

# Diseño de un Sistema biométrico para el reconocimiento de huellas dactilares

First A. Author, *Fellow, IEEE*, Second B. Author, and Third C. Author, Jr., *Member, IEEE*

**Abstract**—Biometric systems uses information about a person metrics to identify an individual. During the last years, computer vision has been used to develop and improve different technics for individual identification based on biometric parameters. Studies focused in the filtering process and implement methods on the frequency and time domain, like Notch Filter or Gabor filter. Usually these technics are only tested but they never reach the implementation phase. This study focuses on the development of a fingerprint recognition and its implementation on an embedded system. This article describes the process on finding an ideal filtering algorithm, the develop of the subsequent stages of the process, like the binarization, thinning and minutiae matching, and the implementation of the process in an embedded system which in this case is a Raspberry Pi 3.

**Index Terms**—Biometric; Gabor Filter; Embedded System; Fingerprint; Computer Vision;

**Resumen**—Los sistemas biométricos usan información sobre las métricas de la persona para identificar al individuo. Durante los últimos años, la visión por computadora se ha utilizado para desarrollar y mejorar diferentes técnicas de identificación individual basadas en parámetros biométricos. Los estudios se centraron en el proceso de filtrado e implementan métodos sobre la frecuencia y el dominio del tiempo, como Filtro Notch o Filtro Gabor. Por lo general, estas técnicas sólo se prueban, pero nunca llegan a la fase de implementación. Este estudio se enfoca en el desarrollo de un algoritmo de reconocimiento de huellas y su implementación en un sistema embebido. Este artículo describe el proceso de búsqueda de un algoritmo de filtrado ideal, el desarrollo de las etapas posteriores del proceso, como la binarización, adelgazamiento y minucias de concordancia, y la implementación del proceso en un sistema embebido que en este caso es un Raspberry Pi 3.

## I. INTRODUCCIÓN

En los últimos años los sistemas biométricos han ido adquiriendo una importancia en el tema de seguridad, ya que son difíciles de filtrar y utilizan la particularidad de las características biológicas, que son únicas para cada individuo. Teniendo en cuenta lo anterior, se han desarrollado diferentes algoritmos de visión artificial que permiten mejorar la precisión de estos sistemas. En el caso específico de las huellas dactilares, se han llevado a cabo estudios en todas las etapas del proceso. Desde el filtrado de la señal de entrada, pasando por la

binarización y adelgazamiento de la muestra, la detección de minucias, y los métodos de comparación.

Teniendo en cuenta que la etapa del filtrado es una de las más importantes del proceso, se han probado diferentes algoritmos de filtrado de imágenes y se han evaluado su desempeño, como el estudio realizado por Kazi [1], el cual probó tres métodos de filtrado como la eualización del histograma, la FFT (Fast Fourier Transform) y el filtro Gabor.

De la misma manera, se encuentran trabajos que también se enfocan en los algoritmos de detección de minucias como es el caso del trabajo desarrollado por Babatunde [2] o Manu Garg [3], quienes prueban y validan distintos métodos de estimación y comparación de minucias en una huella. También se han desarrollado trabajos en otras etapas del proceso de reconocimiento, como la binarización, la cual se han realizado trabajos para determinar el umbral para la binarización de la imagen, como lo hicieron Doo-Rae [4], como el realizado por Tomohiko [5], quien hizo un sistema de detección de núcleos y deltas en la huella para hacer una comparación por estos atributos.

En este trabajo en particular se plantea entonces la implementación de un algoritmo de detección de huellas basado en minucias y utilizando un filtro Gabor principalmente, en la etapa de pre procesamiento.

## II. DISEÑO DEL SISTEMA BIOMÉTRICO

Un sistema biométrico es un sistema automatizado que fundamenta sus decisiones de reconocimiento por medio de una característica única personal que puede ser reconocida o verificada de manera automatizada [6].

Es un sistema capaz de obtener la muestra biométrica del usuario, extraer los datos de la muestra, comparar los datos obtenidos con lo ya existentes en la base de datos e indicar el resultado de la verificación.

### A. Huella dactilar

Es la representación de la forma de la piel de las yemas de los dedos. Es un patrón que permanece invariable durante toda la vida del individuo, por lo que permite realizar por medio de esta un reconocimiento de identidad [6].



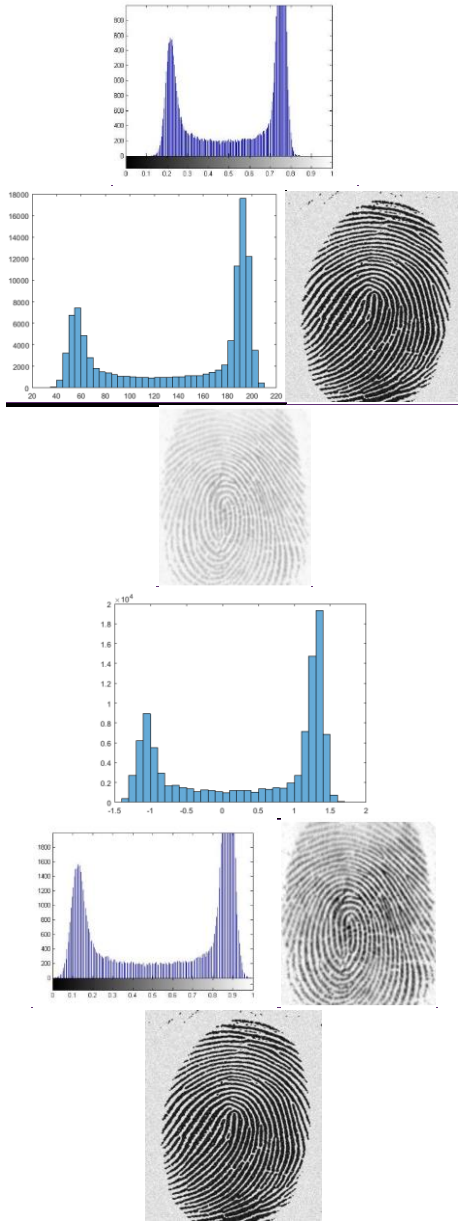


Fig. 3: Normalización aplicada a la huella.

En esta etapa se vio que cuando la media y desviación estaban por debajo de los valores asumidos, las etapas posteriores

presentaban problemas, ya que tomaban parte de la huella como si no tuviese información, mientras que, en el caso contrario, al ser más altos que el valor ideal, tomaba información no relevante.

#### D. Segmentación

La segmentación es el proceso de separar el fondo de la imagen de la parte de la huella que contiene información.

En la imagen de una huella, la región del fondo generalmente presenta una varianza en escala de grises muy baja, mientras que el primer plano, tiene una varianza más alta. Por ende, se utiliza un método basado en un umbral de varianza para realizar la segmentación.

La implementación de la segmentación en la imagen de la huella inicia con la división de la imagen en bloques de  $W \times W$ , se calcula la varianza de cada bloque y si esta es menor al umbral, que en este caso es de 0.25 de varianza, se toma como primer plano, de lo contrario, el umbral se toma como parte del fondo [9].

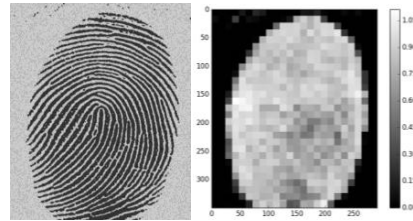


Fig. 4: Segmentación aplicada a la huella.

En el caso de la segmentación se utilizaron bloques de  $9 \times 9$  pixeles. Al utilizar bloques más pequeños, hacía que la eliminara información de la huella, mientras que, si los bloques eran más grandes, iba a tomar partes del fondo como información de la huella, además de aumentar los tiempos de procesamiento.

#### E. Filtro Gabor

El filtro Gabor consiste en el producto entre una función Gaussiana y una senoidal [10] y su implementación consiste en encontrar los valores donde la imagen tenga valores de frecuencia diferentes de cero y es allí donde se aplica el filtro con el objetivo de resaltar las crestas y reducir el ruido en la muestra. El filtro está dado por:

$$G(x, y, \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_{\theta}) \quad (2)$$

$$x_{\theta} = x \cos\theta + y \sin\theta \quad (3)$$

$$y_{\theta} = -x \sin\theta + y \cos\theta \quad (4)$$

Donde  $\theta$  es el ángulo de orientación del filtro gabor,  $f$  la frecuencia y donde  $\sigma_x$  y  $\sigma_y$  son las desviaciones del gaussiano.

Con formato: Español (Colombia)

Con formato: Sin Resaltar

En este caso  $\sigma_x$  y  $\sigma_y$  tienen un valor de 0.5. Este valor es importante en la medida de que, si es muy grande, tendrá un mejor efecto ante el ruido, pero tiende a crear falsas crestas o minucias, mientras que si son muy bajos tendrá menos efectos sobre el ruido [11]. En la figura 5 encontramos lo que sería un filtro Gabor en 2D, donde la señal  $\text{sen}\theta$  tiene una orientación de  $0^\circ$  y una frecuencia de  $1/10$ , con unos valores de  $\sigma_x$  y  $\sigma_y$  de 4, que serán los que tendrá el filtro, permitiéndole actuar específicamente en esos ángulos.

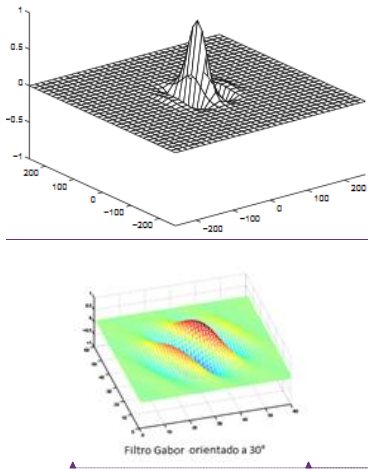


Fig 5. Filtro Gabor en dos dimensiones [11][10].

Con las frecuencias y las orientaciones de cada pixel, se realiza entonces, para cada zona con información relevante, un filtro diferente dependiendo de su orientación y frecuencia estimada, obteniendo los resultados presentes en la figura 6.



Fig. 6: Filtro Gabor aplicada a la huella.

#### F. Estimación de la orientación

La estimación de se utiliza para el mejoramiento de la imagen, en este caso se utiliza el método desarrollado por Kass [11], para el análisis de patrones orientados, utilizando los gradientes en "x", y gradientes en "y" de cada pixel para hallar el ángulo que forman estos gradientes, siendo esta la orientación.

1. Se halla el gradiente en x y el gradiente en y, El operador Sobel horizontal es utilizado para computar el  $\partial x(i,j)$ :

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (5)$$

Y el operador Sobel vertical para determinar  $\partial y(i,j)$ :

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (6)$$

2. se aplica la fórmula de arco tangente para encontrar la orientación de los pixeles en la imagen:

$$\theta(i,j) = \tan^{-1} \frac{\partial y(i,j)}{\partial x(i,j)} \quad (7)$$

Donde  $\theta(i,j)$  es la estimación que necesitamos.

3. Se realiza el cálculo de  $\Phi_x$  y  $\Phi_y$  utilizando el ángulo hallado con los operadores Sobel:

$$\Phi_x(i,j) = \cos(2\theta(i,j)) \quad (8)$$

$$\Phi_y(i,j) = \sin(2\theta(i,j)) \quad (9)$$

4. Teniendo los valores de  $\Phi_x$  y  $\Phi_y$ , se les aplica un filtro gaussiano de orden cero y una desviación estándar de 5, obteniendo la siguiente máscara.

$$\begin{bmatrix} 0.109634 & 0.111842 & 0.109634 \\ 0.111842 & 0.114094 & 0.111842 \\ 0.109634 & 0.111842 & 0.109634 \end{bmatrix} \quad (10)$$

5. Por último, con la salida de los procesos implementados anteriormente, se encuentra la orientación final ya suavizada.

$$O(i,j) = \frac{1}{2} \tan^{-1} \frac{\Phi'_x(i,j)}{\Phi'_y(i,j)} \quad (11)$$

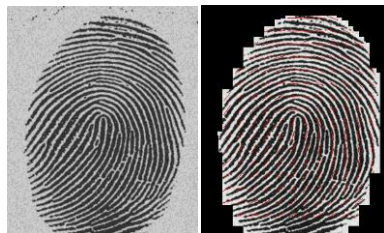


Fig. 7: Estimación de la orientación aplicada a la huella.

En la figura 6, se encuentra la validación de la estimación de la orientación. En este caso se dibujó una en línea roja con la orientación estimada en ciertos puntos, para ver que la orientación estimada era acorde a la presente en la imagen.

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: 10 pto

Con formato: Fuente: (Predeterminada) Times New Roman, 10 pto

Con formato: Fuente: (Predeterminada) Times New Roman, 10 pto

Con formato: Fuente: (Predeterminada) Times New Roman, 9 pto, Sin Cursiva, Color de fuente: Automático

Con formato: Fuente: (Predeterminada) Times New Roman, 9 pto, Sin Cursiva, Color de fuente: Automático

Con formato: Fuente: (Predeterminada) Times New Roman, Sin Cursiva, Color de fuente: Automático

### G. Estimación de la frecuencia

Otro parámetro importante en la implementación del filtro Gabor, es la frecuencia local de las crestas. En este caso se utilizó el método planteado por Raymond Thai [9], en el cual se recorre la imagen utilizando bloques cuadrados del mismo tamaño que los utilizados en la segmentación (9x9), y se halla la media de las orientaciones en los píxeles de ese bloque. Con esto, se gira el bloque la media de la orientación establecida de forma que las crestas queden verticales y permita hacer una proyección de la intensidad de estos píxeles por columnas, obteniendo como resultado un gráfico como el que se presenta en la figura 8.

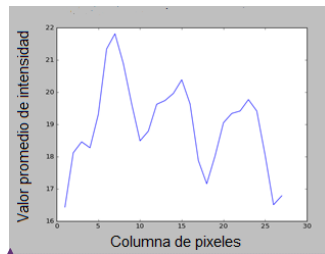


Fig. 8: Filtro Gabor aplicada a la huella.

A partir de esto, se hallan los máximos de la onda y haciendo uso del primer y último máximo de la imagen, se encuentra la diferencia en píxeles entre ellos, siendo esta el periodo, y la inversa de la misma, la frecuencia. Esta frecuencia se les asigna entonces a todos los píxeles de ese bloque. En la figura 7 se observa la imagen de los bloques de la huella y la diferencia de valores de frecuencia en la misma.

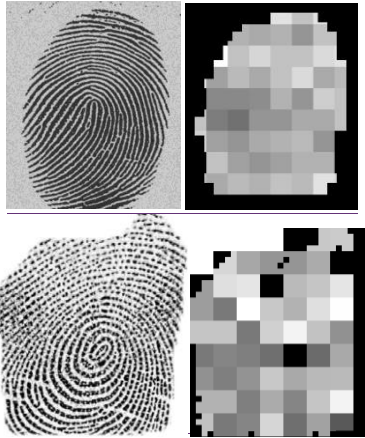


Fig. 7: Estimación de la frecuencia aplicada a la huella.

Con las frecuencias y las orientaciones de cada píxel, se realiza entonces, para cada zona con información relevante, el proceso

de filtrado. La implementación de este consiste en la convolución espacial de la imagen con la máscara de coeficientes perteneciente al filtro, un filtro diferente dependiendo de su orientación y frecuencia estimada, obteniendo los resultados presentes en la figura 6.



Fig. 6: Filtro Gabor aplicada a la huella.

### H. Binarización

La binarización de una imagen consiste en comparar los niveles de gris presentes en la imagen con un valor (umbral) predeterminado [12]. Si el nivel de gris de la imagen es menor que el umbral predeterminado, al píxel de la imagen binarizada se le asigna el valor 0 (blanco), y si es mayor se le asigna un 1 (negro). Este valor de umbral se determinó de manera experimental como cero, ya que al aumentarlo o disminuirlo, agregaba o eliminaba información de la huella respectivamente. De esta forma se obtiene una imagen en blanco y negro. En la figura ## se muestra un ejemplo de imagen binarizada.



## IV. ADELGAZAMIENTO Y DETECTOR DE MINUCIAS

En esta etapa se reducen el grosor de las crestas de la huella para poder hallar más fácilmente las minucias utilizando el método de Crossing Number [7].

### A. Adelgazamiento

En este caso para el adelgazamiento de la huella se hace un proceso de esqueletización de las crestas [13]. Para esto se utilizó el método desarrollado por Guo-Hall [14]. Este método es un algoritmo de dos iteraciones usando operadores con un soporte paralelo, es decir un soporte 3x3.

P1	P2	P3
P8	$P_i$	P4

Con formato: Sin Resaltar

Con formato: Sin Resaltar



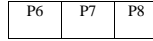
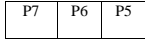


Fig. 8: Matriz de adelgazamiento 3x3.

Como vemos en la figura 9, se refiere a P2,P4,P6 y P8 como vecinos laterales, y P1,P3,P5 y P7 como vecinos diagonales. De igual manera se define C(p), como el número de distintos pixeles 8-conectados en los vecinos de Pi y B(p) como el número de pixeles con valor de 1 en los vecinos de Pi. Adicional, para un mejor entendimiento del proceso, se utilizan los símbolos ¬, && y || como operadores lógicos, de negación, AND y OR, respectivamente. Por último, se tiene las variables  $N_1(p)$  y  $N_2(p)$ , la cual es útil para obtener un mejor adelgazamiento. Con lo anterior tenemos entonces que:

$$N_1(p) = (p_1 || p_2) + (p_3 || p_4) + (p_5 || p_6) + (p_7 || p_8) \quad (11)$$

$$N_2(p) = (p_1 \&\& p_2) + (p_3 \&\& p_4) + (p_5 \&\& p_6) + (p_7 \&\& p_8) \quad (12)$$

$$N(p) = \min(N_1(p), N_2(p)) \quad (13)$$

Con lo anterior se tiene entonces que un pixel con valor de uno se eliminara (se cambiara su valor por cero) si cumple las 3 condiciones siguientes.

- A. C(p)=1
- B.  $2 \leq N(p) \leq 3$
- C. Aplica una de las siguiente:
  1.  $p_2 || p_3 || \bar{p}_4 = 0$  en una iteración impar
  2.  $p_6 || p_7 || \bar{p}_1 = 0$  en una iteración par.

El proceso se realiza entonces hasta que no se eliminan más pixeles después de una iteración.

El resultado de la implementación del adelgazamiento se observa en la imagen 9.

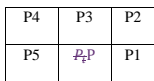


Fig. 9: Resultado de la aplicación de la transformada "hit or miss" en una imagen.

**B. Detector de minucias**

El método utilizado para la detección de las minucias es el Crossing number(CN) [7]:

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}| \quad (14)$$



Si el valor de CN es igual a 1 quiere decir que la minucia es una terminación y si es igual a 3 quiere decir que esta es una bifurcación. Teniendo entonces el tipo de minucia y la posición donde se encuentra, tenemos de la estimación, la orientación de la misma, y será esta información lo que se guardará como se ve en la siguiente tabla

TABLA II. INFORMACION DE LA MINUCIA.

Posición en X	Posición en Y	Tipo de Minucia	Orientación en radianes
460	217	1(Terminación)	-1.03
510	340	3(Bifurcación)	-0.624
530	200	3(Bifurcación)	-1.31
540	380	1(Terminación)	0.906
540	229	1(Terminación)	-0.811
550	258	1(Terminación)	-0.605

- Con formato
- Con formato
- Con formato
- Con formato

**C. Comparación**

Este método busca alinear las minucias de la imagen de entrada con las minucias de la huella base y encontrar el número de minucias coincidentes. Si tomamos la tabla # como las minucias de la huella base, lo que se busca es alinear en posición y ángulo las minucias de entrada con las de la huella base, variando en posición y orientación, y verificara si las otras minucias coinciden en ángulo y posición según el valor de tolerancia dado, que en este caso fue de 5 pixeles y 5 grados.

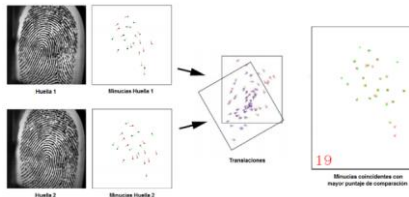


Fig. 10: Método de comparación basado en minucias.

Según la cantidad de minucias se dará un puntaje de similitud, y se repetirá el proceso con las demás minucias, al final se toma la comparación con mayor puntaje y con esta se determinará si es o no la persona. Este puntaje está dado por la ecuación:

$$\sqrt{\frac{\beta^2}{\alpha_i * \alpha_f}} \quad (15)$$

Donde  $\beta$  es el índice de similitudes encontradas y donde  $\alpha_i$  y  $\alpha_f$  representan el número de minucias de cada una de las huellas (inicial y final). Para este caso, se determinó que un puntaje de

[comparación mayor a 0.8, determina que la huella base y la de entrada, coinciden siendo el mismo individuo.](#)

## V. IMPLEMENTACIÓN DEL SISTEMA BIOMÉTRICO

En esta sección se detalla la información del sistema embebido que se utilizó y de los lenguajes de programación utilizados para el desarrollo del proyecto. En esta etapa es importante mencionar que al momento de ingresar una huella se debe realizar una buena presión sobre el sensor para evitar ruido en la toma y los detalles de la huella queden bien marcados. Igualmente, al momento de ingresar una nueva huella a la base de datos, se implementó un método que consistía en tomar 3 huellas al momento del registro de una nueva huella. Solo si la primera toma coincidía con las dos siguientes, se guardaba en base de datos, en caso contrario, pedía el ingreso de las huellas nuevamente. Esto ayudó de igual manera al rendimiento del sistema.

### A. Sistema embebido

Es un sistema electrónico diseñado para realizar determinadas funciones [15], emplea uno o varios procesadores digitales lo que permite aportar "inteligencia" al sistema anfitrión.

Para este caso, se seleccionó la tarjeta integrada Raspberry teniendo en cuenta el trabajo de Casco [16], que indica que esta tarjeta es ideal para proyectos de multimedia complejos o basados en Linux y para el desarrollo de proyectos que requieran una interfaz gráfica y con varias entradas seriales.

Dentro del mercado podemos encontrar diferentes tipos de tarjetas Raspberry, de las cuales se seleccionó la Raspberry pi 3 basados en un test de CPU [17].

Para la visualización de la interfaz gráfica, se seleccionó la pantalla táctil de 5" debido a sus características de buena calidad en la imagen, mejorando la interacción con el usuario y su conexión por HDMI que facilitan su comunicación con el microcontrolador.

### B. Interfaz grafica

Haciendo uso de [los marcos de aplicación utilizados para el desarrollo de aplicaciones de software](#), Qt y PyQt, se desarrolló una interfaz gráfica que le permite al usuario interactuar de manera dinámica con el sistema, permitiendo al usuario el ingreso como administrador, para agregar o eliminar un usuario y crear un nuevo administrador, y el ingreso como usuario para realizar la comparación de la huella.

## VI. RESULTADOS

En esta etapa del desarrollo, se hicieron las pruebas para evaluar el funcionamiento del sistema y su respuesta a ciertas variaciones a la hora de tomar una muestra.

Se tomaron 25 huellas base y se realizaron las pruebas de comparación entre ellas mismas, siendo un total de 625 comparaciones.

Los resultados obtenidos en esta prueba se observan a continuación:

TABLA III. RESULTADO DE LA VALIDACION DEL SISTEMA

True Positive (TP)	21
False Negative (FN)	4
True Negative (TN)	502
False Positive (FP)	98

Donde True Positive (TP), se da cuando una huella es correctamente identificada, True Negative (TN), cuando una huella no es reconocida correctamente por el sistema, False Positive (FP), cuando el sistema da un resultado positivo a dos huellas diferentes y False Negative (FN), cuando el procesamiento da un resultado negativo a pesar de ser la misma huella.

Siguiendo los estándares propuestos por la National Institute of Standards and Technology se evaluaron estos resultados hallando los valores de True Acceptance Rate, el False Acceptance Rate  $\gamma$ , la precisión del sistema (PPV), ~~el False Acceptance Rate (FAR) y el False Negative Rate (FNR) que son medidas utilizadas para la evaluación de los sistemas biométricos~~, definidas por:

$$TPR = \frac{TP}{TP+FN} = 0.84 \quad (16)$$

Lo que sugiere que el sistema tiene una sensibilidad del 84 %.

$$TNR = \frac{TN}{FP+TN} = 0.84 \quad (17)$$

$$PPV = \frac{TP}{TP+FP} = 0.84 \quad (18)$$

[De igual manera, se calcula el FAR \(False Acceptance Rate\), el cual es la probabilidad de un sistema biométrico de aceptar incorrectamente una persona y el FNR \(False Negative Rate\), que es la probabilidad de que una persona sea rechazada incorrectamente. Siendo estos dos valores los utilizados para medir la calidad del sistema.](#)

$$FAR = \frac{FP}{FP+TN} = 1 - TNR = 0.16 \quad (19)$$

$$FNR = \frac{FN}{TP+FN} = 1 - TPR = 0.16 \quad (20)$$

## VII. CONCLUSIONES Y TRABAJO FUTURO

El tiempo de procesamiento se verá afectado también si los parámetros como los de normalización y segmentación, no son adecuados, ya que el algoritmo tendrá que hacer más ciclos para realizar el trabajo.

Se recomienda probar otros métodos de comparación de huellas [19, 20, 21] basados en minucias para validar si es posible tratar de optimizar todo el proceso en general y obtener mejores resultados.

#### BIBLIOGRAFÍA

- [1] K. M. M., "Comparison of fingerprint enhancement techniques through Mean Square Error and Peak-Signal to Noise Ratio," *International Journal on Computer Science and Engineering*, vol. 3, no. 1, 2011.
- [2] Babatunde, "A Modified Approach to Crossing Number and Post-processing Algorithms for Fingerprint Minutiae Extraction and Validation," *Publishing India*, p. 7.
- [3] M. Garg, "Fingerprint Recognition System using Minutiae Estimation," *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 5, p. 6, 2013.
- [4] D.-H. K. a. R.-H. Park, "Fingerprint Binariz".
- [5] T. Ohtsuka, "Fingerprint Core and Delta Detection by Candidate Analy," *Conference on Machine Vision Application*, 2007.
- [6] G. Gabriela, "Extracción de características y comparación de una huella digital," Escuela Superior politécnica del litoral, Guayaquil, Ecuador, 2009.
- [7] J. A. A. Molano, Verificación de correspondencia en huellas dactilares aplicando técnicas de procesamiento y análisis digital de imágenes para la disminución del tiempo de cotejo, Popayán, Cauca: Universidad del Cauca, facultad de Ciencias Naturales, Exactas y de la Educación., 2010.
- [8] D. Persona, "2fa.com," [Online]. Available: [http://2fa.com/wp-content/uploads/Datasheets/UareU\\_4500Reader20100416.pdf](http://2fa.com/wp-content/uploads/Datasheets/UareU_4500Reader20100416.pdf). [Accessed 14 10 2016].
- [9] R. Thai, "Fingerprint Image Enhancement and Minutiae Extraction," The university of Western Australia, Stirling Hwy, Australia, 2003.
- [10] S. N. Prasad, "Gabor filter visualization," Maryland, 2010.
- [11] M. Kaas, "Analyzing Oriented Patterns," Schlumberger Palo Alto Research, Palo Alto, 2008.
- [12] J. J. Esqueda, Fundamentos de procesamiento de imágenes, Baja California: Universidad Autonoma de Baja California, 2005.
- [13] NTNU, "Hit-or-miss transform," [Online]. Available: <http://www.cs.tau.ac.il/~turkel/notes/hit-or-miss.pdf>. [Accessed 8 Febrero 2016].
- [14] Z. G.-R. Hall, "Parallel Thinning with two-subiterations algorithms," *Image Processing and computer vision*, vol. 32, p. 359, 1989.
- [15] B. U. Miñarro, "Sistemas Embebidos," 2006. [Online]. Available: <http://ocw.um.es/ingenierias/sistemas-embbedidos/material-de-clase-1/ssee-t01.pdf>. [Accessed 6 Septiembre 2016].
- [16] T. Klosowski, "El rendimiento de la Raspberry Pi 3, comparado contra los modelos anteriores," 3 marzo 2016. [Online]. Available: <http://es.gizmodo.com/el-rendimiento-de-la-raspberry-pi-3-comparado-contra-1-1762657409>. [Accessed 14 Octubre 2015].
- [17] A. Kopytov, "Sysbench," Gizmodo, 2016. [Online]. Available: <https://github.com/akopytov/sysbench>. [Accessed 29 11 2016].
- [18] Python, "What is Python? Executive Summary," Python, 2016. [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed 29 Octubre 2016].
- [19] OpenCv, "Open Source Computer Vision," 2016. [Online]. Available: <http://opencv.org/>. [Accessed 26 Julio 2016].
- [20] SQLite, "About SQLite," [Online]. Available: <https://sqlite.org/fileformat2.html>. [Accessed 21 Julio 2017].

Con formato: Español (Colombia), No revisar la ortografía ni la gramática