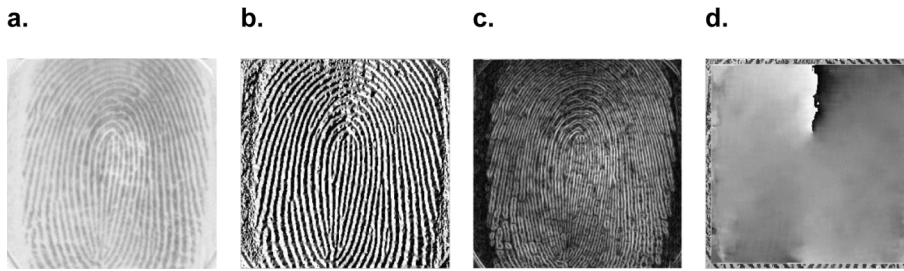
$$\hat{D}(x, y) = \frac{\pi}{2} + \frac{1}{2} \arctan\left(\frac{2G_{xy}}{G_{xx} - G_{yy}}\right) \quad (12)$$

La figura 16 muestra una imagen de una huella dactilar, la imagen que representa la matriz de orientación original, la imagen que representa la matriz de consistencia y la matriz de orientación filtrada con el método de Ratha. Todas estas imágenes se han normalizado para que los píxeles tomen un valor de 0 a 255 y se puedan visualizar. El proceso de normalización es diferente para una imagen cuyos píxeles tienen valores numéricos positivos y negativos, valores numéricos solo positivos o ángulos en radianes.

Ved también

El proceso de normalización de imágenes se explica en el anexo de este módulo.

Figura 16. Imagen real, de orientación, de consistencia y filtrada



a. Imagen que representa una huella dactilar; b. la matriz de orientación D ; c. la matriz de consistencia r ; d. la matriz de orientación filtrada \hat{D}

3.2. Matriz de la coherencia

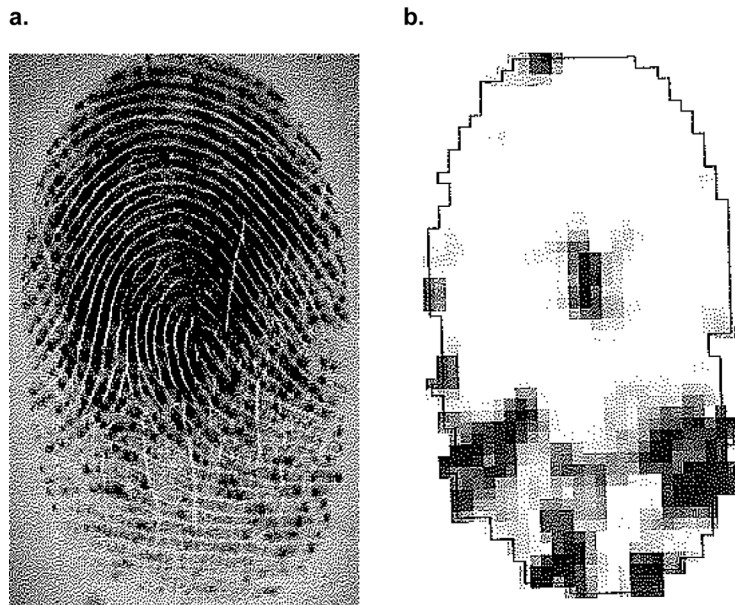
La matriz de la coherencia (C) de una huella dactilar nos informa en cada píxel del grado en el que las orientaciones son paralelas a ellas mismas en un entorno de una ventana. Esta matriz nos puede servir para ver si estamos en una zona donde hay dedo y también para detectar singularidades puesto que cambios bruscos en las orientaciones no pasan en zonas donde hay cordilleras y valles correctamente marcados y paralelos.

Kass y Witkin, en 1987, definieron la coherencia C como la norma de la suma de los vectores de orientación \hat{D} en un entorno E y normalizada de la manera siguiente:

$$C(x, y) = \frac{\left| \sum_{(x',y') \in E(x,y)} \hat{D}(x', y') \right|}{\sum_{(x',y') \in E(x,y)} |\hat{D}(x', y')|} \quad (13)$$

El rango de valores de $C(x, y)$ es $[0, 1]$. Se logra el valor máximo de uno cuando todas las orientaciones son paralelas a ellas mismas y el valor 0 cuando apuntan en direcciones diferentes. La figura 17 muestra una huella dactilar y la correspondiente imagen de coherencia. Notad que la coherencia disminuye (valores oscuros) en las zonas donde la huella es más ruidosa.

Figura 17. Imágenes real y de coherencia



a. Imagen que representa un dedo; b. matriz de coherencia C

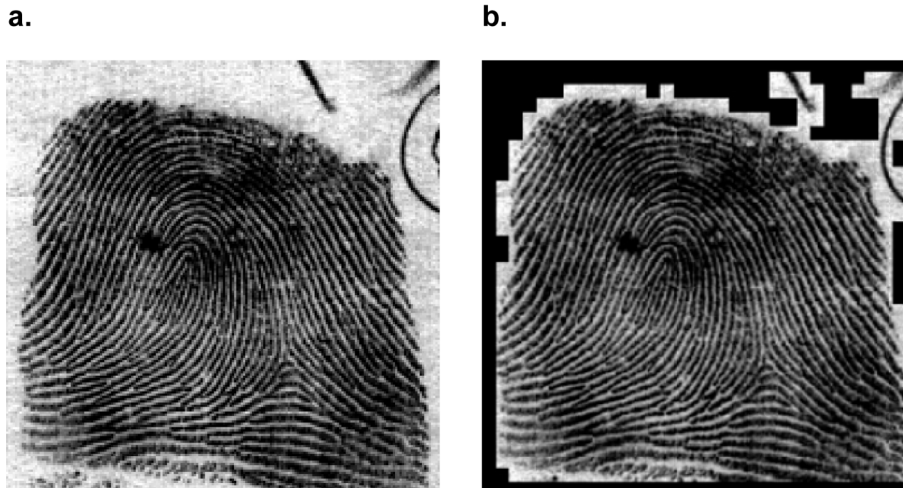
Si se usa el método de Ratha para obtener la imagen de orientación filtrada, entonces se ha demostrado que la ecuación anterior se puede simplificar por la siguiente:

$$C(x, y) = \frac{\sqrt{(G_{xx}(x, y) - G_{yy}(x, y))^2 + 4G_{xy}(x, y)^2}}{G_{xx}(x, y) + G_{yy}(x, y)} \quad (14)$$

3.3. Imagen segmentada

Por lo general, se usa el término **segmentación** para denotar la separación de la zona de la huella digital (también denominado primer plano) respecto a la zona donde no hay huella (también llamada fondo). Una imagen segmentada S es una imagen que solo contiene dos niveles de color. El negro representa el fondo y el blanco, el primer plano. La figura 18 muestra una huella dactilar y su imagen segmentada superpuesta al original. Fijaos en que hay dos rayados en el papel donde se ha imprimido la huella. En estas zonas, la imagen segmentada deduce que hay dedo. Para solucionar este problema, se tiene que llevar a cabo un tratamiento posterior por el que se aísle la forma que genera la imagen segmentada.

Figura 18. Imágenes real y representada



a. Imagen que representa un dedo; b. la misma imagen de a superpuesta a la imagen segmentada
Las zonas donde no hay dedo se han marcado en negro.

La separación del fondo es importante para evitar extraer características de la huella dactilar como son las *minutiae* o singularidades que seguro que son erróneas. Además, permite acelerar los algoritmos de extracción de estas características puesto que solo se aplican en los píxeles contenidos en el primer plano.

Debido a que las imágenes de los dedos son patrones tachados donde se alternan niveles altos de los píxeles con niveles bajos, no es válido usar un algoritmo de binarización con un umbral global. Estos algoritmos de umbral global simplemente ponen en negro los píxeles con valor inferior al umbral y en blanco los píxeles con valor superior o igual al umbral. De hecho, lo que sí discrimina entre huella y fondo es la presencia de patrones con orientación definida. En la huella, hay una orientación dominante en los píxeles en una ventana local. Por otro lado, en el fondo, el patrón es isotrópico (no hay orientación dominante).

Se han publicado varios métodos para obtener la imagen segmentada. En este módulo, vamos a describir un método fácil de entender y bastante eficaz.

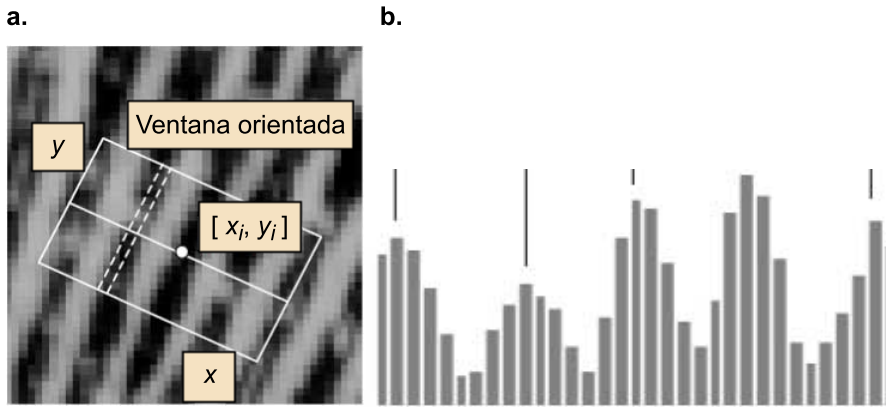
Rath, Chen y Jain, en 1995, aíslan los píxeles de la zona de la huella dactilar según la varianza de los valores de la imagen original I en un segmento local y unidimensional y orientado según el ángulo ortogonal a la orientación de las cordilleras obtenido de la imagen de la orientación \hat{D} .

Antes de nada, se tiene que definir qué es una firma.

Una **firma** de la imagen original I se identifica como $f(x, y, n, \alpha)$ y es un segmento recto unidimensional centrado en el píxel (x, y) y que tiene n elementos y está orientado según el ángulo α .

El valor del elemento i -ésimo de la firma se representa por f_i . En la figura 19, se muestra a la izquierda una imagen con una huella dactilar donde se resalta la firma con una línea blanca. A la derecha, los valores de la firma y la relación entre las posiciones bidimensionales (x, y) de la imagen y las unidimensionales de la firma.

Figura 19. Imagen real y firma



a. Imagen que representa un dedo; b. una firma

Normalmente, n toma un valor impar para que (x, y) esté centrado en medio de la firma. Si no es impar, simplemente queda desplazado un elemento, pero este hecho no es problemático para la obtención de la segmentación. Por otro lado, si el píxel central (x, y) se encuentra ubicado cerca de un margen de la imagen, entonces puede ser que n se tenga que reducir para que la firma no quede fuera de la imagen. Finalmente, a la hora de decidir los n píxeles que componen la firma, se tienen que hacer aproximaciones, puesto que es usual que no se obtengan valores naturales en las posiciones (x, y) cuando se aplica el ángulo α . Uno de los métodos más usados para decidir por qué píxeles de una imagen pasa una recta con ángulo α es el clásico **método de Bresenham**.

Ved también
El método de Bresenham lo describimos en el subapartado 3.3.1.

Dada una imagen I , su correspondiente imagen de orientación filtrada \hat{D} , un valor de n (normalmente toma valores cerca de $n = 15$) y un umbral t , el método de Ratha calcula para cada píxel (x, y) si $S(x, y)$ es fondo (negro o valor 0) o si es huella (blanco o valor 255) tal como se expone a continuación.

Para todo (x, y) calcula:

$$f\left(x, y, \hat{D}(x, y) - \frac{\pi}{2}\right) \quad (15)$$

El término $-\frac{\pi}{2}$ sirve para que la firma sea ortogonal a la cordillera. Deseamos que el ángulo de la firma sea ortogonal a la orientación de la cordillera, por eso:

$$\alpha = \hat{D}(x, y) - \frac{\pi}{2} \quad (16)$$

y entonces:

$$\alpha = \left] -\frac{\pi}{2}, \frac{\pi}{2} \right] \quad (17)$$

Una vez obtenida la firma f , calcula su varianza $v(f)$. Si $v(f) \neq t$, entonces hay poca variación en los valores de los píxeles y , por lo tanto, se considera que es fondo. De lo contrario, se considera que es huella, ya que hay un patrón con mucha variación.

La varianza se calcula con la siguiente fórmula:

$$v(f) = \frac{1}{n} \sum_{i=1}^n (f_i - \mu(f))^2 \quad (18)$$

donde $\mu(f)$ representa la media de la firma, es decir:

$$\mu(f) = \frac{1}{n} \sum_{i=1}^n f_i \quad (19)$$

3.3.1. Algoritmo de Bresenham

El **algoritmo de Bresenham**, del año 1962, determina qué píxeles se seleccionan en una imagen para dibujar una recta entre dos píxeles dados (x_0, y_0) y (x_1, y_1) . Supongamos que la recta que se va a dibujar no es vertical, es decir, que $x_1 \neq x_0$, ya que dibujar una recta vertical es trivial. Además, en el algoritmo que mostramos suponemos que nos encontramos con el caso de que $x_0 \leq x_1$ e $y_0 \leq y_1$. Los otros casos son muy similares.

Es uno de los primeros algoritmos diseñados en el campo de los gráficos por computador. Se han definido nuevos algoritmos para dibujar rectas y también se ha generalizado para dibujar circunferencias. Además, algunas placas gráficas lo tienen implementado en hardware.

algoritmo Bresenham(x_0, x_1, y_0, y_1)

Delta := $\frac{|Y_1 - Y_0|}{|X_1 - X_0|}$

Error := 0

$y := y_0$

para x desde x_0 a x_1

selecciona el píxel (x, y)

Error := Error + Delta

si Error ≥ 0.5 entonces

$y := y + 1$

Error := Error - 1.0

fin para

fin algoritmo

4. Detección de singularidades y del núcleo

La mayoría de los métodos presentados en la bibliografía para detectar singularidades se basan o usan la **matriz de orientación normalizada** \hat{D} . En esta asignatura, vamos a emplear el método más usado que, partiendo de esta matriz de orientación, calcula el **índice de Pointcaré**. Otros métodos exploran las características locales de la matriz de orientación. Una vez localizados los puntos singulares, procedemos a deducir el núcleo de la huella dactilar.

4.1. Detección de singularidades

Kawagoe y Tojo presentaron en 1984 un método basado en el índice de Pointcaré.

El **índice de Pointcaré** en un punto se define como la integral de los vectores de un campo de vectores a través de una curva cerrada que rodea este punto. Sirve para saber el vector resultante en ese punto.

Supongamos que G es un campo discreto de vectores asociado a una imagen de orientaciones \hat{D} . Debemos tener en cuenta que \hat{D} no es realmente un campo de vectores puesto que contiene las orientaciones de las cordilleras dentro del rango $]0, \pi]$ pero no las direcciones dentro del rango $]-\pi, \pi]$. Por este motivo, tendremos que estimar cuál es la dirección de los vectores dada su orientación. El índice de Pointcaré en el punto (x, y) de la imagen \hat{D} se calcula como se expone a continuación.

La curva que contiene los vectores está compuesta por los ocho píxeles más cercanos al punto (x, y) , que denominaremos v_0, v_1, \dots, v_7 . Es indiferente en qué píxel vecino se considera que empieza la curva, es decir, qué vecino es v_0 . Diremos que θ_k es el ángulo que corresponde al valor del píxel vecino v_k en \hat{D} (recordad que el rango de valores es de $]0, \pi]$) y α_k es la estimación que ahora explicaremos de su dirección (el rango de valores de la dirección estimada es $]-\pi, \pi]$). Entonces, el índice de Pointcaré se calcula como:

$$P(x, y) = \sum_{k=0..7} \alpha_k \quad (20)$$

Ahora toca estimar las direcciones de los vecinos. Consideramos que:

$$\alpha_0 = \theta_0.$$

Entonces, el resto de los valores de la dirección $\alpha_1, \dots, \alpha_7$ de los vecinos v_1, \dots, v_7 , según la dirección de las agujas del reloj, se calcula según indica una de las opciones siguientes:

$$\alpha_k = \begin{cases} \theta_k - \theta_{k-1} - \pi & \text{si } (\theta_k - \theta_{k-1}) > \frac{\pi}{2} \\ \theta_k - \theta_{k-1} + \pi & \text{si } (\theta_k - \theta_{k-1}) < -\frac{\pi}{2} \\ \theta_k - \theta_{k-1} & \text{otros} \end{cases} \quad (21)$$

No es difícil demostrar que el índice de Pointcaré en un punto y calculado de la forma descrita solo puede tomar cinco valores diferentes. Según estos valores, detectaremos si estamos en una singularidad y qué tipo de singularidad es:

$$P(x, y) = \begin{cases} 0 & \text{si } (x, y) \text{ no pertenece a ninguna región singular} \\ 2\pi \text{ o } -2\pi & \text{si } (x, y) \text{ pertenece a una espiral} \\ \pi & \text{si } (x, y) \text{ pertenece a un lazo} \\ -\pi & \text{si } (x, y) \text{ pertenece a un delta} \end{cases} \quad (22)$$

En la figura 20, se muestran cuatro ejemplos con los cuatro posibles valores de Pointcaré. Para que sea más fácil de interpretar, los valores de los ángulos se describen con grados $[0, 180]$. Entonces, si aplicamos la fórmula anterior, tenemos:

- 0: No pertenece a ninguna región singular.
- 180: Pertenece a un lazo.
- -180: Pertenece a una delta.
- 360 o -360: Pertenece a una espiral.

Las tablas en negrita representan los valores de la matriz de orientación. Debajo de las tablas en negrita tenemos el valor del resto de los ángulos de cada vecino. El valor en negrita del centro de las tablas que no están en negrita es el valor de Pointcaré.

Figura 20. Valores diferentes de Pointcaré según varias combinaciones de las orientaciones

45	89	135
20		120
9	0	129

45	13	135
89		99
135	189	46

25	44	46
11	180	-15
9	51	9

-44	-32	-58
-46	-360	-36
-54	-37	-53

45	89	135
89		99
141	10	12

135	89	135
89		99
45	10	12

-44	44	46
-52	-180	-36
-49	-2	-87

46	-46	-46
44	0	-36
35	-2	-87

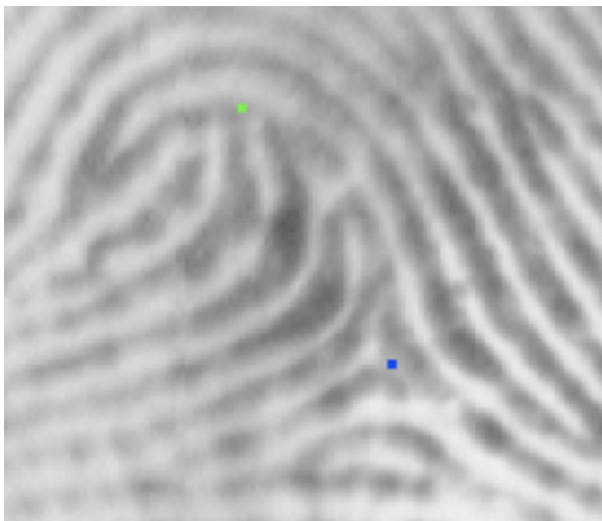
Figura 20

Por ejemplo, el valor 25 de la celda superior izquierda de la primera tabla que no está en negrita resulta de hacer la resta de los ángulos 45-20 de la tabla en negrita que tiene encima.

En la primera fila, tenemos un lazo y una espiral y, en la segunda fila, tenemos una delta y una figura sin patrón.

La figura 21 muestra un trozo de una huella donde se ha marcado una delta en azul (abajo a la derecha) y un lazo en verde (arriba a la izquierda)

Figura 21. Trozo de una huella donde se ha marcado una delta en azul (abajo a la derecha) y un lazo en verde (arriba a la izquierda)



En la mayoría de los casos, cuando se aplica este método para detectar los puntos singulares, aparece un número mayor de puntos que el real. Para solucionar este problema, los autores proponen filtrar consecutivamente la imagen de

orientación filtrada. En cada proceso de filtrado, van desapareciendo los puntos singulares falsos. De manera similar al método que usamos para obtener la matriz de orientación filtrada, calculamos para cada píxel un vector medio:

$$\hat{\Delta}(x, y) = [\hat{\Delta}_x(x, y), \hat{\Delta}_y(x, y)] \quad (23)$$

simplemente con la media aritmética en un entorno E de la forma siguiente (ahora la imagen inicial es \hat{D} en lugar D y no hace falta considerar el gradiente r):

$$\hat{\Delta}_x(x, y) = \frac{1}{n^2} \sum_{(x', y') \in E(x, y)} \cos(2\hat{D}(x', y')) \quad (24)$$

$$\hat{\Delta}_y(x, y) = \frac{1}{n^2} \sum_{(x', y') \in E(x, y)} \sin(2\hat{D}(x', y')) \quad (25)$$

Entonces, volvemos a obtener una nueva matriz de orientación filtrada:

$$\hat{D}(x, y) = \frac{1}{2} \arctan\left(\frac{\hat{\Delta}_y}{\hat{\Delta}_x}\right) \quad (26)$$

Por otro lado, también se pueden aplicar técnicas como suponer que los puntos singulares no pueden aparecer en los extremos del dedo (ayudado con la imagen segmentada). Finalmente, la matriz de la coherencia también puede ser útil para deducir las singularidades. Los puntos singulares son aquellos cuyos vectores de orientaciones no son paralelos, por lo tanto son poco coherentes.

4.2. Detección del núcleo

Una vez se ha logrado detectar los puntos singulares, la posición del núcleo se puede definir simplemente como el punto singular superior. No obstante, aparecen algunos problemas con las huellas dactilares de tipo arco que no tienen singularidades y por lo tanto el núcleo no queda definido. La localización correcta del núcleo es crucial puesto que sirve para obtener una imagen de la huella invariante al desplazamiento aplicado al dedo respecto al sensor. Un error en esta etapa supone fallos en las etapas posteriores como la comparación.

Weigstein, en 1982, propuso un método para encontrar el núcleo sin tener que localizar primero las singularidades. El método, denominado **R92**, escanea fila a fila la matriz de orientación filtrada para encontrar **arcos bien formados**. Un arco bien formado es aquel que crea un segmento horizontal de siete posiciones como el que indica la figura 22. Aunque este método es muy

heurístico, obtiene bastantes buenos resultados. Muestra de ello es el hecho de que es uno de los componentes fundamentales del sistema de identificación de huellas que usa el FBI y otras aplicaciones civiles.

Figura 22. Localización del núcleo a través de la localización previa de siete arcos bien formados

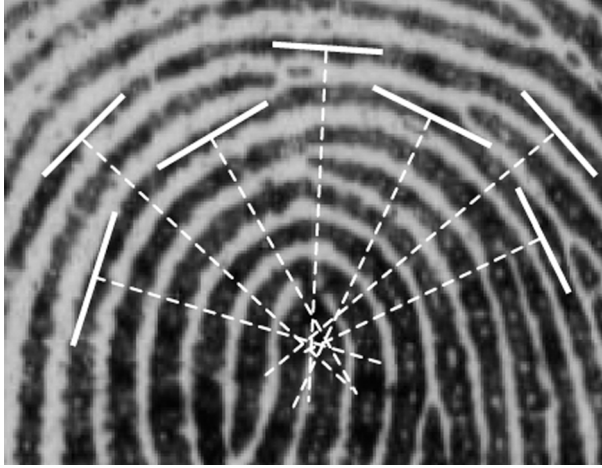


Figura 22

El núcleo se encuentra en el punto donde convergen las rectas ortogonales al segmento.

5. Mejora de la imagen de la huella dactilar

La extracción de las singularidades y del núcleo se basa en la matriz de la orientación filtrada. No obstante, para extraer las *minutiae* también se necesita, además de la matriz de la orientación filtrada, la imagen original del dedo. Este apartado, donde se explica cómo mejorar la imagen de la huella dactilar, se podría haber ubicado dentro del apartado de procesamiento de la imagen, ya que los procesos que se aplican son típicos de los métodos de procesado de imágenes. No obstante, se ha decidido ubicarlo entre el apartado de extracción de singularidades y el apartado de extracción de las *minutiae* por dos motivos, el primero es que esta mejora de la imagen no se aplica cuando se extraen las *minutiae* y el segundo es que algunas técnicas más complejas usan la matriz de orientación filtrada para mejorar la imagen.

Las técnicas aplicadas a la extracción de las *minutiae* confían en gran medida en la calidad de la huella dactilar original. En una imagen de la huella ideal, las cordilleras y los valles se alternan y fluyen de forma local en una misma dirección. En esta situación, las cordilleras se pueden detectar y ubicar fácilmente y, por lo tanto, también se pueden detectar los finales y separaciones de estas, que es donde se ubican las *minutiae*. No obstante, en la práctica y debido a las condiciones de la piel (por ejemplo, húmeda, sucia, con cortes o golpes), el ruido propio del sensor, una presión incorrecta del dedo o, simplemente, dedos con poca calidad inherente, un porcentaje importante de huellas tienen poca calidad. La figura 23 muestra tres imágenes de dedos con mucha, media y poca calidad.

Figura 23. Imágenes de dedos con diferente calidad



En muchos de los casos, las imágenes contienen regiones con una calidad buena, media y baja y la tarea del sistema automático es detectar estas regiones. De las zonas con alta calidad se extraen las *minutiae* sin problemas. Se considera que son **regiones bien definidas**. Para las zonas con calidad media, se pueden aplicar técnicas de visión por computador para mejorar la calidad. Además, se puede informar al usuario (especialista en huellas) que aquellas *minutiae* son poco fiables. Se considera que son **regiones recuperables**. De las zonas con

baja calidad no se intenta extraer las *minutiae*. Simplemente, se considera que son **zonas no recuperables**. Además, se considera que hay tres tipos diferentes de degradación. Según el tipo de degradación, se puede aplicar una técnica de recuperación diferente o tomar decisiones específicas. Los tres tipos son los siguientes:

- las cordilleras no son estrictamente continuas, es decir tienen pequeños agujeros,
- las cordilleras paralelas no están correcta o totalmente separadas, y
- hay muchos cortes, arrugas o golpes en el dedo.

Estos tres tipos de degradaciones generan problemas a la hora de extraer las *minutiae* y ubicarlas en el plano bidimensional de la forma siguiente:

- 1) se extraen falsas *minutiae*,
- 2) desaparecen *minutiae* genuinas, y
- 3) aparecen desplazamientos en la ubicación de las *minutiae* en el plano bidimensional.

Un experto en huellas dactilares tiene normalmente mucha capacidad de extraer o identificar las *minutiae* en zonas de calidad alta y media usando varios trucos visuales tales como la imagen de la orientación, la supuesta continuidad de la cordillera o la tendencia de las cordilleras en un entorno local. En teoría, si lo puede hacer un humano también se puede desarrollar un algoritmo capaz de actuar del mismo modo y de forma automática. No obstante, es difícil que estos algoritmos consigan el mismo nivel de calidad del humano.

Se han desarrollado varias técnicas para mejorar la imagen de la huella, algunas más sofisticadas que otras. Algunas usan la matriz de orientación filtrada o la matriz de coherencia. También se aplican técnicas basadas en el estudio frecuencial de la huella, como por ejemplo los filtros de Gabor. Y otras técnicas aplican métodos basados en varias resoluciones de los píxeles en la imagen. El método explicado en esta asignatura es uno de los más sencillos puesto que trabaja solo en el ámbito del píxel. Por lo tanto, no cambian las estructuras cordillera y valle que se consideren incorrectas como lo hacen los algoritmos más complejos. En particular, no es capaz de llenar pequeñas rupturas de las cordilleras, llenar agujeros dentro de las cordilleras o separar cordilleras que se han juntado debido a la humedad o demasiada presión.

Hong, Wan y Jain, en 1998, definieron una normalización de la imagen para aumentar la calidad basada estrictamente en el nivel del píxel. Esto quiere decir que el valor del píxel depende únicamente de su valor original y de algunos parámetros globales pero no de un entorno local en el píxel. Esta técnica no genera unos resultados muy satisfactorios o definitivos en imágenes muy

corrompidas, no obstante se ha usado como paso previo a otras técnicas más sofisticadas (que consideramos que están fuera del alcance de esta asignatura). El nuevo valor de cada píxel se obtiene de la manera siguiente:

$$\mathbf{I}[x, y] = \begin{cases} m_0 + \sqrt{(\mathbf{I}[x, y] - m)^2 \cdot \frac{v_0}{v}} & \text{si } \mathbf{I}[x, y] > m \\ m_0 - \sqrt{(\mathbf{I}[x, y] - m)^2 \cdot \frac{v_0}{v}} & \text{altrament} \end{cases} \quad (27)$$

donde m y v son la media y la varianza de la imagen original \mathbf{I} que se calculan de la forma siguiente:

- **Media:**

$$m = \frac{1}{n^2} \sum_{x,y=0}^{n-1} \mathbf{I}[x, y] \quad (28)$$

- **Varianza:**

$$v = \frac{1}{n^2} \sum_{x,y=0}^{n-1} (\mathbf{I}[x, y] - m)^2 \quad (29)$$

En contrapartida, m_0 y v_0 son la nueva media y la varianza deseada después de la normalización.

Kim y Park, en el 2002, introdujeron una implementación de este método que trabaja por bloques. La imagen se divide en unos cuantos bloques (se crea una cuadrícula de bloques). Entonces, se aplica la misma ecuación pero en valores diferentes para cada bloque. La media y la varianza de cada bloque de la imagen se representa por m^i y v^i , respectivamente. Además, se puede definir una media y una varianza deseada m_0^i y v_0^i diferente para cada bloque dependiendo de algunas características del bloque, como por ejemplo si se encuentra en una zona muy definida, recuperable o no recuperable.

6. Detección de las *minutiae*

La mayoría de los métodos para comparar huellas dactilares se basan en la comparación de las *minutiae*. Por eso, la detección de las *minutiae* es una tarea extremadamente importante y a la que se le han dedicado muchos esfuerzos en investigación. La mayoría de los métodos requieren convertir previamente la imagen original en binaria. Normalmente, la entrada de estos métodos es la imagen de la huella dactilar mejorada () en lugar de la imagen original. Una vez se ha obtenido la imagen binaria, se requiere un proceso de afinamiento de las cordilleras con el objetivo de que la anchura de las cordilleras sea solo de un píxel. Normalmente, se denomina esqueletonización, ya que el objetivo es extraer el esqueleto de las cordilleras. Y, finalmente, con la imagen de los esqueletos de las cordilleras se obtienen las *minutiae*.

La figura 24 muestra el proceso de extracción de las huellas de forma esquemática con una imagen de ejemplo en cada uno de los procesos. La huella dactilar original, binarizada, esqueletonizada y las *minutiae* que se han detectado.

Figura 24. Esquema del proceso de detección de las *minutiae*



6.1. Proceso de binarización

El problema general de binarizar una imagen ha sido ampliamente estudiado en los campos de la visión por computador y reconocimiento de patrones desde los orígenes de estas ciencias en la década de 1970. Normalmente, es una tarea primordial, ya que sirve para seleccionar qué píxeles contienen el objeto que se debe considerar (en la imagen binaria tienen el valor blanco) y qué píxeles son fondos (en la imagen binaria tienen el valor negro). Una vez se ha obtenido la imagen binaria, es trivial deducir el tamaño, altura, anchura o posición del objeto que se va a considerar.

En estos apuntes, consideramos que un píxel de una imagen binaria solo puede tomar dos valores: $B[x, y] = \{0, 255\}$. El color es negro cuando $B[x, y] = 0$ y el color es blanco cuando $B[x, y] = 255$.

Los métodos más sencillos trabajan con un único umbral para toda la imagen. Si el valor del píxel es superior a este umbral, entonces sitúan el nuevo valor a 255. De lo contrario, lo sitúan a 0. Algunas veces, podemos caracterizar diferentes zonas de la imagen (una malla de bloques cuadrados) con diferentes

Ved también

El proceso de la imagen de la huella dactilar mejorada se ha explicado en el apartado 5 de este módulo.

Reflexión

En lo que queda de apartado, vamos a explicar estos procesos, excepto el primero, que ya se ha tratado en el apartado anterior.

umbrales y considerar los niveles de iluminación. Normalmente, las zonas son rectangulares y se detectan unas 16 o 32. En estos casos, se puede aplicar un umbral local en cada zona y adaptar automáticamente el umbral según la intensidad media de la zona de la imagen. En el caso de las huellas y debido a que las imágenes originales (o incluso las que se han mejorado) tienen una calidad muy baja, es difícil usar estos métodos generales de binarización y se han deducido otros específicos. Se han presentado bastantes métodos de binarización de huellas dactilares, así que en este documento vamos a presentar uno de los más usados.

Ratha, Chen y Jain, en 1995, introdujeron un método de binarización basado en la matriz de orientación filtrada. El proceso de binarización se lleva a cabo para cada píxel siguiendo los siguientes pasos:

- 1) Se genera una ventana de 15×15 orientada según la matriz de orientación filtrada y centrada en el píxel que queremos tratar. Atención en no salirse de la matriz, en los casos de que alguna parte de la ventana quedara fuera de la matriz, se reducen las dimensiones de la ventana.
- 2) Se calcula la firma media con las 15 firmas ortogonales en la matriz de orientación.
- 3) Se suaviza la firma media. Cada píxel de la firma media suavizada se calcula como la media de él mismo y de sus dos vecinos en la firma media (atención con los píxeles extremos).
- 4) Con la firma media, se calcula un único valor que es la media de los 15 valores obtenidos de la firma media. Este nuevo valor es el umbral local para binarizar el píxel central de la firma.

Con el objetivo de mejorar la calidad de la imagen binarizada, algunos sistemas detectan pequeñas manchas negras o blancas en la imagen binaria que se considera que deben ser ruido, puesto que las cordilleras o valles nunca son tan pequeños. Para eliminar estas manchas, se puede usar el método llamado coloreado de manchas¹.

Dada una imagen binaria, el método de coloreado de manchas para la limpieza de pequeñas áreas es un proceso formado por tres etapas:

- 1) **Segmentación de la imagen binaria en regiones:** Se obtiene una imagen nueva donde cada mancha tiene un nivel de gris diferente. A la hora de asignar el nivel de gris para cada región no se usa el 0 ni el 255 puesto que ya están inicialmente asignados a la imagen binaria. Si se usa un formato de imagen estándar, solo tenemos $256 - 2 = 254$ posibles manchas diferentes. A veces, este valor es demasiado bajo. En este caso, se tiene que trabajar con un formato

Reflexión

Es importante considerar algunos trucos para optimizar el proceso, en caso contrario este algoritmo es muy lento. Por ejemplo, si el ángulo en un píxel es muy parecido al ángulo del píxel vecino, algunos cálculos se pueden evitar.

Ved también

Se ha explicado qué es una firma y cómo se obtienen los píxeles que la forman en el subapartado 3.3 del presente módulo.

⁽¹⁾En inglés, *blob colouring*.

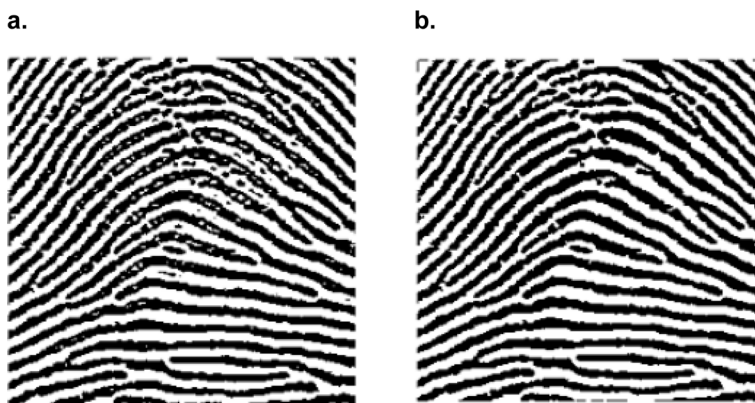
de imagen que permita tener más profundidad de píxel. Es decir, los píxeles, en lugar de estar definidos por un byte, tienen que estar definidos por 2 bytes, entonces podemos llegar a tener $2^{16} - 2 = 65.534$ manchas diferentes.

2) **Detectar las áreas de las manchas:** Se obtiene un vector donde cada posición representa una mancha. La mancha que se ha coloreado con el valor X en el proceso anterior queda representada por la posición X del vector. El área de esta mancha se almacena en $\text{Vector}[X]$. Después de la segmentación (etapa anterior), no puede haber manchas blancas ni negras puesto que todas se tienen que haber reasignado. Por eso, $\text{Vector}[0] = 0$ y $\text{Vector}[255] = 0$.

3) **Las manchas con área superior a un umbral ($\text{Vector}[X] > \text{Umbral}$) se borran:** Para hacerlo, tomaremos la imagen original binaria y la volveremos a pintar. Esto quiere decir que si son manchas negras (blancas) están rodeadas de una sola mancha blanca (negra) y, por lo tanto, se tienen que pintar de color blanco (negro).

En la figura 25, se muestra una imagen binaria original y la imagen limpiada. Se aprecia la eliminación de las pequeñas manchas y agujeros. Además de este proceso de limpieza de la imagen binaria, también se puede llevar a cabo un proceso más inteligente donde se borran los pequeños salientes de las cordilleras o se unen cordilleras que se considera que han sido rotas por cortes o suciedad. Estas técnicas pueden usar la imagen de orientación.

Figura 25. Imagen binaria original y limpia



a. Detalle de una huella dactilar que se ha binarizado; b. imagen después de la binarización con limpieza de manchas

6.2. Proceso de eskeletonización

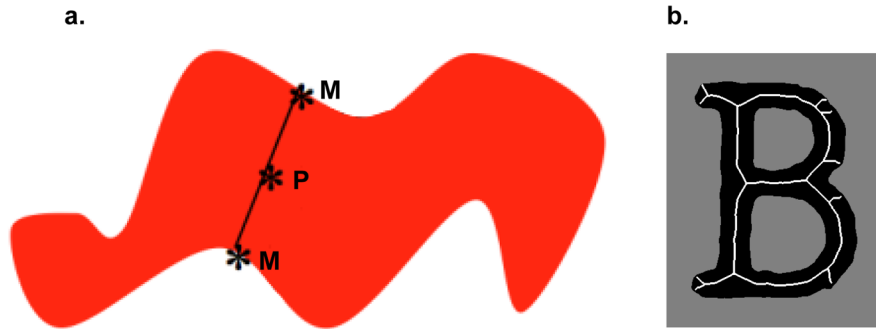
Un píxel de una imagen eskeletonizada solo puede tomar dos valores $E[x, y] = \{0, 255\}$. El color negro, $E[x, y] = 0$, representa un píxel de la cordillera y el color blanco, $E[x, y] = 255$, representa un píxel de un valle.

El proceso de generar el esqueleto también se denomina *transformación del eje medio*². Un píxel forma parte del esqueleto si pasa lo siguiente: se calcula la distancia de él a todos los píxeles del contorno de la figura. En nuestro caso, se calcula la distancia con los píxeles del lado de la cordillera. Si hay más

⁽²⁾En inglés, *medial axis transform*.

de un píxel contorno con la misma distancia, entonces quiere decir que está exactamente en medio de la cordillera y es el eje medio. La figura 26 muestra a la izquierda una figura con un píxel p que forma el esqueleto puesto que tiene dos píxeles contorno M a la misma distancia. A la derecha, se muestra el esqueleto de la letra B.

Figura 26.



a. Ejemplo de un píxel p que forma parte del esqueleto; b. esqueleto de una figura

Gonzalez y Woods, en el 2001, presentaron un algoritmo para generar el esqueleto de un objeto dada una imagen binarizada. Definimos el vecindario del píxel $p = (x, y)$ de la manera siguiente:

$$p_0 = (x, y + 1), p_1 = (x + 1, y + 1), p_2 = (x + 1, y), p_3 = (x + 1, y - 1), p_4 = (x, y - 1), p_5 = (x - 1, y - 1), p_6 = (x - 1, y), p_7 = (x - 1, y + 1)$$

p_3	p_2	p_1
p_4	p_3	p_0
p_5	p_6	p_7

Los comentarios y explicaciones de este algoritmo se centran en el caso específico que queremos encontrar, el esqueleto de las cordilleras. Antes de nada, debemos establecer qué píxeles se consideran frontera de la cordillera:

- p pertenece a la frontera izquierda si p_4 es valle y p es cordillera
- p pertenece a la frontera derecha si p_0 es valle y p es cordillera
- p pertenece a la frontera superior si p_2 es valle y p es cordillera
- p pertenece a la frontera inferior si p_6 es valle y p es cordillera

En este algoritmo, usaremos expresiones lógicas. Por eso, tenemos que definir el valor lógico $n_i = \text{CIERTO}$ si y solo si p_i es cordillera, para $i = 0 \dots 7$. Con estas expresiones lógicas sencillas, podemos definir las siguientes expresiones más complejas que se usan en el algoritmo:

$$\begin{aligned}
 B_0 &= n_4 \cdot (n_2 + n_3 + n_5 + n_6) \cdot (n_6 + \bar{n}_7) \cdot (\bar{n}_1 + n_2) \\
 B_2 &= n_6 \cdot (n_0 + n_4 + n_5 + n_7) \cdot (n_0 + \bar{n}_1) \cdot (\bar{n}_3 + n_4) \\
 B_4 &= n_0 \cdot (n_1 + n_2 + n_6 + n_7) \cdot (n_2 + \bar{n}_3) \cdot (\bar{n}_5 + n_6) \\
 B_6 &= n_2 \cdot (n_0 + n_1 + n_3 + n_4) \cdot (n_4 + \bar{n}_5) \cdot (n_0 + \bar{n}_7)
 \end{aligned}
 \tag{30}$$

No explicaremos en detalle cómo se deducen estas expresiones y pasaremos a definir directamente el algoritmo. Estas expresiones son combinaciones de situaciones que nos podemos encontrar y que son potenciales píxeles que pasarán de ser cordillera a ser valle para así ir adelgazando sucesivamente las cordilleras y quedarnos solo con el esqueleto. Durante el algoritmo, se van seleccionando píxeles de la imagen binaria para después tratarlos. Inicialmente, no hay ningún píxel seleccionado. La manera más sencilla de implementar el método de selección de píxeles es a través de una imagen binaria auxiliar.

algoritmo d' esqueletonización de González y Woods

repetir

```

para todo píxel p en la imagen binaria E[x,y]
  si p pertenece a la frontera izquierda y B4
    entonces selecciona p
fin para todo
borra los píxeles que han sido seleccionados.

```

```

para todo píxel p en la imagen
  si p pertenece a la frontera derecha y B0
    entonces selecciona p
fin para todo
borra los píxeles que han sido seleccionados.

```

```

para todo píxel p en la imagen
  si p pertenece a la frontera superior y B2
    entonces selecciona p
fin para todo
borra los píxeles que han sido seleccionados.

```

```

para todo píxel p en la imagen
  si p pertenece a la frontera inferior y B6
    entonces selecciona p
fin para todo
borra los píxeles que han sido seleccionados.

```

hasta que no haya cambios

fin algoritmo

6.3. Proceso de extracción de las *minutiae*

Una vez se ha extraído el esqueleto de la imagen binaria, tenemos que detectar la posición de las huellas digitales, el tipo de huella (terminal o bifurcación) así como su orientación. Uno de los métodos más sencillos se denomina *cruce de números*³.

⁽³⁾En inglés, *crossing number*.

Arcelli y Baja, en 1984, propusieron el método de cruce de números.

Dado un píxel de la imagen eskeletonizada $\mathbf{p} = (x, y)$ y su valor $E[x, y] = E[\mathbf{p}]$, el **cruce de números** $cn(\mathbf{p})$ se define como la mitad de la suma de las diferencias entre pares de píxeles adyacentes en los ocho vecinos de este píxel:

$$cn(p) = \frac{1}{2} \sum_{i=1..8} |E(p_{i \bmod 8}) - E(p_{i-1})| \quad (31)$$

Donde $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_7$ son los píxeles vecinos a \mathbf{p} que pertenecen a una secuencia específica y que definen los ocho vecinos de \mathbf{p} . Podría ser la secuencia del apartado anterior, por ejemplo:

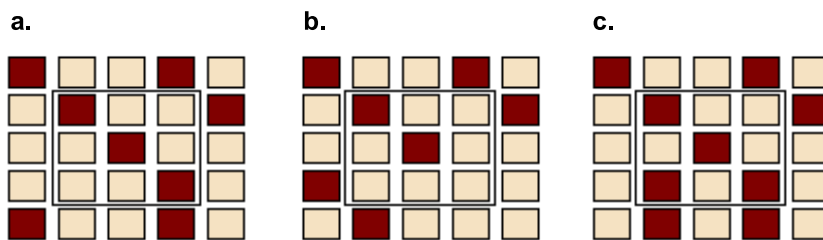
$$\mathbf{p}_0 = (x - 1, y - 1), \mathbf{p}_1 = (x - 1, y), \mathbf{p}_2 = (x - 1, y + 1), \mathbf{p}_3 = (x, y + 1), \mathbf{p}_4 = (x + 1, y + 1), \mathbf{p}_5 = (x + 1, y), \mathbf{p}_6 = (x + 1, y - 1), \mathbf{p}_7 = (x, y - 1)$$

Como en estos apuntes siempre consideramos que los píxeles de la imagen binaria y eskeletonizada solo pueden tomar los dos valores $\{0, 255\}$, entonces es sencillo que nos demos cuenta de que un píxel (x, y) que forma parte de una cresta (color negro), es decir que tiene valor $E[x, y] = 0$ es:

- Un punto aislado si $cn(\mathbf{p}) = \frac{1}{2} \cdot 0 \cdot 255 = 0$.
- Una *minutia* de tipo terminal si $cn(\mathbf{p}) = \frac{1}{2} \cdot 2 \cdot 255 = 255$.
- Un punto en medio de una cresta si $cn(\mathbf{p}) = \frac{1}{2} \cdot 4 \cdot 255 = 510$.
- Una *minutia* de tipo bifurcación si $cn(\mathbf{p}) = \frac{1}{2} \cdot 6 \cdot 255 = 765$.
- Un cruce de cordilleras si $cn(\mathbf{p}) = \frac{1}{2} \cdot 8 \cdot 255 = 1.020$.

Es matemáticamente imposible que el número en negrita de los casos antes expuestos tome un valor impar. La figura 27 muestra un punto intermedio: $cn(\mathbf{p}) = 510$, una *minutia* terminal: $cn(\mathbf{p}) = 255$ y una *minutia* bifurcación: $cn(\mathbf{p}) = 765$.

Figura 27. Tres ejemplos de píxeles que componen una cordillera:



a. un punto intermedio; b. una *minutia* terminal; c. una *minutia* bifurcación

Ahora, debemos detectar la dirección de las *minutiae* ubicada en la posición (x, y) que describiremos por α_{xy} . Como ya se ha indicado en apartados anteriores, esta dirección es de 360 grados (los cuatro cuadrantes) y se calcula a través de la imagen de orientación. Debido a que la imagen de orientación solo tiene la información de dos cuadrantes, tendremos que hacer la conversión de dos cuadrantes $]-\pi/2, \pi/2]$ a cuatro cuadrantes $]-\pi, \pi]$.

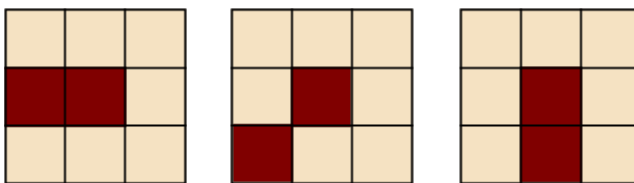
Ved también

La figura 9 del subapartado 2.2 muestra la orientación de los ángulos.

1) En los casos en los que tengamos una *minutia* de tipo terminal, $cn(\mathbf{p}) = 255$:

a) Si $\theta_{xy} \in [0, \pi/2]$ y nos encontramos en una de las situaciones que presenta la figura 28:

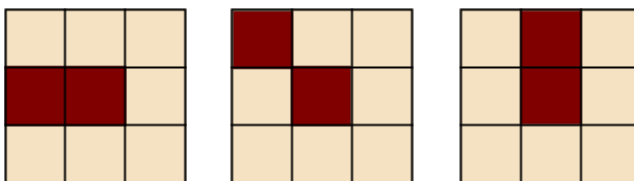
Figura 28



entonces: $\alpha_{xy} = \theta_{xy} - \pi$.

b) Si $\theta_{xy} \in]0, -\pi/2]$ nos encontramos en una de las situaciones que presenta la figura 29:

Figura 29



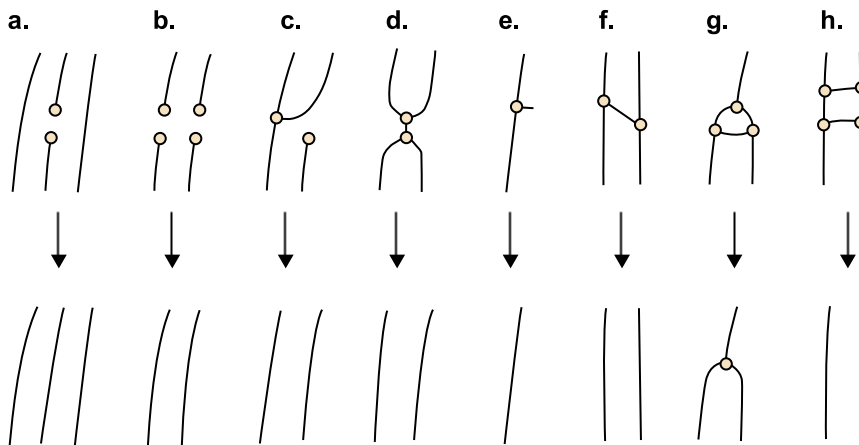
entonces: $\alpha_{xy} = \theta_{xy} + \pi$.

c) En el resto de los casos: $\alpha_{xy} = \theta_{xy} - \pi$.

2) En los casos en los que tengamos una *minutia* de tipo bifurcación, $cn(\mathbf{p}) = 765$, también se podría encontrar el ángulo de la *minutia* haciendo una casuística parecida. No obstante, el número de combinaciones es muy superior. Por este motivo, varios investigadores optan por una solución diferente. Partiendo de la imagen binarizada, entonces se invierte la imagen. De este modo, las crestas quedan representadas en color blanco y los valles, en color negro. Con esta imagen invertida, se aplica todo el mismo proceso mencionado antes hasta llegar a la definición de las *minutiae* terminal y su ángulo. Las *minutiae* terminal de la imagen invertida son las *minutiae* bifurcación de la imagen binaria original.

Finalmente, es interesante mencionar que, con el algoritmo que hemos presentado, se suelen detectar bastantes *minutiae* falsas. Uno de los métodos más usados para la eliminación de falsos positivos (es decir, borrar *minutiae* falsas) es el algoritmo de **Xiao** y **Raafat** presentado en 1991. El algoritmo, usando la información de la imagen y el conjunto de *minutiae* encontrado, deduce las combinaciones o estructuras que tienen que ser eliminadas si se encuentran a una distancia inferior a un umbral. La figura 30 muestra estas combinaciones.

Figura 30.



a y b. Conecta *minutiae* terminales encaradas según sus ángulos; c y d. borra bifurcaciones encaradas según sus ángulos; e. borra pequeñas cordilleras con distancia inferior a un umbral; f. borra puentes entre cordilleras y considerando el ángulo de la imagen de orientación; g. borra triángulos y deja solo una *minutiae* bifurcación dependiendo de una distancia mínima; h. borra estructuras de tipo escalera (generalización del caso f)

7. Comparación de huellas dactilares

En las librerías informáticas, para desarrollar aplicaciones con dispositivos biométricos, existen las rutinas de comparación de huellas dactilares. Estas, dadas un par de imágenes de huellas, pueden devolver un resultado continuo que normalmente está limitado de 0 a 1 o pueden devolver directamente una decisión binaria: parecidas/diferentes. El resultado continuo entre 0 y 1 puede representar dos cosas. A veces, representa una probabilidad, entonces, 1 significa que las dos huellas son completamente iguales y 0 completamente diferentes. Pero a veces representa una distancia, entonces, el valor 0 indica que son completamente iguales y 1 completamente diferentes. Todos los algoritmos básicos de comparación de huellas devuelven unos valores continuos, por lo tanto, en los casos en los que la salida de la rutina es binaria, es porque hay un umbral modificable por el supervisor del sistema. Si la salida continua del algoritmo es superior (inferior) al umbral de la probabilidad (distancia), entonces se decide que las huellas son iguales. De lo contrario, se decide que son diferentes.

La mayoría de los algoritmos de comparación de huellas dactilares operan con la información del conjunto de *minutiae* extraídas de las huellas para comparar pero también se han presentado algunos algoritmos que trabajan directamente con la imagen. La calidad de la comparación de estos últimos siempre ha sido menor pero tienen la ventaja de que no hacen falta todos los procesos de obtención de la imagen binaria y extracción de las *minutiae* antes comentados.

A partir de este punto, denominaremos T a la representación de la huella dactilar que se ha usado en el proceso de matriculación del usuario (proviene del inglés *template*). Y denominamos I (proviene del inglés *input*) a la representación de la nueva huella dactilar que deseamos comparar. En el caso de los algoritmos que comparan directamente las huellas, las representaciones de estas son exactamente las imágenes de las huellas I y T. En los otros algoritmos, normalmente es un conjunto de *minutiae*. A lo largo de este apartado, denominamos tanto a la huella como a su representación por T y por I.

Los procesos de extracción de las características de las huellas dactilares y de comparación son normalmente muy similares tanto si nos encontramos llevando a cabo una identificación como una verificación. Esto se debe a que la identificación se puede llevar a cabo ejecutando un conjunto de comparaciones una a una entre parejas de representaciones de huellas. Los procesos de clasificación de huellas se usan normalmente para reducir el tiempo de busca de una huella en una base de datos grande en un proceso de identificación.

Ved también

Los sistemas de identificación y verificación se han explicado en el módulo “La biometría para la identificación de las personas” y los procesos de clasificación se explicarán en el siguiente apartado, “Clasificación de las huellas dactilares”, de esta asignatura.

El proceso de comparación de dos huellas dactilares es un problema muy difícil que lleva muchos años en continua investigación y desarrollo. Esto se debe a la enorme diferencia que puede existir entre dos huellas dactilares del mismo dedo y puestas en el sensor en un intervalo muy corto de tiempo. Se dice que las huellas tienen una gran **variabilidad** dentro de la clase. Es decir, que la impresión de la misma huella puede variar mucho. Los principales factores son:

- **Desplazamiento:** El mismo dedo se puede poner en diferentes posiciones en el área de lectura del sensor en varias adquisiciones. Este hecho genera un desplazamiento de la región donde está el dedo en la imagen. Por ejemplo, con solo desplazar el dedo 2 milímetros (imperceptible para el usuario) se genera un desplazamiento de 40 píxeles en un escáner de dedo usual con una resolución de 500 dpi.
- **Rotación:** De manera similar al desplazamiento, el mismo dedo se puede poner en diferentes ángulos respecto al eje vertical del sensor. Normalmente, los algoritmos toleran una rotación hasta 20 grados hacia la derecha o hacia la izquierda. Si la rotación es superior, entonces los algoritmos no son capaces de darse cuenta de que es el mismo dedo.
- **Solapamiento parcial:** El desplazamiento y la rotación del dedo causan que parte de la huella dactilar caiga fuera del área del sensor. Este hecho provoca que la parte de la huella que aparece en las dos imágenes sea menor de lo normal.
- **Distorsión no lineal:** Debido a la plasticidad del dedo, que la superficie del dedo es tridimensional y que el sensor es bidimensional, la huella es una imagen bidimensional que muestra una superficie tridimensional, pero con distorsiones difíciles de controlar. Estas distorsiones pueden ser del tipo **compresión de la imagen** (normalmente en las partes centrales de la huella) o **extensión** (normalmente en las partes externas de la huella) y dependen de la presión que se ha aplicado con el dedo hacia el sensor.
- **Condiciones de la piel:** Las cordilleras y valles son captadas por el sensor si están tocando la superficie del sensor de forma uniforme. No obstante, la presión, la suciedad de la piel, las enfermedades de la piel, el sudor o cortes hacen que la imagen cambie en sucesivas adquisiciones del dedo.
- **Ruido:** El propio sensor para adquirir la imagen genera ruido y distorsiona la imagen. Un ejemplo habitual es la grasa que queda del dedo en el área de cristal del sensor. Dependiendo de la pulcritud de ese cristal, la imagen será cambiante. Otra fuente de distorsión puede ser la comunicación del

sensor con el ordenador o sistema usado para comparar las imágenes. En función del protocolo de comunicación o de la largura del cable o de la red, se pueden generar errores de comunicación.

- **Errores en la extracción de características:** Los propios algoritmos de extracción de características introducen errores. Estos errores pueden aparecer en cualquiera de los procesos internos del algoritmo. Por ejemplo, tenemos que hacer aproximaciones que inducen a errores cuando estimamos la orientación o cuando estimamos la posición del dedo.

La figura 31 muestra unas imágenes del mismo dedo que no pudieron ser reconocidas por la mayoría de los algoritmos presentados al concurso Fingerprint Verification Competition del año 2002.

Figura 31. Dos huellas de un mismo dedo pero con características muy diferentes



Por otro lado, basándose en datos grandes, nos podemos encontrar con dos dedos diferentes que tienen una estructura muy parecida (posición de la singularidad e imagen de orientación). No obstante, se ha visto que la probabilidad de que las *minutiae* caigan en posiciones similares es muy baja. En la figura 32, se muestran dos huellas de dedos diferentes y con una estructura muy similar. Pero si nos fijamos atentamente, veremos que las *minutiae* están ubicadas en posiciones diferentes.

Figura 32. Dos huellas de dos dedos diferentes pero con características parecidas



Muchos algoritmos de comparación de dedos y con metodologías muy diversas se han presentado durante los últimos veinte años. La mayoría de estos algoritmos no tienen problemas a la hora de comparar huellas de buena calidad. No obstante, la comparación de las huellas de baja calidad sigue siendo un reto hoy en día. En los casos de los AFIS supervisados por un humano, se suele aplicar un algoritmo de detección de la calidad de la huella. Las huellas dactilares que no superan el nivel de calidad no se introducen en el sistema en el proceso de matrícula.

Los métodos para comparar un par de huellas dactilares se pueden clasificar en tres grandes familias:

1) **Comparación basada en la correlación:** Las imágenes de las huellas se ubican una sobre la otra y la correlación entre los píxeles se calcula aplicando diferentes alineamientos como, por ejemplo, diferentes rotaciones y traslaciones.

2) **Comparación basada en las *minutiae*:** Es la técnica más usada con diferencia. Además, es la base de la metodología que usan los expertos para comparar huellas de forma manual. La huella dactilar queda representada por un conjunto de *minutiae* y la comparación se basa en alinear el conjunto de *minutiae* generado por la huella *template* con el conjunto de *minutiae* generado por la huella *input*.

3) **Comparación basada en otras características que no son las *minutiae*:** Extraer las *minutiae* es una tarea difícil en huellas de muy baja calidad. No obstante, hay otras características que son más fáciles de extraer, aunque la imagen sea muy ruidosa. Por ejemplo, la información de la orientación local, la frecuencia de las cordilleras en un punto, la forma de las cordilleras o la información de la textura. La comparación también se basa en alinear estas características locales.

A continuación, nos vamos a adentrar en explicar las dos primeras familias de métodos de comparación.

7.1. Técnicas basadas en la correlación de imágenes

T' e I' son dos imágenes **vectorizadas** de huellas dactilares correspondientes a la imagen registrada T y a la imagen I nueva que se quiere identificar.

Una **imagen vectorizada** es aquella cuya matriz se ha convertido en un vector simplemente concatenando las columnas de la imagen.

Una medida intuitiva de la diversidad de estas imágenes es la suma de las diferencias al cuadrado, SSD, entre las intensidades de los píxeles:

$$SSD = \|T' - I'\|^2 = (T' - I')^t (T' - I') = \|T'\|^2 + \|I'\|^2 - 2T'^t I' \quad (32)$$

donde el superíndice t denota la matriz transpuesta. Si los términos T'^2 y I'^2 son constantes e independientes de la correlación entre imágenes, la diversidad entre las dos imágenes vectorizadas se minimiza cuando la relación cruzada, $CC(T', I') = T'^t I'$, entre T' e I' se maximiza.

Debido a que el término $T'^t I'$ aparece en forma negativa a la función de la diversidad, entonces podemos decir que $CC(T', I')$ es una función que representa la similitud entre imágenes. No obstante, debido al posible desplazamiento y rotación que aparece inevitablemente al colocar el dedo en el sensor, la similitud entre las imágenes no se puede calcular directamente superponiendo ambas imágenes y aplicando la función CC .

Si $I^{(\Delta_x, \Delta_y, \theta)}$ es la imagen I donde se ha aplicado una rotación con ángulo θ con origen en el centro de la imagen y un desplazamiento Δ_x y Δ_y píxeles en las direcciones x e y , respectivamente. Si $I^{(\Delta_x, \Delta_y, \theta)}$ es la imagen vectorizada de $I^{(\Delta_x, \Delta_y, \theta)}$. Entonces, la similitud entre las dos imágenes se puede medir de la manera siguiente:

$$S(T, I) = \max_{\Delta_x, \Delta_y, \theta} CC(T', I^{(\Delta_x, \Delta_y, \theta)}) \quad (33)$$

La aplicación directa de esta ecuación pocas veces aporta resultados aceptables debido a los problemas de variabilidad que ya hemos comentado. Esta ecuación solo tiene en cuenta la rotación y el desplazamiento de los dedos pero no considera todo el resto de los factores de distorsión o variabilidad. Además,

Reflexión

La tercera familia de métodos de comparación no se va a explicar en este material.

Reflexión

Los algoritmos para encontrar soluciones aproximadas no se explican en este material por el poco uso que se tiene de esta técnica.

el cómputo directo de esta ecuación es muy costoso. Se han definido algunos algoritmos para encontrar soluciones aproximadas. Una solución sencilla sería adaptar el algoritmo de la transformada generalizada de Hough.

7.2. Técnicas basadas en la comparación de *minutiae*

Tal como ya se ha indicado, la técnica de comparación de huellas dactilares basada en la comparación de *minutiae* es la más extendida. El principal motivo es la analogía estricta con la manera como los especialistas forenses comparan los dedos y la aceptación en tribunales como prueba de identidad de personas en la mayoría de los países del mundo. Si la metodología del sistema automático es la misma que la del sistema manual, entonces, los especialistas forenses pueden defender las pruebas de una manera más convincente.

Si T e I son la representación de la huella *template* y la huella *input*, asumiremos que esta representación es un conjunto de *minutiae* del tipo:

$$T = \{m_1, m_2, \dots, m_m\}, m_i = \{x_p, y_p, \theta_p, t_i\}, i = 1 \dots m$$

$$I = \{m'_1, m'_2, \dots, m'_n\}, m'_j = \{x'_j, y'_j, \theta'_j, t'_j\}, j = 1 \dots n$$

donde m y n denotan el número de *minutiae* en los conjuntos T e I , respectivamente. La posición y la orientación de las *minutiae* en la imagen son x_p, y_p, θ_p y el tipo de *minutiae* es $t_i \in \{Terminal, Bifurcación\}$.

Consideramos que las *minutiae* m_i y m'_j se emparejan si son del mismo tipo, la distancia espacial, sd , entre ellas es inferior a un umbral de tolerancia r_0 y la diferencia de direcciones, dd , entre ellas es también inferior a un umbral de tolerancia θ_0 . Estos umbrales son necesarios para compensar los errores inevitables que surgen en los algoritmos de extracción de características y las distorsiones debidas a la huella en sí (plasticidad, suciedad). Formalmente, dos *minutiae* se emparejan si se cumplen las tres condiciones siguientes:

$$t_i = t'_j \quad (34)$$

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 \quad (35)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 2\pi - |\theta'_j - \theta_i|) \leq \theta_0 \quad (36)$$

La definición de la distancia de direcciones toma el mínimo de $|\theta'_j - \theta_i|$ y de $2\pi - |\theta'_j - \theta_i|$ para considerar la circularidad de los ángulos. Por ejemplo, la diferencia entre 5° y 350° no es 345° sino $360^\circ - (350^\circ - 5^\circ) = 15^\circ$.

Alinear las dos huellas dactilares es un paso obligado para maximizar el número de *minutiae* emparejadas. El alineamiento correcto de dos huellas requiere que podamos recuperar la información del desplazamiento en las x y las y así como la rotación θ que hemos llevado a cabo cuando se ha ubicado el dedo en el sensor. Sea $map()$ una función que transforma una *minutiam'* $_j$ de **I** con una *minutiam''* $_j$ según las transformaciones de desplazamiento $[\Delta_x, \Delta_y]$ y rotaciones θ . Usualmente, el centro de la rotación se considera la posición media de las *minutiae*. Definimos $map()$ formalmente como:

$$map_{\Delta_x, \Delta_y, \theta}(m'_j) = m''_j \quad (37)$$

donde:

$$m''_j = \{x''_j, y''_j, \theta''_j, t''_j\} \quad (38)$$

La nueva posición y el ángulo de rotación se calculan como:

$$\theta''_j = \theta_j + \theta \quad (39)$$

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} \quad (40)$$

Y, obviamente, el tipo de *minutiae* no se ve afectado por estos tipos de transformaciones, $t''_j = t'_j$.

Por otro lado, definimos $mm()$ como una sencilla función que devuelve 1 si dos *minutiae* se emparejan, es decir, si son del mismo tipo y su diferencia es inferior a los umbrales r_0 y θ_0 . Formalmente:

$$mm(m''_j, m_i) = \begin{cases} 1 & \text{si } t''_j = t_i \text{ i } sd(m''_j, m_i) \leq r_0 \text{ i } dd(m''_j, m_i) \leq \theta_0 \\ 0 & \text{otros} \end{cases} \quad (41)$$

Entonces, el problema de etiquetar las *minutiae* de dos huellas dactilares se puede formular como un problema de encontrar un punto máximo. Formalmente:

$$\max_{\Delta_x, \Delta_y, \theta, P} \sum_{i=1}^m mm(map_{\Delta_x, \Delta_y, \theta}(m'_{P(i)}), m_i) \quad (42)$$

donde $P(i) = j$ puede ser cualquier función que determina el etiquetado entre *minutiae* de una de las huellas con *minutiae* de la otra huella. Una *minutia* tiene exactamente una pareja en la otra huella o no tiene ninguna.

- $P(i) = j$: Indica que la pareja de m_i de **T** es la *minutia* m'_j de **I**.

- $P(i) = Nul$: Indica que la *minutia* m_i de T no tiene pareja.
- Una *minutia* m'_j de I no tiene pareja en T si $P(i) \neq j, \forall i = 1 \dots m$.
- Cada *minutia* de I se asocia como mucho a una *minutia* de T, es decir, P es una función inyectiva dentro del conjunto de las huellas. O se asocia a una sola *minutia* o se asocia al *Nul*. La expresión matemática es:

$$\forall i = 1 \dots m, k = 1 \dots m, i \neq j \Rightarrow P(i) \neq P(k) \text{ o } P(i) = P(k) = Nul \quad (43)$$

Es importante destacar que, cuando se asocian dos *minutiae*, $P(i) = j$, no significa que las *minutiae* m'_j de I y m_i de T sean exactamente iguales después de la transformación debida a $\Delta_x, \Delta_y, \theta$, sino que quiere decir que es probable que sean la misma *minutia*. También se tiene que destacar que el problema de etiquetar dos conjuntos de *minutiae* se ha convertido en un problema de encontrar un máximo en una función. Este máximo depende de dos factores, el primero es la transformación global de una huella respecto a la otra $\Delta_x, \Delta_y, \theta$ y el segundo es la asociación de *minutiae* P . Pero a la hora de encontrar la solución óptima, se tienen que considerar estos dos factores a la vez. En el campo de la visión por computador, este es el problema de emparejar imágenes (*image registration*) y es muy complejo. Se han presentado varios algoritmos y todavía se presentan otros nuevos.

Nosotros os mostramos una solución bastante sencilla al problema. La sencillez del algoritmo se basa en suponer que primero se puede calcular la transformación de la huella (es decir, los parámetros $\Delta_x, \Delta_y, \theta$) y después, dados estos parámetros, se puede calcular la asociación de huellas P . El método aporta una solución subóptima, que quiere decir que quizás hay una solución más buena pero no la hemos sabido encontrar. Por lo tanto, uno de los parámetros de transformación puede tener un valor un poco diferente o una asociación de dos *minutiae* puede ser diferente.

Ratha, Rover y Jain, en 1996, propusieron el algoritmo siguiente para encontrar la transformación de una huella dactilar. Antes de nada, discretizaron el espacio de transformación:

$$\Delta_x^+ \in \{\Delta_{x_1}^+, \Delta_{x_2}^+, \dots, \Delta_{x_a}^+\} \quad (44)$$

$$\Delta_y^+ \in \{\Delta_{y_1}^+, \Delta_{y_2}^+, \dots, \Delta_{y_b}^+\} \quad (45)$$

$$\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_b^+\} \quad (46)$$

El número de elementos discretizados en el desplazamiento, a , y en el ángulo, b , depende de la aplicación. Cuanto mayores son, más lento es el algoritmo. Además, si se discretiza en muchos valores, nos encontramos que todas las muestras caen casi siempre en valores discretos diferentes y entonces el algo-

ritmo no funciona. Con esta discretización, se define una matriz tridimensional A donde cada eje es uno de los tres parámetros de transformación y donde cada celda es una posible combinación de estos valores de forma discreta:

$$A[\Delta_x^+, \Delta_y^+, \theta^+] \quad (47)$$

A continuación, se muestra el algoritmo.

algoritmo de Ratha, Rover y Jain

$A[\Delta_x^+, \Delta_y^+, \theta^+] = 0$ para todo valor de Δ_x^+ , Δ_y^+ y de θ^+

```

para toda minutia  $m_i$  de  $T$ 
  para toda minutia  $m'_j$  de  $I$ 
    si  $t'_j = t_i$  entonces
      para todo valor de  $\theta^+$ 
         $\theta_{aux} = \theta'_j + \theta^+$ 
        si  $dd(m_{aux}, m_i) \leq \theta_0$ 
          
$$\begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \cos \theta^+ & -\sin \theta^+ \\ \sin \theta^+ & \cos \theta^+ \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix}$$

           $\Delta_x^+ =$  discretización de  $\Delta_x$ 
            en el valor discreto más cercano
           $\Delta_y^+ =$  discretización de  $\Delta_y$ 
            en el valor discreto más cercano
          si  $\Delta_x^+$  i  $\Delta_y^+$  están dentro del dominio
            entonces  $A[\Delta_x^+, \Delta_y^+, \theta^+] = A[\Delta_x^+, \Delta_y^+, \theta^+] + 1$ 
          fin si
        fin si
      fin si
    fin si
  fin algoritmo

```

Fijaos en que este algoritmo usa solo el umbral θ_0 y no hace falta usar el umbral r_0 . Una vez se ha acabado este proceso de acumular candidatos en la matriz A , entonces se decide que los valores resultantes de la transformación entre huellas Δ_x , Δ_y , θ son los que han aparecido más veces. Es decir, la posición de la celda de A cuyo contenido es mayor:

$$\{\Delta_x, \Delta_y, \theta\} = \max_{\{\Delta_x^+, \Delta_y^+, \theta^+\}} A[\Delta_x^+, \Delta_y^+, \theta^+] \quad (48)$$

Podrían aparecer varias celdas con el máximo valor; en estos casos, se elige una de las celdas al azar.

Una vez tenemos los valores de transformación, Δ_x , Δ_y , θ , proponemos un algoritmo muy sencillo para emparejar las *minutiae*:

1) Antes de nada, se tienen que convertir todas las *minutiae* de I con los valores obtenidos:

$$m_j'' = \text{map}_{\Delta_x, \Delta_y, \theta}(m_j') \quad (49)$$

2) Después, calcularemos las distancias de todas las *minutiae* m_j'' con todas las *minutiae* m_i y obtendremos una matriz **D** donde cada celda representa la distancia entre las *minutiae*. Esta distancia se puede calcular como la distancia ponderada de la distancia espacial y la distancia angular:

$$[j, i] = \alpha \cdot sd(m_j'', m_i) + (1 - \alpha) \cdot dd(m_j'', m_i)$$

donde $0 \leq \alpha \leq 1$.

3) Después, iremos recurriendo a la matriz **D** y ordenaremos sus celdas de menor a mayor.

4) Con las celdas de **D** ordenadas, iremos seleccionando las celdas de menor valor a mayor valor. Si el valor menor es $D[j, i]$ entonces:

a) Llevaremos a cabo la asignación $P(i) = j$.

b) Anularemos todas las celdas de la matriz **D** tales que $D[j, k]$ y $D[k, i]$ para todo k . Este paso se lleva a cabo para garantizar que el etiquetado sea biyectivo.

c) Este proceso se acaba cuando ya no quedan más celdas para asignar o el valor de estas supera un umbral. Un modo de decidir este umbral puede ser:

$$sd(m_j'', m_i) \leq r_0 \quad i \quad dd(m_j'', m_i) \leq \theta_0 \quad (50)$$

5) Si han quedado *minutiae* sin asignar puesto que el proceso de asignación ha acabado porque se ha superado el umbral, entonces las *minutiae* restantes se tienen que asignar a *Nul*:

$$P(i) = Nul \quad o \quad P(i)^{-1} = Nul.$$

8. Clasificación de las huellas dactilares

Tal como ya se dijo en el módulo “La biometría para la identificación de las personas”, la identificación de una persona requiere la comparación de su huella con todas las huellas de la base de datos. Esta base de datos podría ser muy grande en algunas aplicaciones forenses o civiles (más de 50 millones de huellas). En estos casos, si no se aplica ninguna técnica de clasificar los dedos, la respuesta temporal del sistema no es admisible. Se puede reducir el tiempo empleado en el proceso de identificación si se reduce el número de comparaciones que se van a llevar a cabo. Algunas veces, se puede usar la información del sexo, la raza, la edad y otros datos relacionados con la persona con la intención de reducir la porción de la base de datos en la que buscar. No obstante, algunas veces esta información no es accesible (por ejemplo, en la identificación de criminales). En estos casos, la huella dactilar de la persona desconocida se compara solo con las huellas que pertenecen a la misma clase de huella que ella.

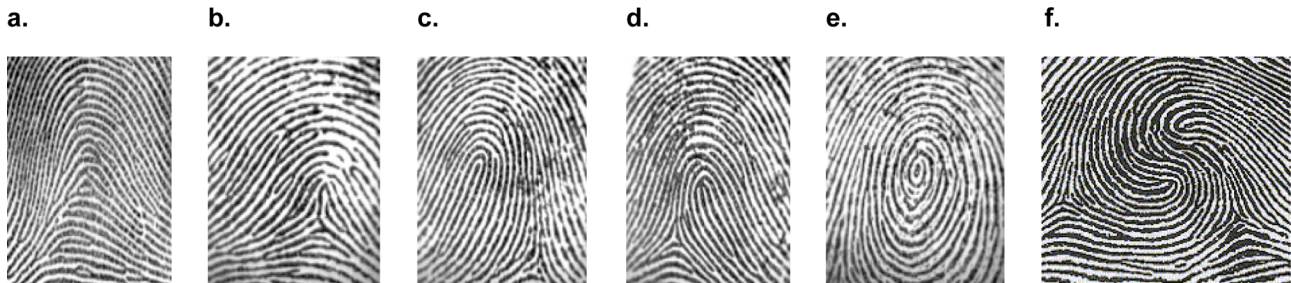
El problema de la clasificación de las huellas dactilares se refiere al problema de asignar a las huellas una clase específica de una **manera consistente** (cada huella va a parar solo a una clase) y segura (que siempre vaya a parar a la misma clase). El método más usado para generar estas clases se basa en las características globales de la huella, es decir, la estructura global de las cordilleras y valles y las singularidades.

La primera clasificación de huellas conocida fue generada por **John Evangelist Purkinje** en 1823. Purkinje era profesor de la Universidad de Breslavia (Polonia) y clasificó los dedos en nueve categorías aunque nunca mencionó su utilidad para la identificación personal. El primer estudio serio y profundo fue llevado a cabo por **Sir Francis Galton** en 1892. Galton era un antropólogo británico y dividió las huellas en tres clases principales (arco, lazo y espiral) y más tarde añadió unas subdivisiones. Más o menos en la misma época, **Juan Vucetich**, que era el jefe de la policía en Buenos Aires, Argentina, diseñó un sistema de clasificación parecida. Este sistema se hizo famoso por ser el primer sistema que se usó para identificar a un asesino en 1892. Unos diez años más tarde, el británico **Sir Edward Henry** redefinió el método Galton y en el año 1900 incrementó el número de clases.

El **sistema Galton-Henry** de clasificación de huellas dactilares se adoptó en muchos países por primera vez y se empezaron a crear las primeras bases de datos.

De hecho, los sistemas actuales se basan todavía en aquella primera clasificación. La figura 33 muestra las seis clases más comunes del esquema Galton-Henry: arco, arco tipo tienda, lazo izquierdo, lazo derecho, espiral y doble lazo.

Figura 33. Clases más comunes del esquema de Galton-Henry



Huellas dactilares: a. tipo arco; b. arco-tienda; c. lazo izquierdo; d. lazo derecho; e. espiral; f. doble lazo

8.1. Descripción formal de las clases de huellas dactilares

La descripción formal de las clases de huellas es la siguiente:

- **Arco:** El dedo tiene cordilleras que entran por un lado, se ensartan hacia un pequeño montículo y salen por el otro lado. No hay puntos singulares de ningún tipo. No hay ni lazos ni deltas.
- **Arco tipo tienda:** Son huellas parecidas al arco pero el montículo es mucho más pronunciado y esto hace que haya una curvatura mucho más pronunciada, casi vertical. Hay dos puntos singulares, hay un lazo y debajo hay un delta.
- **Lazo izquierdo:** Unas cuantas cordilleras entran por el lado izquierdo, dibujan una curva y se vuelven hacia la izquierda. Hay dos puntos singulares, un lazo y un delta. A diferencia del arco tipo tienda, el delta está ubicado debajo del lazo pero un poco a la derecha.
- **Lazo derecho:** Es parecido al lazo izquierdo, pero simétrico al eje vertical.
- **Espiral:** Contiene como mínimo una cordillera que hace un tomo de 360 grados alrededor del centro del dedo. Hay cuatro singularidades. Dos lazos encarados y prácticamente en la misma posición y dos deltas más abajo y uno a cada lado.
- **Doble lazo:** Este tipo de huella es parecida a la espiral pero las singularidades de tipo lazo están separadas. De este modo, ya no encontramos una cordillera que dé una vuelta de 360 grados y hay una cordillera que recorre el dedo de uno de los lazos al otro.

La clasificación de las huellas es una tarea muy difícil debido a la gran variabilidad dentro de las clases y la pequeña variabilidad entre una clase y la otra. Además, las huellas tienen ruido, lo que hace que la clasificación todavía sea más compleja. Se han presentado varios métodos para deducir automáticamente las clases de huellas. Los más conocidos se basan en:

- **Sistema de reglas:** Cada clase de huella se representa por un conjunto de normas. Por ejemplo, para ser un arco tienda tiene que haber un lazo encima de un delta.
- **Sintáctico:** Los patrones se describen a través de una gramática. Es decir, se definen unas reglas de producción y unos puntos terminales. Una huella pertenece a una clase si lo acepta el analizador sintáctico.
- **Estructural:** La huella se define con un grafo. Los nodos del grafo son regiones de la huella donde el ángulo es similar. Los arcos indican que las regiones se están tocando.
- **Estadístico:** De cada huella se genera un vector de características y entonces se aplica un clasificador de propósito general estadístico. El más conocido es el clasificador de Bayes.
- **Red neuronal:** Los elementos de la imagen de orientación se usan como características de entrada de una red neuronal. Normalmente, la red contiene varias capas de perceptrones.

La selección del método de clasificación depende del número de clases y de la distribución natural de las huellas dactilares dentro de las clases. Desafortunadamente, el número de clases es muy pequeño y, además, las huellas no se distribuyen de manera uniforme dentro de las clases. Estudios prácticos han demostrado que la distribución es la siguiente: lazo izquierdo 32%, lazo derecho 32% y espiral 28%. Solo el 8% restante son arcos y arcos tipo tienda de forma equitativa. Un problema añadido a la clasificación de huellas es que hay muchas huellas cuya clase es muy ambigua y se duda entre una clase y otra. Por ejemplo, el paso de lazo izquierdo a arco o de arco a lazo derecho es continuo. Por eso, se tiene que poner un umbral y decidir a partir de qué punto creemos que es de un tipo u otro. Se puede usar un sistema que indique el nivel de ambigüedad de la huella, por ejemplo una probabilidad de pertenecer a una clase.

Reflexión

En este módulo, no entramos en explicar en detalle los métodos para deducir automáticamente las clases de huellas dactilares.

Resumen

En este módulo, hemos explicado el proceso de comparación de dos huellas y también el proceso de clasificación. Se han descrito los algoritmos de visión por computador que son necesarios en estos procesos debido a que no podemos suponer que el alumnado del máster tenga conocimientos de estos algoritmos. También se ha descrito cómo se representa una huella dactilar en un registro que no sea una imagen.

Se han descrito los sensores básicos usados para capturar una huella y obtener la imagen digital así como la diferencia que existe desde el punto de vista tanto tecnológico como de aplicación de obtener una imagen en línea y fuera de línea.

También es importante mencionar la descripción de las diferentes clases de huellas dactilares. Este conocimiento puede servir para saber qué tipos de huellas tenemos. También hemos aprendido a saber comparar dos huellas dadas dos imágenes y deducir si pertenecen al mismo dedo o no (obviamente no nos hemos convertido en expertos forenses).

Finalmente, hemos aprendido de rebote algunos métodos de visión por computador. Estos métodos se usan normalmente en procesos industriales más sencillos. Pero, debido a que el campo de la visión por computador está entrando con mucha fuerza en el sector, son conocimientos importantes de cara a un ingeniero o un informático.

Actividades

1. Rellenad la tabla 3 consultando los apuntes y otras fuentes de información.

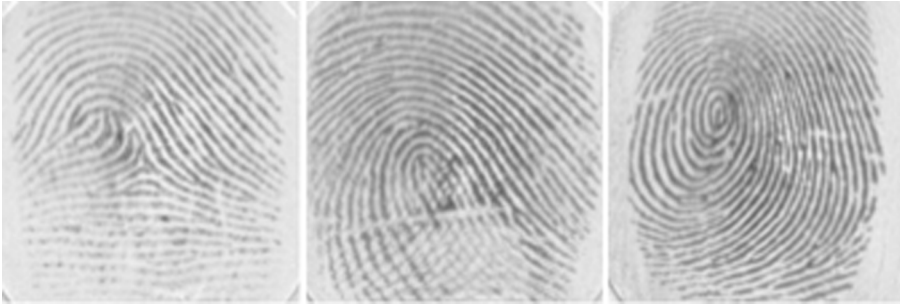
Tabla 3

	Tecnología	Ventajas	Inconvenientes
Óptico			
Capacitancia			
Térmicos			
Ultrasonido			

2. En un sensor de huellas de 200 píxeles por pulgada y en uno de 600 píxeles por pulgada, ¿cuántos píxeles ocupa una cordillera y un valle? ¿Cuál creéis que puede ser la mínima resolución de un sensor?

3. Marcad sobre las imágenes de la figura 34 las tres posibles singularidades y clasificad las huellas dactilares.

Figura 34



4. Marcad sobre la imagen de la figura 35 dos *minutiae* que sean una de cada tipo. Además, rellenad la tabla 4. Supongamos que la coordenada (0, 0) es la inferior izquierda. Suponemos que 1 píxel ocupa 1 milímetro.

Figura 35

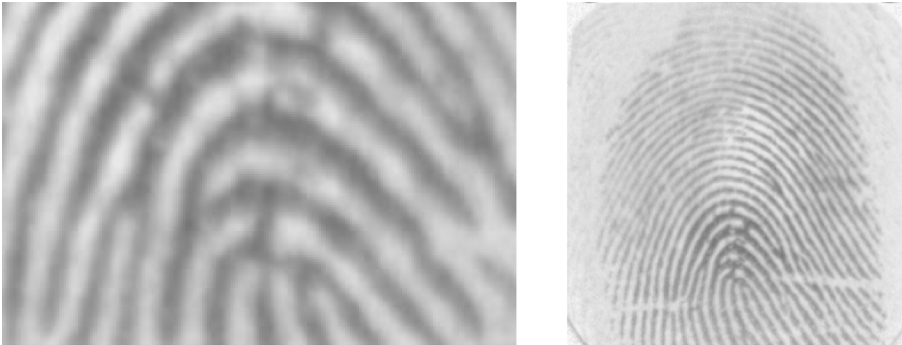


Tabla 4

	Tipo	Coordenada x	Coordenada y	Ángulo (radianes)
1	Terminal			
2	Bifurcación			

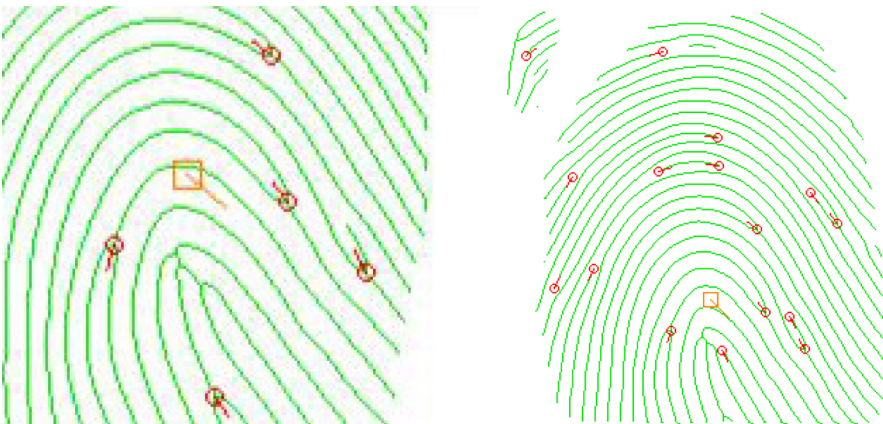
5. Comentad cuáles son los tres niveles de abstracción de información de las huellas dactilares. Describid también qué niveles son los más usados y desde cuándo.
6. Comentad por qué en la matriz de orientación solo hay ángulos comprendidos en dos cuadrantes.
7. ¿Por qué es necesario trabajar sobre la imagen de orientación filtrada en lugar de la imagen de orientación original en los procesos finales de la comparación de huellas?
8. Comentad cuál es el objetivo de la matriz de consistencia y su necesidad.
9. Deseamos verificar si una imagen de una huella parcial (figura 36a) forma parte de una huella (figura 36b).

Figura 36



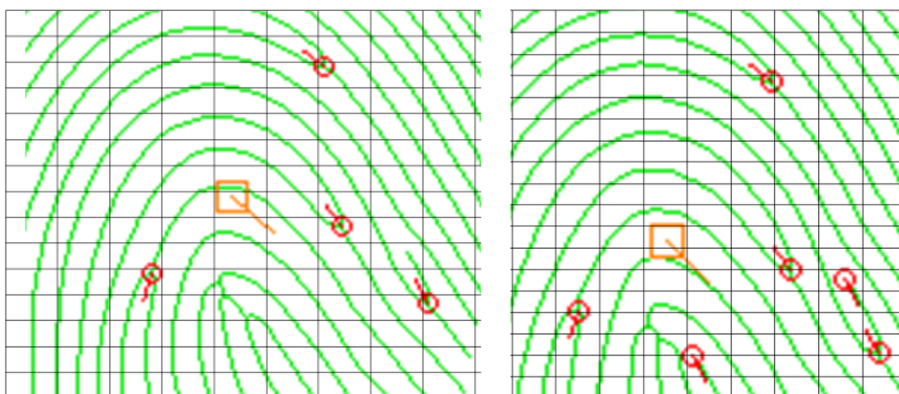
Para hacerlo, primero hemos extraído el esqueleto de las sierras y también las *minutiae* (círculos en rojo) y el punto central de la huella (cuadrado rojo).

Figura 37



Entonces hemos alineado las imágenes tal y como se muestra en la figura 38.

Figura 38



Decidid un buen alineamiento entre *minutiae* y calculad el coste de este alineamiento. Las celdas representan la resolución que cabe considerar a la hora de calcular la distancia. La celda (0, 0) es la de la izquierda abajo. Mostrad la ecuación usada para calcular el coste del alineamiento.

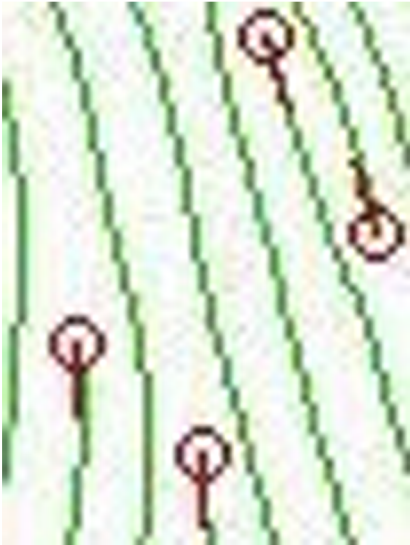
10. Dadas las imágenes de huellas parciales de las figuras 39a y 39b y los registros obtenidos, calculad la alineación y su coste, siendo los valores de las *minutiae*:

a) {[T, (9, 44), 180°], [T, (25, 58), 181°], [T, (37, 8), 202°], [T, (36, 24), 10°]}

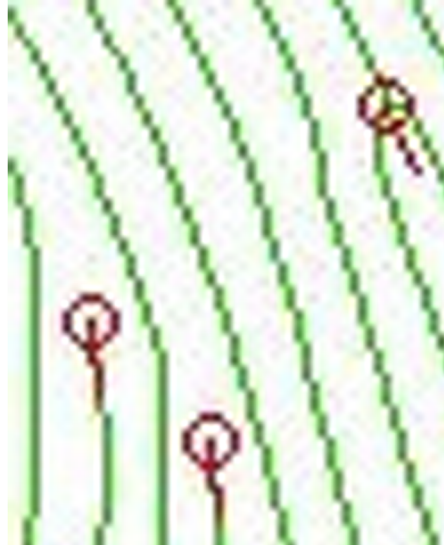
b) {[T, (10, 49), 181°], [T, (26, 58), 182°], [B, (39, 16), 206°]}

Figura 39

a)



b)



Abreviaturas

α_{xy} ángulo del gradiente máximo de la cordillera en su punto (x, y) ; Es ortogonal a θ_{xy}

Δ singularidad de tipo delta (delta)

Δ_x matriz del gradiente horizontal. $\hat{\Delta}_x$, matriz del gradiente horizontal filtrada

Δ_y matriz del gradiente vertical. $\hat{\Delta}_y$, matriz del gradiente vertical filtrada

B imagen binaria

C matriz de coherencia

C matriz resultado de convolucionar una imagen **I** con una máscara **W**

D matriz direccional $D[x,y] \in [-\frac{\pi}{2}, \frac{\pi}{2}]$; matriz de los ángulos de las cordilleras $D(x,y) = \theta_{xy}$.

\hat{D} , matriz direccional filtrada

E imagen eskeletonizada

E imagen original de la huella dactilar

I imagen original de la huella dactilar normalizada

I y T imágenes *input* y *template* de las huellas dactilares

O singularidad de tipo espiral (*whorl*)

r matriz de consistencia $r[x, y] \in [0, \infty[$

S matriz segmentada

U singularidad de tipo lazo (*loop*)

W máscara de convolución, normalmente es una imagen de 3×3

Bibliografía

Jain, Anil; Flynn, Patrick; Ros, Arun (ed.) (2008). *Handbook of biometrics*. Springer.

Maltoni, Davide; Maio, Dario; Jain, Anil; Prabhakar, Salil (2009). *Handbook of fingerprint recognition*. Springer.

Anexo

Las matrices de píxeles o imágenes no normalizadas que vamos obteniendo en el proceso de comparación de dos huellas dactilares como, por ejemplo, la matriz del gradiente horizontal Δ_x y la matriz del gradiente vertical Δ_y , la matriz de consistencia (norma del gradiente) r o la matriz del ángulo del gradiente D o la matriz del ángulo filtrada \hat{D} no se pueden visualizar directamente puesto que no se encuentran en el rango normalizado de los píxeles $[0, 255]$. Por este motivo, cuando nos muestran estas imágenes, en realidad no es directamente el valor sino una imagen normalizada. El proceso de normalización depende del rango de valores original de estas matrices. A continuación, os mostramos tres procesos de normalización que hemos comprobado que funcionan bien.

1) Normalización de las matrices de los gradientes

Mostramos los valores para Δ_x , pero lógicamente también sirve para Δ_y . Las imágenes normalizadas son Δ'_x y Δ'_y . Vemos que el rango de valores es:

$$\Delta_x[x, y] \in [-4 \cdot 255, 4 \cdot 255] \quad (51)$$

Debido a que pocas veces se logran valores extremos y que la media está centrada en el cero, entonces decidimos truncar estos valores extremos y añadir un desplazamiento que es la mitad del rango máximo normalizado:

$$\Delta'_x[x, y] = \text{Truncate}_{[0, 255]}(\Delta_x[x, y] + 127) \in [0, 255] \quad (52)$$

2) Normalización de la matriz de consistencia

Debido a que la consistencia siempre toma valores positivos, simplemente truncaremos el valor inicial:

$$r[x, y] = \text{Truncate}_{[0, 255]}(r[x, y]) \in [0, 255] \quad (53)$$

3) Normalización de la matriz direccional

El rango de valores de la matriz direccional es:

$$D[x, y] \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (54)$$

La transformación que proponemos está compuesta de tres pasos. Primero, convertimos el rango de valores a positivo sumando $\frac{\pi}{2}$. Segundo, normalizamos para que el rango sea de 0 a 1 dividiendo por π . Y, tercero, multiplicamos por 255 para tener el rango completo. La expresión final es:

$$D[x, y] = \frac{D[x, y] + \frac{\pi}{2}}{\pi} \cdot 255 \in [0, 255] \quad (55)$$

Fijaos en que el negro absoluto, valor 0, equivale a un ángulo original de $-\frac{\pi}{2}$. El valor gris, 127, equivale al ángulo 0. El valor blanco, 255, equivale al ángulo $\frac{\pi}{2}$. Obviamente, se podría haber usado otro convenio, donde por ejemplo el ángulo 0 fuera el negro absoluto.

