

Análisis de desempeño de un filtro para el mejoramiento de imágenes de huellas dactilares

Victor Hugo García Ortega, Osvaldo Espinosa Sosa

Centro de Investigación en Computación, IPN,
Av. Juan de Dios Batíz, esq. M. Otón de Mendizábal, 07738
México, D.F
v.garciaortega@lycos.com

Resumen. Este trabajo presenta los resultados de un algoritmo basado en la Transformada de Fourier para el filtrado y mejoramiento de imágenes de huellas dactilares, dicho algoritmo es parte de un sistema AFAS (Automatic Fingerprint Authentication System), el cual esta basado en la detección de minucias en huellas dactilares para el control de acceso a recintos. Este algoritmo fue probado en imágenes de buena y mala calidad obteniendo resultados muy satisfactorios. El algoritmo esta sustentado en los trabajos conocidos, sin embargo, presenta variantes con respecto a éste, además, fue probado en una computadora personal Pentium III y un Procesador Digital de Señales (DSP) a fin de poder medir y comparar el tiempo de ejecución del algoritmo en este procesador de arquitectura específica para la ejecución de algoritmos como el que aquí se plantea

1 Introducción

Básicamente los sistemas biométricos [12] basados en huellas dactilares son de dos tipos [8]:

- Automatic Fingerprint Authentication System (AFAS).
- Automatic Fingerprint Identification System (AFIS).

En un AFAS la entrada es la identidad de la persona y la imagen de la huella dactilar de esa persona; y la salida es una respuesta de SI ó NO, indicando si la imagen de entrada pertenece a la persona cuya identidad es proporcionada.

En un AFIS la entrada es solo la imagen de la huella dactilar y la salida es una lista de identidades de personas que pueden tener la huella dada, además de una puntuación de cada identidad indicando el grado de similitud entre ésta y la huella dada.

Ambos sistemas utilizan los detalles formados en las huellas dactilares. Estos detalles llamados "*ridges*" son definidos como un segmento de curva simple. La combinación de varios *ridges* forman un patrón de huella dactilar. Las pequeñas características formadas por el cruce y terminación de *ridges* son llamadas *minucias*. La finali-

dad del filtro propuesto es el poder eliminar todo el ruido posible de la imagen a fin de poder extraer las minucias de la imagen [5,6, 2].

Cabe mencionar que para la adquisición de la imagen se uso el *5th Sense Parallel Evaluation Kit*, el cuál es el Kit de desarrollo del sensor de huella dactilar FPS110. Dicho sensor es fabricado por la compañía Veridicom. El sensor es del tipo capacitivo, el cuál cuenta con una resolución de 500 dpi y con un tamaño de arreglo de píxeles de 300x300 [4].

Existen muchas técnicas específicas para realizar un mejoramiento de huellas dactilares. Las técnicas descritas en [13] y [11] son de las más efectivas, pero estas técnicas están diseñadas para usarse en los sistemas del tipo AFIS (como en sistemas policíacos del FBI); y por lo tanto requieren demasiado tiempo de procesamiento en comparación con sistemas mas pequeños como son los sistemas del tipo AFAS.

El principio básico de operación en estos métodos es dividir a la imagen en bloques, creando un filtro específico para cada bloque basado en la dirección promedio del bloque (imagen direccional) [5, 6, 9] y entonces filtrar cada bloque. En [13] este proceso se hace en el dominio del tiempo y en [11] se hace en el dominio de la frecuencia.

Por lo tanto, se requiere un sistema el cual cree un filtro adaptado a la dirección exacta del bloque. La información direccional de un bloque específico de la imagen esta contenido en la magnitud de la Transformada de Fourier del bloque, por lo que sí se considera un bloque el cual contenga dos o tres ridges aproximadamente paralelos y se toma su Transformada de Fourier, entonces se supone que las frecuencias dominantes de ese bloque corresponden a los ridges en el bloque.

2 Algoritmo

Es posible mejorar un bloque específico por sus frecuencias dominantes multiplicando la Transformada de Fourier del bloque por su magnitud un cierto número de veces, entonces cuando se calcula la Transformada Inversa de Fourier se observa que el bloque ha sido mejorado.

Si multiplicamos por el cuadrado o el cubo de la magnitud de la transformada de Fourier se obtendrían mejores resultados, sin embargo las regiones donde se localizan las minucias suelen ser alteradas, por lo que en vez de multiplicar por factores enteros a la magnitud de la transformada de Fourier se debe utilizar una fracción [7].

Para el cálculo de la transformada de Fourier se implementó el algoritmo de la transformada rápida de Fourier (FFT- Fast Fourier Transform) descomposición en tiempo (Decimation in Time) en la PC y en frecuencia (Decimation in Frequency) en el DSP, ambos base 2 (radix 2).

El algoritmo propuesto consiste básicamente en realizar dos filtrados a la imagen (ver figura 1), donde el segundo filtrado solo se hace a las regiones que lo necesiten (las mas ruidosas). En este trabajo se llegó a la conclusión de que este filtro basado en la Transformada de Fourier trabaja mucho mejor sobre la imagen binaria que sobre la imagen en niveles de gris como se describe en [7]. Por esta razón primero se binariza la imagen con un umbral dinámico y después se filtra. Después de realizar el primer filtrado, en algunas ocasiones suelen haber regiones todavía con demasiado ruido, por

lo que un segundo filtrado es necesario a fin de poder eliminar dicho ruido. El umbral compuesto determina las regiones ruidosas que necesitan el segundo filtrado, las cuales se les aplica el filtro nuevamente.

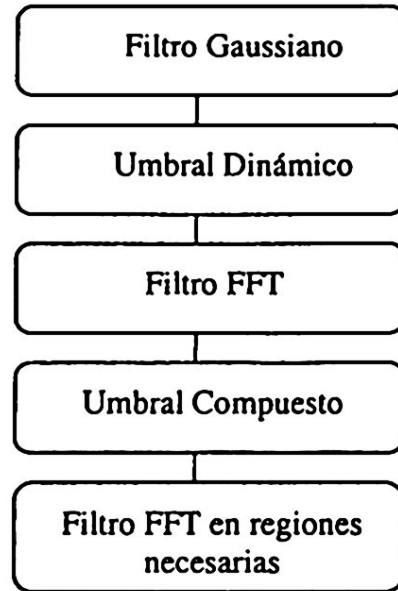


Fig. 1. Algoritmo

2.1 Filtro Gaussiano

Este filtro es un filtro pasabajas, cuyo efecto es el de realizar un suavizado en la imagen, en el cuál el núcleo es una Gaussiana 2-D, de media cero y desviación estándar σ , de la forma:

$$G(x, y) = e^{-(x^2 + y^2)/2\sigma^2} \quad (1)$$

La transformada de Fourier de la Gaussiana es todavía Gaussiana y prácticamente no posee lóbulos laterales, por lo que su respuesta en frecuencia es muy buena.

2.2 Umbral dinámico

Para poder separar claramente las regiones que pertenecen al fondo de la imagen, de los ridges, se requiere aplicar alguna técnica de segmentación, esto se hace binarizando la imagen a través de un proceso de umbralización [10].

Existen diferentes tipos de umbrales, global, local y dinámico [10], este último consiste en dividir la imagen en bloques B_{KL} de 15x15 píxeles, de los cuales se obtiene su media μ_{KL} . Entonces la media μ_{KL} de un bloque B_{KL} , esta definida por:

$$\mu_{KL} = \left(\sum_{j=LB}^{(L+1)(B-1)} \sum_{i=KA}^{(K+1)(A-1)} I(i, j) \right) / (AB) \quad (2)$$

Donde A, B es el tamaño del bloque y K, L es el número de bloques obtenidos de la imagen. Entonces μ_{KL} es el umbral del bloque B_{KL} , de tal forma que cada bloque tendrá un umbral diferente. Este método funciona muy bien sobre imágenes de mala calidad, a diferencia del umbral global que funciona bien en imágenes de buena calidad [1].

2.3 Filtro FFT

Para mejorar a la imagen a través de la magnitud de su transformada de Fourier, ésta es dividida en bloques de 32x32 píxeles, a los cuales su transformada de Fourier $F(u,v)$ es calculada de acuerdo a:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp \left\{ -j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (3)$$

La imagen mejorada $g(x,y)$ en cada bloque es obtenida por:

$$g(x,y) = F^{-1} \left\{ F(u,v) x |F(u,v)|^k \right\} \quad (4)$$

Donde $F(u,v)$ es la transformada de Fourier del bloque y k es la fracción por la cual se multiplica a la magnitud de la transformada de Fourier, y F^{-1} es la transformada Inversa de Fourier obtenida de acuerdo a:

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp \left\{ j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (5)$$

2.4 Umbral compuesto

Puesto que la umbralización global funciona bien en una imagen de buena calidad, y el umbral dinámico en imágenes de mala calidad, en este trabajo se combinaron ambos métodos para obtener un algoritmo al cual se le denominó *umbralización compuesta*. Este algoritmo se muestra en la figura 2. y se utiliza después de realizar el primer filtrado basado en la FFT, en donde a veces, se tienen todavía regiones de mala calidad. La determinación de estas regiones de mala calidad se hace usando el

método de segmentación por varianza [14], en este método la imagen se divide en bloques B_{KL} de 15×15 píxeles de los cuales se extrae su varianza v_{KL} dada por:

$$v_{KL} = \left(\sum_{j=LB}^{(L+1)(B-1)} \sum_{i=KA}^{(K+1)(A-1)} [I(i,j) - \mu_{KL}]^2 \right) / (AB) \quad (6)$$

Donde A, B, K, L tienen el mismo significado que en la ecuación 2. La varianza obtenida de los niveles de gris correspondientes al fondo y/o a regiones uniformes es muy pequeña (cero en regiones perfectamente uniformes), y la varianza obtenida en regiones donde se encuentra la imagen de la huella dactilar en buenas condiciones es alta, con este método los bloques de mala calidad se marcan para que posteriormente estos sean nuevamente filtrados.

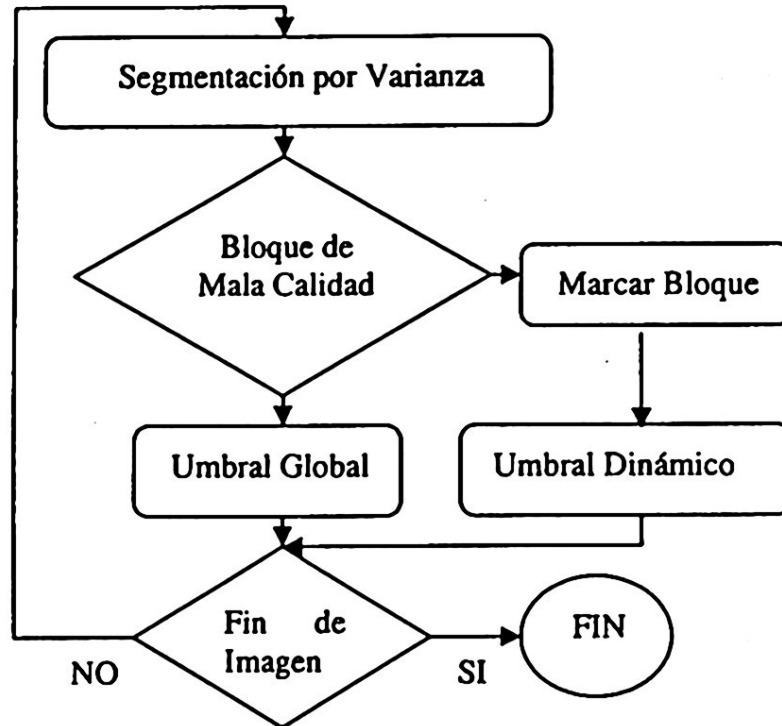


Fig. 2. Algoritmo propuesto denominado umbralización compuesta

3 Plataformas usadas para la ejecución del algoritmo

3.1 Computadora personal

Este algoritmo se programó en una computadora personal (PC) en lenguaje "C" usando el ambiente de programación visual Builder C++ versión 5.0 y en Matlab versión 5.0. Los requerimientos de la PC donde se hicieron las pruebas es el siguiente:

- Computadora personal Pentium III a 550Mhz,
- 128Mb de memoria RAM.

3.2 DSP

El DSP usado pertenece a la familia TMS320C3x, de Texas Instruments. La arquitectura de la familia 'C3x' tiene un alto desempeño debido a su amplio y preciso rango dinámico de unidades de punto flotante, gran capacidad de memoria, alto grado de paralelismo(al tener una arquitectura Harvard) y el controlador de DMA[3].

Específicamente se uso el DSP TMS320C31-50, el cuál realiza el procesamiento de la imagen. Este DSP es un procesador con 32 bits en el bus de datos y 24 bits en el bus de direcciones, así mismo trabaja con una frecuencia de reloj de 25 MHz, al dividir internamente su frecuencia de oscilador de 50 MHz en un factor de dos. Por lo tanto su ciclo de instrucción es 40ns y realiza 275 Millones de operaciones de punto fijo por segundo(MOPS), 50 Millones de operaciones de punto flotante por segundo(MFLOPS), 25 Millones de instrucciones por segundo(MIPS).

Además estos procesadores tienen un modo de direccionamiento especializado en realizar el proceso conocido como reversión de bits para el cálculo del algoritmo de la FFT. Con esto la ejecución de dicho algoritmo es muy rápida.

4 Resultados y conclusiones

Experimentalmente se encontró que un valor de $k=1.1$ en el filtro fue el óptimo para el mejoramiento en imágenes de buena calidad, y para imágenes de mala calidad $k=1.3$ presenta mejores resultados [5, 6, 2]. Este método presenta un problema en los bordes de cada bloque por lo se necesito un traslape de 26 píxeles entre cada bloque para eliminar este efecto [5, 6].

El filtro basado en la FFT se probó con 5 imágenes de diferentes personas, las cuales tienen diferentes tamaños y diferentes tipos de calidad. Los resultados del algoritmo en 4 de esas imágenes se muestran en la figura 3.

De los resultados mostrados se puede concluir que la eficiencia de la metodología propuesta para el filtrado y mejora de imágenes de huellas dactilares es grande. Se puede observar que para las imágenes de mala calidad los resultados de la metodología propuesta son muy buenos, en cambio con las imágenes de buena calidad los resultados son muy similares a los descritos en [7].

Con estos resultados el porcentaje de eficiencia obtenido en el AFAS es muy grande [2, 1].

Los resultados de los diferentes tiempos de ejecución de este algoritmo para las 5 personas en los diferentes lenguajes de programación y plataformas, se muestran en la tabla 1.

Cabe mencionar que para realizar la implementación del filtro en el ensamblador del DSP se implementaron las funciones de logaritmo natural y exponencial en el DSP, para esto se utilizó el código de las funciones que proporciona Texas Instruments [3].

En cuanto a los tiempos de ejecución del algoritmo en la PC se tiene la ejecución más rápida al programarse el algoritmo en lenguaje "C", en esta misma plataforma al programarse el algoritmo sobre Matlab los tiempos se incrementan considerablemente.

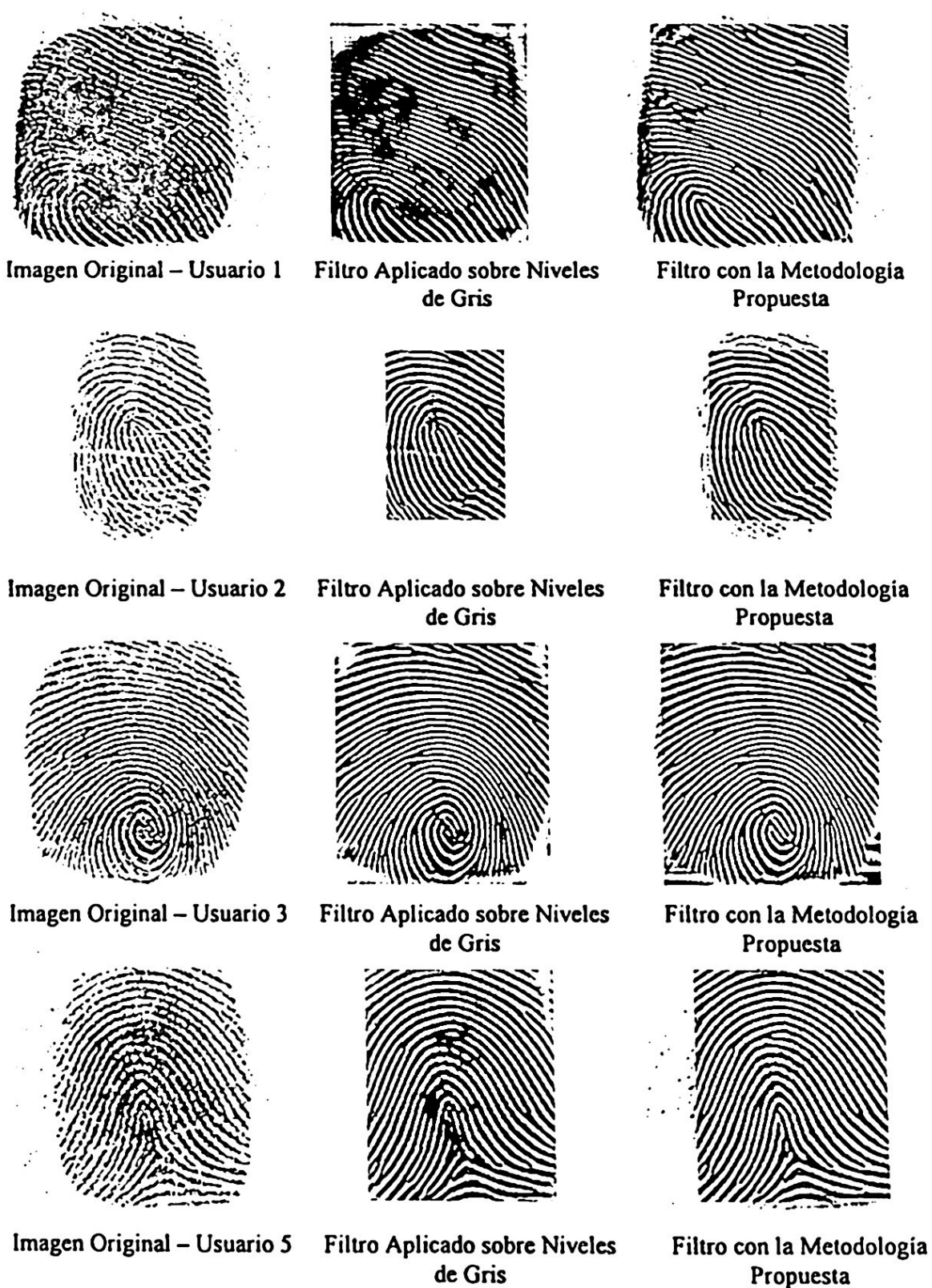


Fig. 3. Imágenes de prueba

Tabla 1. Tiempos de ejecución del filtro basado en la FFT

Imagen	Algoritmo en "C"	Algoritmo en el DSP	Algoritmo en Matlab
Usuario1	10 seg.	42 seg.	54 seg.
Usuario2	4 seg.	14 seg.	26 seg.
Usuario3	10 seg.	39 seg.	51 seg.
Usuario4	5 seg.	21 seg.	33 seg.
Usuario5	7 seg.	31 seg.	38 seg.

En el DSP los tiempos son mayores que en la PC, a pesar de la arquitectura de este procesador, lo posibles factores que ocasionan esto, son sin duda primeramente la velocidad de 25Mhz del DSP contra 550 Mhz de la PC y el uso de métodos numéricos para el cálculo y evaluación del logaritmo natural y la exponencial en el ensamblador del DSP, en la PC se hace uso del coprocesador matemático para el cálculo de estas funciones.

Referencias

- [1] V. H. Garcia, "Tesis de Maestría: Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos", CIC-IPN, México D.F., 2004.
- [2] V. H. Garcia, O. Espinosa, "Estrategias para la Autenticación de Huellas Dactilares", Congreso Internacional ANIEI, Zacatecas, Zacatecas, 2003.
- [3] Texas Instruments, TMS320C3X User's Guide. 2003.
- [4] Veridicom, FPS110 Data Sheet. 2003
- [5] V. H. Garcia, O. Espinosa, "Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos", Congreso Internacional CIC-INDI, México DF, 2001.
- [6] V. H. Garcia, "Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos Basado en el DSP TMS320C31", Congreso Internacional ELECTRO-2001, Chihuahua, Chihuahua, 2001
- [7] A.J. Willis and L. Myers, "A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingertips", Pattern Recognition, vol. 34, No 2, pp. 255-270, February 2001.
- [8] L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, Intelligent biometric techniques in fingerprint and face recognition, 1999.
- [9] Mardia, K.V., Baczkowski, A. J., Feng, "Statistical methods for automatic interpretation of digitally scanned finger prints", Pattern Recognition Letters. pp 1197-1203, 1997.
- [10] R. González, R. Woods, Digital Image Processing, 1996.
- [11] Sherlock, B.G., Monroe, D.M., and Millard, K., "Fingerprint Enhancement by Directional Fourier Filtering", IEEE Proceedings Vision, Image and Signal Processing, Vol. 141, No 2. pp 87-94. 1994.
- [12] B. Miller, "Vital signs of identity", IEEE Spectrum, 31(2), pp. 22-30, 1994.
- [13] O' Gorman, L. And Nickerson, J.V., "An approach to fingerprint filter design", Pattern Recognition, Vol. 22, No. 1, pp 29-38. 1989.
- [14] B.M. Mehre and B. Chatterjee, "Segmentation of fingerprint images A composite method", Pattern Recognition, vol. 22, pp. 381-385, 1989.