Reconocimiento de Firmas Manuscritas en Línea usando Alineamiento Dinámico en Tiempo y Descomposición por Ondoletas

Por

Lic. Juan Carlos Sánchez Díaz

Tesis sometida como requisito parcial para obtener el grado de

Maestría en Ciencias en la especialidad de Electrónica

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Supervisada por

Dr. Rogerio Adrian Enríquez Caldera

Dr. Juan Manuel Ramírez Cortés

©INAOE

Tonantzintla, Pue.

Mayo de 2010

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir y distribuir cualquier parte de este documento

RESUMEN

El crecimiento del comercio electrónico y de las transacciones bancarias personales o no personales requiere de un incremento en la seguridad de dichas operaciones. La clonación y robo de tarjetas de crédito es una práctica frecuente entre los defraudadores, que emplean la información presente en estos documentos para acceder a las cuentas bancarias de los afectados y cometer fraudes. Anualmente, este tipo de delitos representan pérdidas millonarias en todo el mundo. El uso de parámetros biométricos, tales como información de la firma del usuario en este tipo de transacciones incrementa la seguridad de los procesos.

En el desarrollo del presente proyecto de tesis, se presenta un método de reconocimiento de firmas haciendo uso de la técnica de Alineamiento Dinámico en tiempo modificado (DTW por sus siglas en inglés) y de los coeficientes de la Transformada "Wavelet" a un cierto nivel de aproximación. La técnica original DTW fue modificada por medio de la incorporación de información basada en puntos de apareamiento directo (*"Direct Matching Points",* DMP), con lo cual se obtuvo una mejora en la medida de similitud utilizada en el sistema. Las pruebas se realizaron a nivel de la firma completa (global) y a nivel de los trazos que la componen ("strokes"). Para tal motivo se diseñó e implementó un sistema de evaluación en la plataforma de MATLAB. A su vez es evaluado el desempeño del sistema haciendo uso de distintas funciones Wavelet, para determinar con cuál de ellas se presenta un mejor resultado, en función de las siguientes figuras de mérito: Razón de Falsos Rechazos (FRR por sus siglas en inglés), Razón de Falsas Aceptaciones (FAR por sus siglas en inglés) y Razón de Error Igual (ERR por sus siglas en inglés).

Durante el desarrollo de la investigación se pudo comprobar que el algoritmo DTW modificado mejoró los resultados en comparación con el algoritmo DTW original. También se pudo corroborar que haciendo uso de los coeficientes de la Transformada Wavelet es posible mejorar el desempeño del sistema con respecto del algoritmo DTW original, tanto en el número de reconocimientos exitosos que se logran, como en la reducción del tiempo de procesamiento necesario para realizar el análisis de cada firma, lo que permite, en principio, generar sistemas más efectivos y rápidos comparados con aquellos que emplean de manera exclusiva el algoritmo DTW para el reconocimiento de firmas manuscritas.

AGRADECIMIENTOS

Ilimitado tendría que ser el espacio destinado a agradecer a todos aquellos a los que me han tendido la mano en el desarrollo de este y muchos otros proyectos a lo largo de toda mi vida.

No podría dejar de mencionar, en primer lugar a mis padres, quienes siempre me han bridado su amor cariño y comprensión. Han sido fuente de inspiración y fortaleza hasta en los momentos más difíciles de mi vida y gracias a su apoyo siempre he podido salir adelante.

Un agradecimiento especial a mis hermanos, quienes siempre me han apoyado en el desarrollo de mis proyectos.

Y no podría dejar de mencionar al resto de mi familia, quienes también me han enseñado el verdadero valor de la vida.

A mis asesores, el Dr. Rogerio Enríquez Caldera quien me ha brindado la preparación académica necesaria para desarrollar este proyecto así como la paciencia para mejorar el contenido de esta tesis, así como un agradecimiento muy especial al Dr. Juan Manuel Ramírez Cortés cuyos valiosos consejos y enseñanzas han colaborado en forma muy activa a la estructuración de este proyecto. La sabiduría que he obtenido de las enseñanzas de ambos serán sin duda alguna una herramienta que me será de utilidad el resto de mi vida.

Sin lugar a dudas, no puedo dejar de agradecer a todos mis maestros, cuyas enseñanzas son el bloque de construcción de los conocimientos que he podido adquirir a lo largo de mi formación académica. También no puedo omitir de esta mención a mis amigos, quienes amablemente han contribuido con sus comentarios y observaciones

Hago un reconocimiento muy especial al INAOE quien ha fungido como alma mater de mi formación profesional y al CONACYT cuyo apoyo económico ha sufragado esta investigación.

A todos ellos dedico el presente proyecto.

Tabla de contenidos

CAPÍTULO 1. INTRODUCCIÓN 8
1.1 Características generales de los sistemas de verificación 8
1.2 Trabajos realizados 10
1.3 Justificación 11
1.4 Objetivos 12
1.4.1 Objetivo general 13
1.4.2 Objetivos particulares 13
1.5 Metodología 13

CAPITULO 2. FUNDAMENTOS TEORICOS 16
2.1 Normalización de secuencias de tiempo y alineamiento 16
2.2 Alineamiento dinámico de tiempo (DTW) 19
2.3 Transformada Wavelet 26
2.3.1 Breve historia de la transformada wavelet 26
2.3.2 Transformación Wavelet 27
2.3.3 Relación con los esquemas de filtrado sub banda 30

CAPITULO 3 DESARROLLO DEL SISTEMA 34
3.1 Esquema general del sistema de evaluación de firmas realizado 34
3.2 Programa de gestión para adquisición de firmas 35
3.2.1 Introducción 35
3.2.2 Justificación 35
3.2.3 Tableta digitalizadora 36
3.2.4 Software desarrollado 37
3.3 Sistema de Procesamiento 39
3.3.1 Introducción 39
3.3.2 Preprocesamiento 39
3.3.3 Extractor de características 41
3.3.4 Bloque de plantillas 49

3.3.5 Bloque Verificador 50
3.4 Tipos de análisis implementados en el sistema para la realización de pruebas. 51
3.4.1 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma completa. 51
3.4.2 Comparación de los coeficientes de aproximación de firma plantilla y firma de prueba considerando la firma completa. 51
3.4.3 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma dividida en trazos. 52
3.4.4 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma dividida en trazos. 53
3.5 Herramienta de procesamiento desarrollada en MATLAB 53

CAPITULO 4 RESULTADOS Y CONCLUSIONES 54
4.1 Resultados 54
4.2 Conclusiones 62
4.3 Trabajo futuro 63

APENDICE 64
A1 Programación dinámica 64
A2 Particularidades de la aplicación de adquisición de firmas 70
A3 Descripción de la interface gráfica creada con MATLAB 72
A3.1 Tareas implementadas 72

REFERENCIAS 81

LISTA DE FIGURAS

Figura 1 Diagrama a bloques para un sistema de reconocimiento de firmas 8
Figura 2. Trayectoria resultante de la unión de los puntos de dos funciones de “warping” 18
Figura 3. Construcción de la matriz de distancias para el ejemplo de DTW 23
Figura 4.Detalle de matriz de distancias para el ejemplo de DTW 23
Figura 5. Inicialización de la matriz de distancia acumulada para el ejemplo de DTW 24
Figura 6. Matriz de distancia acumulada para el ejemplo de DTW 25
Figura 7. Alineamiento sugerido para las secuencias de ejemplo 25
Figura 8. Resultado de aplicar DTW a dos secuencias de tiempo con gran similitud 26
Figura 9. Estructura jerárquica en la obtención de coeficientes wavelet 32
Figura 10. Obtención de coeficientes wavelet mediante banco de filtros 33
Figura 11. Diagrama a bloques del sistema propuesto 34
Figura 12. Tableta digitalizadora empleada 37
Figura 13. Pantalla principal del programa de adquisición de firmas 37
Figura 14. Diagrama de bloques del extractor de características 41
Figura 15. Diagonal de trayectoria DTW en secuencias iguales 43
Figura 16. Ilustración gráfica de los DMP's 44
Figura 17. Diagrama de flujo del algoritmo homologador y evaluador de trazos 49
Figura 18. Bloque extractor de características para prueba tipo 1 51
Figura 19. Bloque extractor de características para prueba tipo 2 52
Figura 20. Bloque extractor de características para prueba tipo 3 52
Figura 21. Bloque extractor de características para prueba tipo 4 53
Figura 22. Histograma del número de trazos de la muestra 55
Figura 23. Histograma del tiempo de firmado de la muestra 55
Figura 24 Gráfica FAR/FRR del empleo de las funciones X,Y vs Magnitud/Fase 56
Figura 25 Gráfica FAR/FRR algoritmo DTW modificado vs algoritmo DTW normal 57
Figura 26 Gráfica comparativa FAR/FRR para distintos tipos de análisis implementados 58
Figura 27 Gráfica comparativa FAR/FRR análisis tipo 2 para diferente DWT 60
Figura 28 Gráfica comparativa análisis tipo 2 Wavelet Coiflet3 distintos niveles de aproximación 61
Figura 29. Ejemplo del problema de decisión secuencial asíncrona 64
Figura 30. Estructura de celosía para el problema de decisión secuencial síncrona 65
Figura 31. Solución al problema de ejemplo de decisión secuencial síncrona 66
Figura 32. Interface gráfica desarrollada 72
Figura 33. Ventana correspondiente a la tarea de adquisición 73
Figura 34. Ventana de la tarea de obtención de datos 74
Figura 35. Ventana de la tarea para generación de plantillas 75
Figura 36. Ventana de la tarea de ajuste de configuraciones 76
Figura 37. Ventana de la tarea de análisis global 78
Figura 38. Ventana de la tarea de verificación 79
Figura 39. Ventana de la tarea de reconocimiento 80

LISTA DE TABLAS

Tabla 1. Características del espacio muestral 54
Tabla 2. Resultados EER del empleo de las funciones X,Y vs Magnitud/Fase 57
Tabla 3 Resultados EER algoritmo DTW modificado vs DTW normal 58
Tabla 4. Resultados EER para distintos tipos de análisis implementados..... 59
Tabla 5 Resultados EER para distintas DWT análisis tipo 2..... 60
Tabla 6 EER para análisis tipo 2 Wavelet Coiflet3 distintos niveles de aproximación. 61

CAPÍTULO 1. INTRODUCCIÓN

1.1 Características generales de los sistemas de verificación

Para la siguiente sección y salvo que se indique lo contrario, las referencias han sido tomadas de [2].

La firma manuscrita ha sido uno de los métodos más empleados en la validación de documentos institucionales a lo largo de la historia. Pese a la existencia de nuevas tecnologías en la obtención de parámetros biométricos [1], el estudio de la firma manuscrita ha sido considerado con nuevos intereses, debido a su facilidad de adquisición, amplia aceptación y, en parte al crecimiento del uso de internet. Los esfuerzos de estos estudios están enfocados en facilitar la integración de las tecnologías de verificación de firmas a los equipos estándar, tales como computadoras, teléfonos celulares o asistentes personales digitales (PDA's por sus siglas en inglés), con el propósito de brindar soluciones para una gran variedad de aplicaciones comerciales como transacciones bancarias, seguridad de documentos médicos, seguridad de identidad, comercio electrónico, etc.

El desarrollo de sistemas para el reconocimiento de firmas manuscritas data de 1965 [1].

Se puede dividir un sistema de reconocimiento de firmas manuscritas de acuerdo con el siguiente diagrama de bloques:

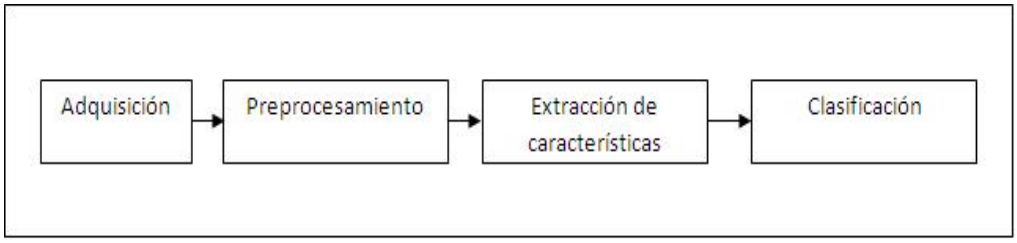

Figura 1 Diagrama a bloques para un sistema de reconocimiento de firmas

En base al método de adquisición de las firmas manuscritas, se pueden identificar dos categorías de sistemas para el reconocimiento: estático (“offline”) y dinámico (“online”). En el primer caso la firma es representada como una imagen en escala de grises, mientras que en el segundo un dispositivo genera señales electrónicas representativas del proceso de firmado mientras se realiza la firma.

En la etapa de preprocesamiento, se enriquecen los datos adquiridos haciendo uso de algoritmos de procesamiento digital de señales.

Para el caso de la etapa de extracción de características suelen emplearse dos tipos: funciones o parámetros.

Cuando se usan funciones, la firma se caracteriza en términos de una función de tiempo discreto cuyas muestras representan un conjunto de características. En el caso de parámetros, la firma es caracterizada por un vector de elementos, cada uno de ellos representativo de algún parámetro de la firma. Los parámetros pueden ser locales o globales. Los parámetros globales son obtenidos de la firma completa, mientras que en el caso de los locales son extraídos de partes específicas de la misma.

Entre las funciones con características más consistentes, para el caso del reconocimiento de firmas “on line” se encuentran: la posición, velocidad y la inclinación del bolígrafo.

En ocasiones, los parámetros obtenidos son coeficientes derivados de herramientas matemáticas. Entre estas se encuentran, la transformada de Fourier, transformada Hadamard, transformada coseno, transformada Wavelet y la transformada Radom.

Finalmente, el bloque de clasificación es el encargado de evaluar la autenticidad de la firma bajo prueba cotejando sus características en contra de las que se encuentran almacenadas en una base de datos desarrollada durante la fase de entrenamiento.

Una de las técnicas empleadas en el proceso de verificación cuando se emplean funciones como características es un algoritmo derivado de la programación dinámica llamado Distorsión Dinámica en Tiempo (“*Dynamic Time Warping*”, DTW). El DTW permite la compresión o la expansión del eje de tiempo de dos secuencias representativas de la firma para obtener el mínimo valor de disimilitud entre ellas.

Aunque la superioridad del DTW no ha sido probada ante otras técnicas de comparación, ha sido ampliamente usado en diversas investigaciones.

El algoritmo de DTW también proporciona información sobre el alineamiento realizado sobre las secuencias de tiempo sobre las que ha operado [11], por lo que es posible analizar las disimilitudes temporales globales y locales de dicha secuencia.

Aunque el DTW posee muchas ventajas tiene el inconveniente de emplear mucho tiempo de procesamiento debido a que son necesarias poco más de NxM (donde M y N son las longitudes de las secuencias de entrada) cálculos para completar el algoritmo [11].

En un sistema de verificación, la firma de entrada es comparada contra la firma de un usuario almacenada como plantilla, para determinar su autenticidad, es decir, dicho usuario reclama su identidad mediante una firma y el sistema tiene la función de brindar una decisión. Para el caso de los sistemas de reconocimiento, la firma de entrada es comparada contra todas las plantillas almacenadas en el sistema para determinar que usuario enrolado ha ingresado la firma y en el caso de no existir ninguna coincidencia la firma se determina como no auténtica.

La importancia de este último aspecto radica en el hecho de que si es necesario emplear el algoritmo DTW para un sistema de reconocimiento de firmas, es preferible que lo haga haciendo uso de parámetros de la firma y no la firma como tal, con el propósito de reducir el número de datos presentes en cada secuencia y en consecuencia, el tiempo de procesamiento. Estos parámetros deben ser altamente representativos de la firma con el objeto de que el sistema sufra un mínimo impacto en su desempeño con respecto al logrado haciendo uso de la información de la firma completa.

1.2 Trabajos realizados

Alrededor del mundo se están realizando múltiples esfuerzos por lograr sistemas para el reconocimiento y adquisición de firmas manuscritas de forma automática haciendo uso de diversas técnicas. Algunos de los trabajos más sobresalientes relacionados con el empleo de la técnica de DTW y DWT se mencionan en los siguientes párrafos.

L. Bovino, en [4], emplea la información de posición, velocidad y aceleración en combinación con el algoritmo de DTW y un verificador multiexperto, logrando una taza de EER del 0.4%, entrenando al sistema con 45 firmas (3 firmas de 15 firmantes) y probando el sistema con 750 firmas genuinas (50 firmas de 15 firmantes) y 750 firmas falsas (50

firmas de 15 firmantes). En el trabajo de V. Di Lecce [17] se obtienen una FAR del 0.55% y una FRR del 3.2% haciendo uso de características basadas en la forma de la firma empleando como verificador el algoritmo de DTW y un sistema multiexperto por voto mayoritario. La base de datos tiene las mismas características que la empleada por L. Bovino en [4]. K. Huang y H.yan en [5] hacen uso de la información de velocidad y presión contenida en la firma, empleando la técnica de DTW obteniendo una EER de 4% para una base de datos consistente en 4600 firmas. I. Nakanishi [3] emplea información obtenida a partir de los coeficientes de detalle de la transformada wavelet y un proceso de verificación adaptivo obteniendo un 4% de EER haciendo uso de 20 firmas de 4 firmantes en la etapa de entrenamiento, 98 firmas genuinas de 4 firmantes y 200 firmas falsas de 5 firmantes en la etapa de prueba. En el trabajo de Berrin Yanikoglu y Alisher Kholmatov [18] se emplea una fusión de la técnica de DTW en combinación con descriptores de Fourier sobre una base de datos estándar denominada MCYT-100 que cuenta con 5000 firmas 2500 genuinas y 2500 falsas de 100 firmantes obteniendo un EER del 7.22%.

1.3 Justificación

La firma manuscrita sigue siendo hoy en día uno de los métodos de validación de documentos más empleado en el mundo.

En las transferencias bancarias personales, la firma es uno de los métodos empleados en el protocolo de seguridad al realizar pagos con tarjetas de crédito y débito [6].

El comercio electrónico se ha convertido en una actividad común y popular en el mundo. Tan sólo en 2008 las ventas en línea de productos alcanzaron los 204 mil millones de dólares [7]. El pago de este tipo de compras a través de internet suele hacerse mayormente por medio de tarjetas de crédito o debito.

Cuando se efectúa un pago, es necesario ingresar el número de la tarjeta de crédito y el código de seguridad de la misma, además de información adicional del tarjetahabiente (generalmente datos relacionados con el código postal de la zona en la que reside). Por si solos, estos datos de seguridad son relativamente fáciles de obtener y falsificar, por lo que personas no autorizadas pueden hacer uso de esta información para cometer fraudes.

Otra práctica común que se ha extendido como una forma de fraude es denominada “skimming”. Este tipo de fraude consiste en la clonación de la tarjeta del usuario mediante un dispositivo conocido como “skimmer”. En lugares en los cuales la tarjeta es separada de la vista de su propietario (restaurantes, bares, etc), el defraudador puede hacer uso de este dispositivo para obtener la información contenida en la banda magnética de la tarjeta y posteriormente emplear esta información para clonarla. Paralelamente, se puede obtener la información correspondiente al número de tarjeta y códigos de seguridad, para usarlos posteriormente en transacciones electrónicas.

Otra forma común de fraude sucede cuando una tarjeta de crédito es extraviada o robada. El defraudador puede hacer uso de ella en establecimientos comerciales para realizar pagos mientras el dueño la reporta al banco. En este tipo de establecimientos sólo se requiere que el usuario firme el comprobante de pago y presente una identificación, acciones que no son una tarea difícil para un defraudador experimentado.

El fraude con tarjetas de crédito representa un problema grave a nivel mundial. Se calcula que tan sólo en el reino unido en 2006 este tipo de fraudes tuvieron un costo de 535 millones de libras esterlinas [8], es decir unos 750 a 830 millones de dólares al tipo de cambio en esa época.

Para México, la situación no es mucho menos grave, ya que en 2007 se reportaron fraudes con tarjetas bancarias por 100 millones de dólares, 35 millones de usuarios habían reportado clonación de sus tarjetas y según la Asociación Nacional de tiendas de Autoservicio (ANTAD), 7 millones de tarjetas eran susceptibles de clonación [9].

Aunque en principio parece que equipar a comercios y usuarios con sistemas digitalizadores de firmas resulta una opción costosa, éstos se están volviendo cada vez más populares y económicos, ya que son cada vez más empleados con el propósito de generar recibos electrónicos. De esta forma un sistema de verificación que permita detectar la falsificación de firmas al momento de realizar pagos y transferencias electrónicas podría ayudar a disminuir y posiblemente terminar con este tipo de fraudes.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar e implementar un sistema que permita la verificación y reconocimiento de firmas manuscritas haciendo uso de la transformada wavelet y DTW.

1.4.2 Objetivos particulares

- Elaborar un software que permita la gestión en el proceso de adquisición de firmas haciendo uso de una tableta digitalizadora.
- Obtener 20 firmas de 50 firmantes para evaluar el funcionamiento y desempeño del sistema.
- Modificar el algoritmo original de DTW, con el propósito de obtener información sobre la disimilitud de las secuencias de entrada, alineamiento de las mismas y puntos de estabilidad global.
- Elaborar un sistema de evaluación que permita organizar los archivos generados por el gestor del proceso de adquisición de firmas, generar la base de datos para el proceso de verificación, obtener coeficientes de transformación Wavelet haciendo uso de cualquier tipo de Wavelet madre y nivel de aproximación, procesar las firmas tanto a nivel global como local, entregar y salvar resultados de los análisis.
- Realizar diversas pruebas con el sistema haciendo uso de la modificación de la técnica DTW propuesta a nivel de función y también a nivel de parámetros (coeficientes Wavelet), para distintos valores con el propósito de buscar la implementación que genere mejores resultados.

1.5 Metodología

<u>Método empleado en el desarrollo del sistema de adquisición:</u>

1. Investigación sobre el empleo de tabletas digitalizadoras para aplicaciones en el sistema operativo Microsoft Windows XP.

2. Investigación sobre el desarrollo de aplicaciones con tabletas digitalizadoras haciendo uso de Microsoft Visual Basic.
3. Desarrollo de la aplicación para la adquisición de firmas haciendo uso de una tableta digitalizadora.

Método empleado en el desarrollo del sistema de procesamiento:

1. Investigación sobre programación dinámica.
2. Investigación sobre algoritmo de DTW.
3. Pruebas con el algoritmo original de DTW [10].
4. Investigación sobre puntos de correspondencia uno a uno en el algoritmo DTW (DMP).
5. Implementación de algoritmo DTW modificado.
6. Pruebas de algoritmo DTW modificado.
7. Investigación sobre Transformada Discreta Wavelet (DWT).
8. Pruebas con Transformada Discreta Wavelet a funciones experimentales.
9. Desarrollo del bloque de preprocesamiento de datos.
10. Desarrollo del sistema de evaluación.

Método empleado en la evaluación del sistema:

1. Selección de firmas a emplear en el proceso y el número de firmantes. En el caso de las pruebas elaboradas, se hizo uso de 20 firmas de 50 firmantes. Aunque las bases de datos de firma empleadas en los trabajos mencionados presentan variaciones entre ellas, se ha determinado este número de firmas en base a una competencia denominada SVC2004, cuya base de datos consiste de 40 firmantes y 20 firmas de cada uno, además de 20 falsificaciones profesionales de cada firmante.
2. Selección de las firmas candidatas para la elaboración de plantillas. En el caso de las pruebas realizadas se trató de las firmas marcadas con los números 2, 7 11, 13 y 18.
3. Prueba para determinar la FRR. Se comparan la firma plantilla de un firmante dado en contra del resto de sus firmas.

4. Prueba para determinar la FAR en falsificaciones entrenadas (“skilled forgeries”). Se comparan la firma plantilla de un firmante dado en contra de firmas falsificadas entrenadas del firmante correspondiente a dicha plantilla. Las firmas falsificadas entrenadas son elaboradas por firmantes a los que se les ha dado una imagen de la firma y la reproducen en la tableta digitalizadora. Sólo las firmas más parecidas a la imagen son empleadas.
5. Prueba de tiempo de proceso. Se realiza contando el tiempo que cada procesamiento de firma tarda en efectuarse en una misma computadora.

CAPITULO 2. FUNDAMENTOS TEORICOS

2.1 Normalización de secuencias de tiempo y alineamiento

La presente sección es un resumen de los aspectos más relevantes de la información encontrada en [11].

El resultado de la digitalización de una firma es un conjunto de datos que dependen del tiempo. Debido a su naturaleza, es muy poco probable que una firma tenga exactamente el mismo número de muestras que otra, aunque se trate del mismo firmante. Debido a esto es necesario hacer que las dos secuencias correspondientes a las firmas en el eje de tiempo tengan el mismo número de muestras antes de poder realizar una comparación, a este proceso se le denomina normalización del eje del tiempo. La solución más simple para el problema de normalización temporal y alineamiento es una técnica de normalización lineal de tiempo. En esta situación las disimilitud entre dos secuencias de tiempo X e Y puede definirse simplemente como:

$$d(X,Y) = \sum_{i_x=1}^{T_x} d\left(i_x, i_y\right) \qquad (2.1.1)$$

donde i_x, i_y representan los índices de tiempo de las secuencias X e Y respectivamente y satisfacen:

$$i_y = \frac{T_y}{T_x}\, i_x \qquad (2.1.2)$$

Cabe hacer notar que debido a que estos índices son enteros, es necesario aplicar alguna técnica de redondeo. La función $d\left(i_x, i_y\right)$ indicada en la sumatoria es una medida de disimilitud entre los puntos i_x, i_y de las secuencias de tiempo, como puede ser simplemente la diferencia en magnitud entre los valores de dichos puntos, elevada al cuadrado (distancia euclidiana) u otro tipo de medición.

Esta normalización para el eje del tiempo sería una solución adecuada para resolver el problema de comparar dos firmas, sin embargo, el alineamiento producido por la técnica implícitamente supone que la tasa de variación de los puntos que conforman la secuencia de tiempo es proporcional a la duración de la misma, lo que en general no representa un buen modelo de las firmas debido a que durante el proceso de firmado se suelen tener variaciones temporales que exhiben una compresión o expansión temporal de los puntos que la conforman.

Un esquema más general del alineamiento y normalización del eje de tiempo consiste en el uso de dos funciones de "warping" ϕ_x, y ϕ_y, que relacionan los índices de las dos secuencias i_x e i_y a un eje "normal" común de tiempo *k*. De esta forma:

$$i_x = \phi_x(k) \qquad k = 1,2, \dots T \qquad (2.1.3)$$

y

$$i_y = \phi_y(k) \qquad k = 1,2, \dots T \qquad (2.1.4)$$

Donde T es el número total de puntos existentes en las secuencias de tiempo normalizadas. En base a estas funciones se puede definir una medida de disimilitud global $d(X,Y)$ como la distorsión acumulada de las funciones normalizadas:

$$d(X,Y) = \sum_{i_x=1}^{T} \frac{d\left(\phi_x(k), \phi_y(k)\right) m(k)}{M_\phi} \qquad (2.1.5)$$

Donde $d\left(\phi_x(k), \phi_y(k)\right)$ es una distorsión de tiempo corto definida para $X(\phi_x(k))$ y $Y(\phi_y(k))$, *m(k)* es una ponderación no negativa de trayectoria y M_ϕ es un factor de normalización de trayectoria. Se menciona la palabra trayectoria porque al relacionar las funciones $\phi_x(k)$ y $\phi_y(k)$ en un plano para todos los puntos $k = 1,2, \dots T$ se obtiene una trayectoria sobre una rejilla, que inicia en el primer elemento de tiempo de ambas secuencias y finaliza después de T movimientos en la esquina contraria de la misma $\phi_x(k) = T_x$ y $\phi_y(k) = T_x$, donde T_x y T_y representan la última muestra de las secuencias X e Y respectivamente. La figura 2 ilustra el concepto.

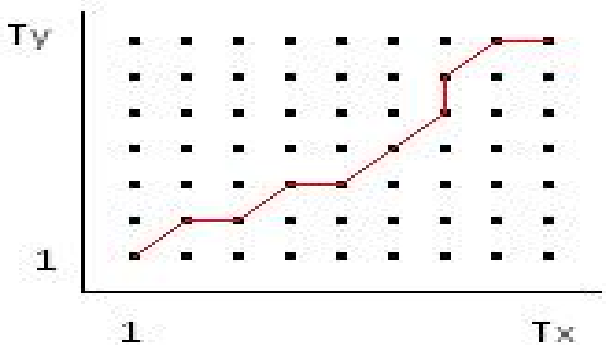

Figura 2. Trayectoria resultante de la unión de los puntos de dos funciones de "warping"

Formalmente, el conjunto de pares $\phi = (\phi_x, \phi_y)$ es llamado trayectoria. Existen muchas posibles trayectorias para normalizar un par de secuencias de tiempo dadas, así que la pregunta sería ¿Cuál de todas las posibles trayectorias se debe elegir? Una elección lógica y popular sería definir la disimilitud de ambas secuencias de tiempo como el mínimo de los valores de disimilitud sobre todas las posibles trayectorias. Esto es:

$$d(X,Y) \triangleq \min_{\phi} d_{\phi}(X,Y) \qquad (2.1.6)$$

donde ϕ es una posible trayectoria de "warping" que normaliza las secuencias X y Y.

El punto básico es que encontrar el "mejor" alineamiento entre un par de secuencias es equivalente a encontrar la "mejor" trayectoria a través de la rejilla que mapea las características de una secuencia sobre la otra.

La forma de la ecuación (2.1.6) indica una optimización y que las técnicas de programación dinámica pueden ser aplicables en el proceso de solución.

2.2 Alineamiento dinámico de tiempo (DTW)

Al igual que en la sección anterior, la información ha sido obtenida de [11] salvo que se indique lo contrario.

Restricciones de la normalización en el tiempo

El objetivo del empleo de la programación dinámica en los algoritmos de normalización y alineamiento temporal es poder minimizar las disimilitudes entre dos secuencias de tiempo dadas.

Los algoritmos empleados en la solución de los problemas de decisión secuencial síncrona y asíncrona de la programación dinámica son empleados también para determinar las funciones de “warping” empleadas en dicha normalización, sin embargo es necesario imponer restricciones a estos algoritmos con el propósito de que satisfagan ciertas características que el eje temporal impone. Estas condiciones son:

- Restricciones de puntos terminales
- Condiciones de monotonía
- Restricciones de continuidad local
- Restricciones de trayectoria global
- Pesos de pendiente

1. Restricciones de puntos terminales

Para la normalización en tiempo, los puntos terminales, es decir los límites fijos de las secuencias de tiempo, imponen una serie de restricciones en las funciones de “warping”. Dichas restricciones son:

Punto inicial: $\phi_x(1) = 1, \qquad \phi_y(1) = 1$

Punto final: $\phi_x(T) = T_x, \qquad \phi_y(T) = T_y$

2. Condiciones de monotonía

Para mantener el orden temporal de las secuencias mientras se realiza la normalización de tiempo se deben satisfacer las siguientes condiciones:

$$\phi_x(k+1) \geq \phi_x(k)$$
$$\phi_y(k+1) \geq \phi_y(k)$$

Estas condiciones evitan un alineamiento temporal inverso en las funciones resultantes.

3. Restricciones de continuidad local

Esta restricción se incorpora para asegurar un alineamiento temporal propio, sin dar lugar a pérdidas potenciales de información. Estas pueden tomar muchas formas. Un ejemplo propuesto por Saoke y Chiba [21] es:

$$\phi_x(k+1) - \phi_x(k) \leq 1$$
$$\phi_y(k+1) - \phi_y(k) \leq 1$$

Durante el desarrollo de la tesis, ésta ha sido la restricción elegida. Para profundizar más en otros tipos de restricciones se puede consultar la referencia [11].

4. Restricciones de trayectoria global

Esta condición es una consecuencia de la restricción de continuidad local, e indica que ciertos puntos del plano (i_x, i_y) quedarán excluidos de la trayectoria óptima.

5. Pesos de las pendientes

Es posible modificar la política de la programación dinámica seleccionando los pesos de los posibles caminos a seguir en desplazamientos de un solo paso. En una escala local también es posible asociar una función de peso relacionada con la restricción de trayectoria de manera que se pueda aplicar una preferencia entre estas trayectorias. Las funciones mencionadas anteriormente son conocidas como funciones de peso de las pendientes.

Considerando los métodos y características de la programación dinámica en combinación con las restricciones mencionadas para el caso del eje temporal, se puede describir una técnica para lograr el alineamiento de dos secuencias de tiempo. Esta técnica es conocida como Alineamiento dinámico de tiempo (DTW por sus siglas en inglés).

Para esto reescribimos la ecuación (2.1.5) en términos de T_xy T_y haciendo uso de de las restricciones mencionadas. De esta forma obtenemos:

$$M_{\phi} d(X,Y) \triangleq D\left(T_x, T_y\right) = \min_{\phi_x,\phi_y} \sum_{k=1}^{T} d\left(\phi_x(k), \phi_y(k)\right) m(k) \qquad (2.2.1)$$

La distorsión parcial mínima acumulada entre los puntos (1,1) e (i_x, i_y) es:

$$D\left(i_x, i_y\right) \triangleq \min_{i'_x, i'_y, T'} \sum_{k=1}^{T'} d\left(\phi_x(k), \phi_y(k)\right) m(k) \qquad (2.2.2)$$

donde:

$$\phi_x(T') = i_x \qquad \phi_y(T') = i_y$$

El programa de recursión dinámica se convierte entonces en:

$$D(i_x, i_y) = \min_{(i'_x, i'_y)} \left[D(i'_x, i'_y) + \varsigma\left((i'_x, i'_y), (i_x, i_y) \right) \right] \qquad (2.2.3)$$

donde ς representa la función ponderada acumulada de distorsión (distancia local) entre los puntos (i'_x, i'_y) y (i_x, i_y), denotada como:

$$\varsigma\left((i'_x, i'_y), (i_x, i_y) \right) = \sum_{l=0}^{Ls} d\left(\phi_x(T' - l), \phi_y(T' - l) \right) m(T' - l) \qquad (2.2.4)$$

Con Ls el número de movimientos en el trayecto desde (i'_x, i'_y) hasta (i_x, i_y) de las funciones ϕ_x y ϕ_y. m es el peso de la pendiente descrito por la restricción 5. También queda implícito el hecho de que:

$$\phi_x(T' - Ls) = i'_x$$
$$\phi_y(T' - Ls) = i'_y$$

Llegado este punto podemos resumir la implementación del algoritmo DTW en los siguientes pasos:

1. Inicialización

$$D_A(1,1) = d(1,1)m(1)$$

2. Recursión

   Para $1 \leq i_x \leq T_x$, $1 \leq i_y \leq T_y$ de forma que i_x, e i_y se encuentren dentro de una posible trayectoria local permitida, computar:

$$D_A(i_x, i_y) = \min_{(i'_x, i'_y)} \left[D_A(i'_x, i'_y) + \varsigma\left((i'_x, i'_y), (i_x, i_y) \right) \right]$$

donde $\varsigma\left((i'_x, i'_y), (i_x, i_y)\right)$ ha sido definida en la ecuación (2.2.4).

3. Finalización

$$d(X,Y) = \frac{D_A(T_x, T_y)}{M_\phi} \qquad (2.2.5)$$

Un ejemplo de la aplicación de esta técnica se describe a continuación:

Considerando dos secuencias de tiempo $X_y = \{1, 2, 3, 4\}$ y $X_x = \{1, 2, 3, 3, 4\}$. Definimos la matriz de distancias d_m, que contiene las distancias entre todos los puntos de cada secuencia. La figura siguiente ilustra la forma de construir dicha matriz:

4	3	2	1	1	0
3	2	1	0	0	1
2	1	0	1	1	2
1	0	1	2	2	3
	1	2	3	3	4

Figura 3. Construcción de la matriz de distancias para el ejemplo de DTW

La matriz se muestra en color verde y consiste, para este ejemplo, del valor absoluto de las diferencias entre los puntos de la secuencia. Esta matriz ilustra las distancias entre los puntos y servirá para generar las funciones de costo para cada paso. La figura 4 muestra en detalle la matriz de distancias:

$d_m =$

3	2	1	1	0
2	1	0	0	1
1	0	1	1	2
0	1	2	2	3

Figura 4.Detalle de matriz de distancias para el ejemplo de DTW

Posteriormente se procede a la creación de la matriz D_A.- Si definimos la restricción local de forma que únicamente es posible moverse hacia arriba, a la derecha o en forma diagonal, se tendría la siguiente ecuación recursiva:

$$D_A(i_x, i_y) = \min \begin{Bmatrix} D_A(i_x - 1, i_y) + d_m(i_x, i_y) \\ D_A(i_x, i_y - 1) + d_m(i_x, i_y) \\ D_A(i_x - 1, i_y - 1) + d_m(i_x, i_y) \end{Bmatrix}$$

Comenzamos la construcción de la matriz D_A con la observación de que los valores de la fila inferior y la primera columna son simplemente la suma de los valores a lo largo de la fila inferior y la primera columna de d_m. La figura 5 ilustra este proceso:

$D_a =$

6				
3				
1				
0	1	3	5	8

Figura 5. Inicialización de la matriz de distancia acumulada para el ejemplo de DTW

Una vez hecho lo anterior, se aplica la ecuación recursiva anterior hasta completar la matriz. La figura 6 muestra la matriz D_A concluida, el valor del cuadro sombreado es el valor de disimilitud estimado:

$D_a =$

6	3	1	1	0
3	1	0	0	1
1	0	2	3	5
0	1	3	5	8

Figura 6. Matriz de distancia acumulada para el ejemplo de DTW

Se observa que $D_a(T_x, T_y) = 0$. Para encontrar las funciones de "Warping" ϕ_x y ϕ_y, Aplicamos el mismo criterio comenzando desde $D_a(T_x, T_y)$ hasta llegar a $D_a(1,1)$ ("backtracking"). Haciendo esto encontramos:

$$\phi_y = \{1,2,3,3,4\}$$
$$\phi_x = \{1,2,3,4,5\}$$

En la figura 7 se muestra el alineamiento sugerido por el DTW:

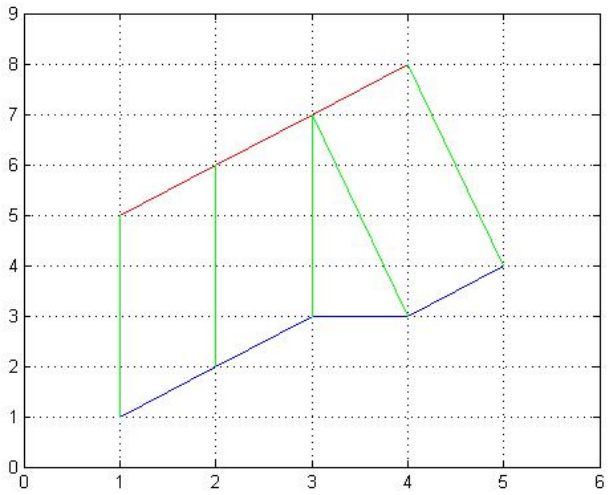

Figura 7. Alineamiento sugerido para las secuencias de ejemplo

Las secuencias se han desplazado en el eje y con el propósito de mostrar el alineamiento. Las líneas verdes indican los puntos que pueden ser relacionados para lograr una disimilitud mínima.

A manera de ilustración, se aplica el proceso descrito a un par de funciones de tiempo representativas de una firma. El resultado se puede observar en la figura 8:

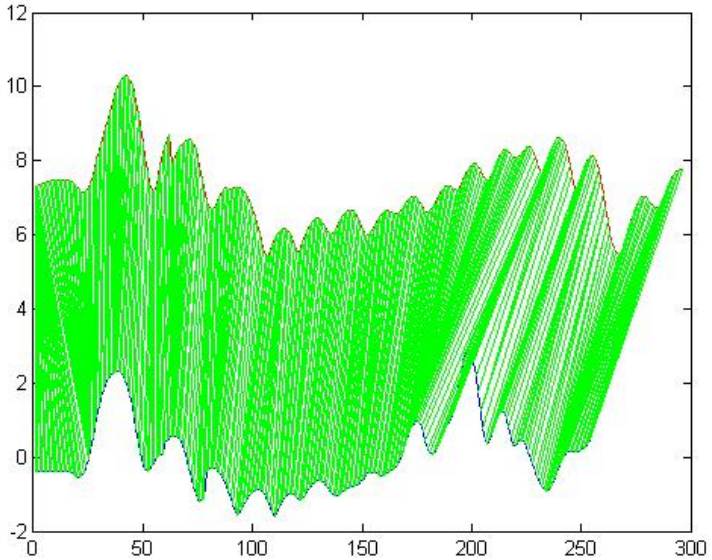

Figura 8. Resultado de aplicar DTW a dos secuencias de tiempo con gran similitud

2.3 Transformada Wavelet

2.3.1 Breve historia de la transformada wavelet

La teoría de wavelets ha provisto de un nuevo método para descomponer una función o una señal. La presente sección es un resumen con los aspectos más importantes encontrados en [12].

El concepto de “wavelets” u ondoletas comenzó a aparecer en la literatura sólo hasta principios de los años 80.

En muchas aplicaciones, especialmente en el análisis de tiempo – frecuencia de una señal, la transformada de Fourier de una señal no contiene ninguna información local. Para eliminar esta debilidad en 1946 Dennis Gabor, un físico e ingeniero Británico - Húngaro introdujo por primera vez la Transformada de Fourier en ventanas o transformada de Fourier de tiempo corto. La diferencia remarcable de este método es el

aspecto local del análisis de la transformada de Fourier, cuya resolución en tiempo es igual al tamaño de la ventana.

Sin embargo, las “wavelets” de Gabor sufrían de serias dificultades algorítmicas, que fueron resueltas exitosamente por Henrique Malvar en 1987.

El primer logro mayor en el campo del análisis wavelet fue logrado por Daubechies, Grossmann y Meyer en 1986 con la construcción de expansiones wavelet no ortogonales.

Meyer(1986) y Mallat (1988) permitieron que las bases de las wavelets ortogonales pudieran ser construidas sistemáticamente de un formalismo general. Su colaboración culminó con un notable descubrimiento hecho por Mallat en 1989, el análisis multiresolución.

Inspirada por el trabajo de Meyer, Ingrid Daubechies (1988) hizo una contribución notable a la teoría Wavelet mediante la construcción de wavelets ortonormales de soporte compacto con un determinado grado de suavidad.

2.3.2 Transformación Wavelet

Para el caso de la transformada wavelet continua tenemos en base a [13] una transformación del la forma:

$$(T^{wav}f)(a,b) = |a|^{-1/2} \int dt\, f(t)\psi\left(\frac{t-b}{a}\right) \qquad (2.3.2.1)$$

Donde $\psi(t)$ cumple la propiedad:

$$\int dt\, \psi(t) = 0 \qquad (2.3.2.2)$$

El núcleo de la transformación es una función $\psi^{a,b}$ en la cual, a representa el parámetro de dilatación y b el de traslación de dicha función. De esta forma la transformación provee una descripción en tiempo y frecuencia de la función f. Se puede observar que esta

función tiene los anchos de tiempo adaptados a su frecuencia, es decir, las $\psi^{a,b}$ de alta frecuencia son muy estrechas, mientras las $\psi^{a,b}$ de baja frecuencia son más anchas. Gracias a lo anterior la transformada wavelet es buena para visualizar fenómenos de corto tiempo de vida y alta frecuencia tales como los transitorios [13].

Si se restringen los valores de a y b de la ecuación (2.3.2.2) a valores discretos, $a = {a_0}^m$ y $b = nb_0{a_0}^m$, con $a_0 > 1$, $b_0 > 0$ fijos, se obtiene una forma de la transformada Wavelet que depende de los valores discretos $m, n \epsilon Z$ y cuya ecuación es la siguiente:

$$T_{m,n}^{wav}(f) = {a_0}^{-m/2} \int dt\, f(t)\psi({a_0}^{-m}t - nb_0) \qquad (2.3.2.3)$$

Fijando a_0 = 2 y b_0=1 entonces existe una $\psi_{m,n}$, con buenas propiedades de localización en tiempo y frecuencia, es decir, un incremento en *m* de una unidad duplicará el ancho de la función wavelet, lo que permitirá encontrar señales con frecuencias más bajas, pero incrementando la incertidumbre de su localización temporal al doble de tiempo que con la *m* original. De esta forma tenemos:

$$\psi_{m,n}(x) = \psi(2^{-m}x - n) \qquad (2.3.2.4)$$

De acuerdo con [13], en la vida real, f no está dada como una función, sino como una serie de muestras que la representan. Debido a que f consiste de valores discretos, el cálculo de $\int dx\, f(x)\psi_{m,n}(x)$ requiere del uso de fórmulas de cuadratura. Para las escalas más pequeñas de interés (exponente m más grande), este procedimiento no emplea muchas muestras de f y el cálculo se realiza de forma rápida. Sin embargo para las escalas más grandes es necesario llevar a cabo muchas operaciones lo que incrementa el tiempo en el proceso de cálculo.

En [13] se propone un proceso por el cual es posible reducir este tiempo de cálculo. Haciendo uso de los principios del análisis multiresolución y de las bases ortogonales, una nueva función ϕ es introducida, de forma que:

$$\psi(x) = \sum_k d_k \, \phi(x-k), \quad (2.3.2.5)$$

$$\phi(x) = \sum_k c_k \, \phi(2x-k), \quad (2.3.2.6)$$

Donde para cada una de las ecuaciones anteriores, un número finito de coeficientes d_k y c_k son diferentes de cero[1]. La integral de ϕ no converge a cero (ψ si cumple con esta característica) y se normaliza de tal forma que $\int dx \, \phi(x) = 1$. Aun teniendo en cuenta que ϕ no es una wavelet por la situación anteriormente descrita, se define $\phi_{m,n}(x) = 2^{-\frac{m}{2}}\phi(2^{-m}x - n)$. De esta forma se puede calcular:

$$\langle f, \psi_{m,n} \rangle = \sum_k d_k \, \langle f, \phi_{m,n+k} \rangle, \qquad (2.3.2.7)$$

Así, el problema de calcular los coeficientes wavelet se reduce a calcular $\langle f, \phi_{m,n} \rangle$ (una serie de combinaciones finitas de ella que da como resultado $\langle f, \psi_{m,n} \rangle$) y por otra parte tenemos:

$$\langle f, \phi_{m,n} \rangle = \frac{1}{\sqrt{2}} \sum_k c_k \, \langle f, \phi_{m-1,2n+k} \rangle, \qquad (2.3.2.8)$$

De tal manera que $\langle f, \phi_{m,n} \rangle$ se puede calcular en forma recursiva, partiendo de la escala más pequeña (donde es más fácil de estimar) hacia la más grandes. Todo esto logrado por convoluciones finitas simples.

[1] Cabe hacer mención que la ecuación (2.3.2.5) difiere al análisis multiresolución en un factor de escalamiento de 2. Esto es, para el análisis multiresolución:

$$\psi(x) = \sum_k d_k \, \phi(2x-k),$$

2.3.3 Relación con los esquemas de filtrado sub banda

El análisis multiresolución lleva en forma natural a un esquema rápido y jerárquico para el cálculo de los coeficientes wavelet de una función dada, suponiendo que se han calculado o han sido dados los productos internos de f con $\phi_{j.k}$, a una escala fina dada [13]. Reescalando las unidades (o bien reescalando f), se puede suponer que la etiqueta de esta escala es $j = 0$. De esta forma es fácil calcular los coeficientes$\langle f, \psi_{j.k} \rangle$ para $j \geq 1$. De [13] tenemos:

$$\psi = \sum_{n} g_n \phi_{-1.n} \quad (2.3.3.1)$$

donde $g_n = \langle \psi, \phi_{-1,n} \rangle = (-1)^n \overline{h_{-n+1}}$. En consecuencia:

$$\psi_{j,k}(x) = 2^{-\frac{j}{2}} \psi\left(2^{-j}x - k\right) = \sum_{n} g_{n-2k}\, \phi_{j-1,n}(x) \qquad (2.3.3.2)$$

En base a $(2.3.3.2)$ se puede construir:

$$\langle f, \psi_{1,k} \rangle = \sum_{n} \overline{g_{n-2k}}\, \langle f, \phi_{0,n} \rangle \qquad (2.3.3.3)$$

Por tanto $\langle f, \psi_{1,k} \rangle$ se puede obtener mediante la convolución de la secuencia $(\langle f, \phi_{0,n} \rangle)_{n \in \mathbb{Z}}$ con $(\overline{g_{-n}})_{n \in \mathbb{Z}}$, reteniendo sólo las muestras pares. De forma similar se tiene:

$$\langle f, \psi_{j,k} \rangle = \sum_{n} \overline{g_{n-2k}}\, \langle f, \phi_{j-1,n} \rangle \qquad (2.3.3.4)$$

que puede ser empleada para calcular los coeficientes de $\langle f, \psi_{j,k} \rangle$ mediante la operación descrita anteriormente (convolución y diezmado con un factor de 2) de $\langle f, \phi_{j-1,n} \rangle$, si estos son conocidos.

De manera análoga al proceso anterior, de la ecuación:

$$\phi_{j,k}(x) = 2^{-\frac{j}{2}}\phi\left(2^{-j}x - k\right) = \sum_{n} h_{n-2k}\phi_{j-1,n}(x) \qquad (2.3.3.5)$$

se obtiene

$$\langle f, \phi_{j,k}\rangle = \sum_{n} h_{n-2k}\langle f, \phi_{j-1,n}\rangle \qquad (2.3.3.6)$$

De esta forma, el procedimiento puede establecerse como [13]: comenzando por $\langle f, \phi_{0,n}\rangle$ se calculan los coeficientes $\langle f, \psi_{1,k}\rangle$ por medio de la ecuación (2.3.3.3) y los coeficientes $\langle f, \phi_{1,k}\rangle$ mediante (2.3.3.6). Se pueden aplicar nuevamente las fórmulas (2.3.3.3) y (2.3.3.6) para calcular $\langle f, \psi_{2,k}\rangle$ y $\langle f, \phi_{2,k}\rangle$ de $\langle f, \phi_{2,k}\rangle$, y así sucesivamente. En cada paso no sólo se calculan los coeficientes wavelet $\langle f, \psi_{j,k}\rangle$ correspondientes al nivel j, también se calculan los coeficientes $\langle f, \phi_{j,k}\rangle$ para el mismo nivel, que permiten calcular el próximo nivel de coeficientes wavelet.

El proceso puede verse también como el cálculo de aproximaciones burdas sucesivas de f [13], que poseen diferente información entre dos niveles sucesivos. Visto de esta forma se inicia con una escala fina de aproximación a f, $f^0 = P_0 f$ (se hace énfasis nuevamente en que P_j es la proyección ortogonal en el espacio V_j mientras que se denotara la proyección ortogonal de W_j como Q_j). Se descompone $f^0 \in V_0 = V_1 \oplus W_1$ en $f^0 = f^1 + \delta^1$ donde $f^1 = P_1 f^0 = P_1 f$ es una aproximación más burda de f en el análisis multiresolución y $\delta^1 = f^0 - f^1 = Q_1 f^0 = Q_1 f$ es lo que se ha perdido en la transición de $f^0 \longrightarrow f^1$. En cada uno de estos espacios V_j, W_j se tienen las bases ortonormales $\left(\phi_{j,k}\right)_{k\in\mathbb{Z}}$ y $\left(\psi_{j,k}\right)_{k\in\mathbb{Z}}$ respectivamente, en forma tal que:

$$f^0 = \sum_{n} c_n^0 \phi_{0,n}\,, f^1 = \sum_{n} c_n^1 \phi_{1,n}, \quad \delta^1 = \sum_{n} d_n^1 \psi_{1,n}$$

Las fórmulas (2.3.3.3) y (2.3.3.6) permiten la transformación de los coeficientes de base ortogonal $\left(\phi_{0,n}\right)_{n\in\mathbb{Z}} \longrightarrow \left(\phi_{1,n}, \psi_{1,n}\right)_{n\in\mathbb{Z}}$ en V_0:

$$c_k^1 = \sum_{n} \overline{h_{n-2k}} c_n^0, \quad d_k^1 = \sum_{n} \overline{g_{n-2k}} c_n^0 \qquad (2.3.3.7)$$

Haciendo uso de la notación $a = (a_n)_{n\in\mathbb{Z}}$, $\bar{a} = (\overline{a_{-n}})_{n\in\mathbb{Z}}$ y $(Ab)_k = \sum_n a_{2k-n} b_n$, se pueden reescribir las expresiones anteriores como:

$$c^1 = \bar{H}c^0, \;\; d^1 = \bar{G}c^0 \qquad (2.3.3.8)$$

La aproximación burda de $f^1 \in V_1 = V_2 \oplus W_2$ puede ser descompuesta nuevamente en $f^1 = f^2 + \delta^2$ con $f^2 \in V_2$ y $\delta^2 \in W_2$ de manera que:

$$f^2 = \sum_n c_n^2 \phi_{2,n}, \;\; \delta^2 = \sum_n d_n^2 \psi_{2,n} \qquad (2.3.3.9)$$

De lo anterior se obtiene:

$$c^2 = \bar{H}c^1, \;\; d^2 = \bar{G}c^1$$

En forma esquemática se puede representar la estructura jerárquica encontrada por medio de la siguiente figura:

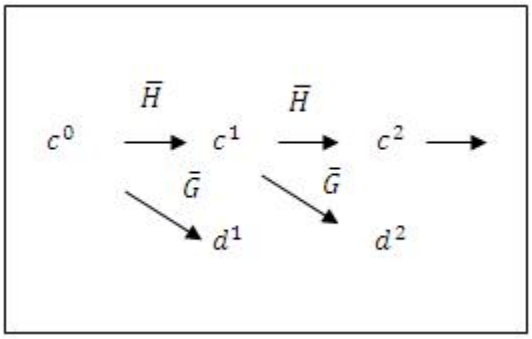

Figura 9. Estructura jerárquica en la obtención de coeficientes wavelet

Considerando que el proceso de obtención de los coeficientes consiste de una convolución y de un diezmado por dos de las muestras resultantes, el esquema anterior se convierte el siguiente:

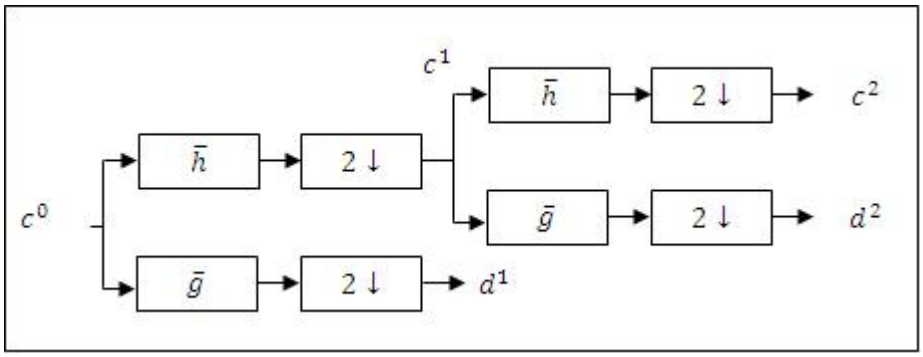

Figura 10. Obtención de coeficientes wavelet mediante banco de filtros

La figura 10 muestra la implementación de codificación sub banda que permite obtener los coeficientes de la transformada wavelet en una forma jerárquica.

Como ejemplo se puede considerar el caso de la transformada Haar aplicada a una secuencia de datos. Para este ejemplo es considerado un esquema sub-banda con una secuencia de entrada $X = \{1,2,3,4,5,6,7,8\}$. Para el caso de la transformada Haar se tiene de [13] $\bar{h} = \{\frac{1}{\sqrt[2]{2}}, \frac{1}{\sqrt[2]{2}}\}$, $\bar{g} = \{\frac{1}{\sqrt[2]{2}}, -\frac{1}{\sqrt[2]{2}}\}$. De esta forma se puede calcular:

$$c^1 = \left\{\frac{3}{\sqrt[2]{2}}, \frac{7}{\sqrt[2]{2}}, \frac{11}{\sqrt[2]{2}}, \frac{15}{\sqrt[2]{2}}\right\}, \quad d^1 = \{-\frac{1}{\sqrt[2]{2}}, -\frac{1}{\sqrt[2]{2}}, -\frac{1}{\sqrt[2]{2}}, -\frac{1}{\sqrt[2]{2}}\}$$

Para estimar los coeficientes c^2 y d^2 se emplean como entrada los coeficientes c^1 y se procede en forma análoga al caso anterior. Esto da como resultado:

$$c^2 = \{5,12\}, \quad d^2 = \{-2,-2\}$$

CAPITULO 3 DESARROLLO DEL SISTEMA

3.1 Esquema general del sistema de evaluación de firmas realizado

El sistema desarrollado permite la realización de cuatro tipos de análisis de firmas, siendo el núcleo central de todos ellos la técnica de DTW modificada. Los cuatro tipos de análisis que el sistema es capaz de realizar son:

- 1. Análisis a nivel de firma completa con DTW
- 2. Análisis a nivel de firma completa usando coeficientes de DWT con DTW.
- 3. Análisis a nivel de trazos o “strokes” de la firma con DTW.
- 4. Análisis a nivel de trazos usando coeficientes de DWT con DTW.

Aunque es posible realizar varios tipos de análisis, el esquema general del sistema es el mismo para todos los casos, siendo el único bloque variable el correspondiente al extractor de características. Dicho esquema se muestra a continuación:

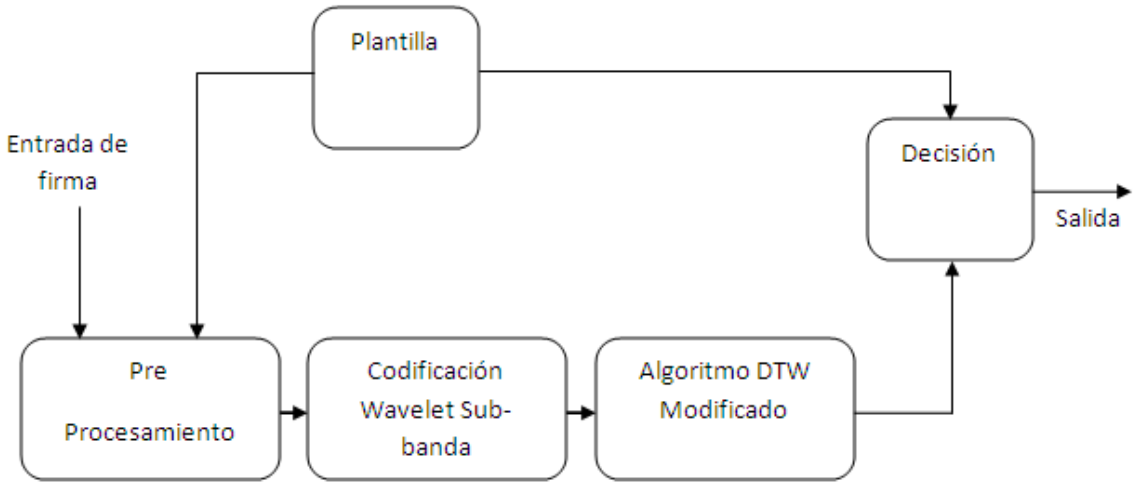

Figura 11. Diagrama a bloques del sistema propuesto

Mientras que en los modelos tradicionales de reconocimiento de firmas se emplea únicamente la información del extractor de características en la etapa de verificación, el presente sistema usa en esta etapa la información del extractor de características en combinación con información del bloque de plantillas generando un nivel de decisión

dependiente del firmante. En las secciones subsecuentes de este capítulo se hace una descripción detallada de los bloques que conforman al sistema.

3.2 Programa de gestión para adquisición de firmas

3.2.1 Introducción

Este es el primer bloque diseñado para la construcción del sistema. Con él es posible adquirir los datos de la firma desde la tableta digitalizadora (“Tablet”) y guardarlos en un archivo que pueda ser empleado posteriormente por MATLAB para el procesamiento de la información adquirida. Dichos datos consisten en los valores de posición X e Y de la pluma sobre la superficie de la tableta así como la presión ejercida por el sujeto de prueba durante la realización de su firma. También se incluye un campo que sirve para indicar el momento el que el firmante realiza un levantamiento de la pluma, y por tanto, poder separar los trazos independientes (“strokes”) que conforman la firma. Esta parte del sistema fue elaborada haciendo uso de Visual Basic 6.0, ya que este software cuenta con los controladores necesarios para el desarrollo de la herramienta de adquisición.

3.2.2 Justificación

Se pensó en desarrollar este bloque-herramienta en un software distinto a MATLAB debido a que MATLAB no cuenta con un controlador que le permita adquirir la información de la tableta digitalizadora de una forma directa. A su vez, la tableta digitalizadora no cuenta con un controlador o Kit de Desarrollo de Software (SDK por sus siglas en inglés), que le permita a MATLAB emplear la tableta digitalizadora como un objeto o instrumento virtual.

Mediante una revisión sobre tarjetas digitalizadoras y controladores para el sistema operativo Windows, se pudo comprobar que Microsoft tiene una plataforma genérica para el uso de tabletas digitalizadoras en aplicaciones elaboradas con Visual Studio, que comprende a Visual Basic. También se pudo comprobar que la plataforma mencionada es

compatible con la versión 6.0 de Visual Basic. Debido a que se contaba con conocimientos básicos previos en el manejo de este intérprete, se optó por emplearlo en el desarrollo de la aplicación.

Otra ventaja de esta implementación es que se trata de una aplicación ejecutable, cuya salida es un archivo cuyo formato puede ser interpretado por múltiples herramientas de procesamiento existentes, entre ellas Matlab, Microsoft Office Excell, Labiew, entre otras. De esta forma, la herramienta puede ser empleada por otros investigadores para trabajos similares sin que sea necesario contar con el software de MATLAB.

3.2.3 Tableta digitalizadora

Para el desarrollo de la investigación se empleó una tarjeta digitalizadora Genius G-PEN 340, cuyas características son las siguientes:

- Área de trabajo: 3” x 4”
- Niveles de presión medibles: 1024
- Resolución: 2000 LPI (Líneas por pulgada) por defecto, 4000 LPI como máximo.
- Tasa máxima de reporte de datos: 100 puntos por segundo.
- Precisión: ±0.25 mm
- Ángulo máximo de inclinación de la pluma durante escritura: 60°.

Se eligió esta tableta debido a que es fácil de conseguir, su bajo costo, la posibilidad de poder medir la presión de escritura y a que sus características satisfacen los requisitos básicos para adquirir firmas.

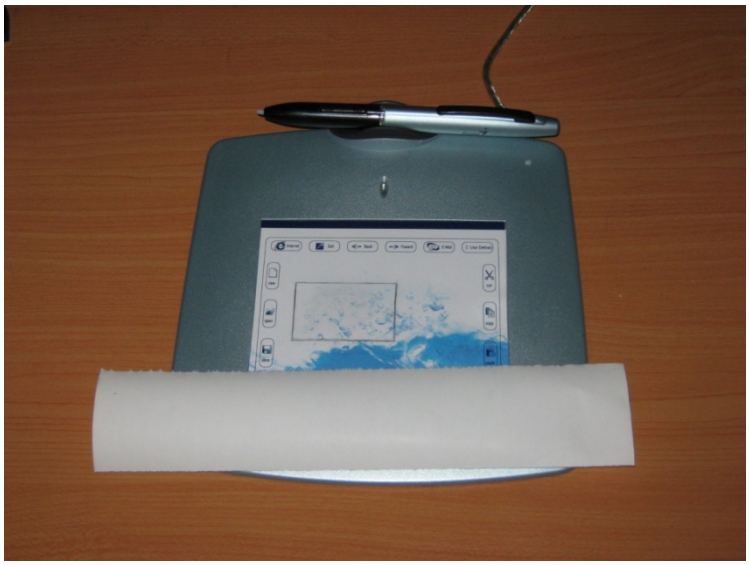

Figura 12. Tableta digitalizadora empleada

3.2.4 Software desarrollado

3.2.4.1 Descripción de la aplicación

La ventana principal de la aplicación desarrollada parara la adquisición de firmas puede observarse en la figura 13:

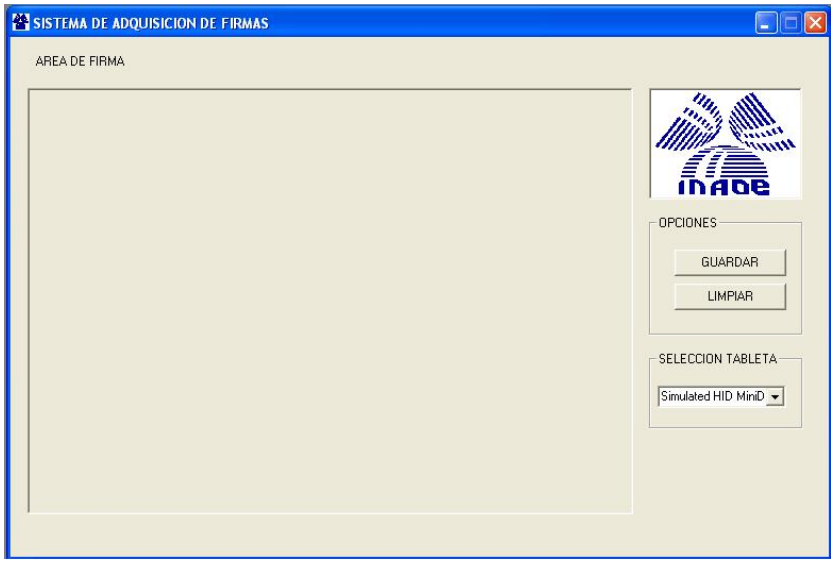

Figura 13. Pantalla principal del programa de adquisición de firmas

El programa desarrollado exhibe esta ventana cuando se ha detectado que existe por lo menos una tableta digitalizadora conectada.

El usuario plasma su firma haciendo uso de la tableta en el área marcada como “AREA DE FIRMA”.

Cabe mencionar que para realizar la adquisición de las firmas se marcó un rectángulo sobre la tableta digitalizadora, haciendo uso de un lápiz, que corresponde con esta área; lo anterior con el objeto de marcar una frontera de referencia para los firmantes. También se hizo uso de papel térmico durante el proceso de firmado con el objeto de brindar una retroalimentación visual a la persona que firma y, de esta forma, asemejar el proceso a una situación más real.

El botón “LIMPIAR” se emplea con el objeto de borrar las firmas que presentan errores. Durante el proceso de adquisición, estos errores se debieron principalmente a que los firmantes no estaban familiarizados con el uso de este tipo de tarjetas digitalizadoras, o bien porque rebasaron las fronteras establecidas por el área de la firma.

El botón “GUARDAR” se emplea cuando la firma adquirida tiene la calidad suficiente, es decir no presenta falta de trazos, ha quedado dentro del área de firmado y el usuario ha determinado que los trazos son similares a como usualmente hace su firma, para almacenarse de forma definitiva.

La caja de selección de la parte inferior del botón guardar permite seleccionar qué tableta digitalizadora será empleada en el caso de que existiera más de una, además de brindar información sobre el nombre del controlador que se encarga de administrarla.

El archivo de salida es de tipo CSV (Valores separados por coma por sus siglas en inglés) [15]. Este archivo contiene 4 columnas de datos. En la primera columna se encuentran los datos correspondientes a la posición X, la segunda contiene la información de la posición Y, la tercera columna contiene la información correspondiente a la presión y la última de ellas contiene marcas (número 1) que indican el momento en que ocurre un levantamiento de la pluma. Esta última columna se usa para determinar de forma directa el momento en que terminan los trazos (strokes) de la firma. Los valores de X e Y se acondicionan en esta aplicación de forma que ambos tienen como valor mínimo a 0, de esta forma no importa sobre qué posición del área de firmado de la tableta digitalizadora se firme, los valores siempre tendrán como referencia mínima los que se han indicado.

Haciendo uso de esta aplicación en la forma indicada se obtuvieron todas las muestras de firmas que fueron empleados en la investigación. Para más detalles de la aplicación consultar el apéndice A2

3.3 Sistema de Procesamiento

3.3.1 Introducción

Esta es la parte del software encargada de tomar los datos adquiridos en la etapa anterior para determinar, de acuerdo con una serie de algoritmos establecidos, la semejanza de la firma adquirida con alguna firma de referencia previamente almacenada en el sistema. Para ello, primero es necesario acondicionar dichos datos con el propósito de que sea posible lograr una comparación que no dependa de características posicionales o de la escala de la firma; esta es la función del bloque de preprocesamiento. El bloque extractor de características realiza un procesamiento con el propósito de obtener de la firma información relevante que pueda lograr una discriminación eficiente de la misma con respecto a la de otros firmantes. Una vez que se ha obtenido dicha información se coteja con alguna referencia previamente almacenada para determinar el grado de similitud entre ambas y poder determinar si la firma actual es auténtica o no. En las notas subsecuentes se describen a detalle los bloques mencionados.

3.3.2 Preprocesamiento

Para el presente trabajo, es empleada la información de posición en X e Y de la firma.

Debido a que fue necesario hacer que el sistema fuera capaz de poder cotejar los datos de la firma ingresada con la de las referencias internas, que en lo subsecuente serán llamadas plantillas, fue diseñado un esquema específico de preprocesamiento.

Con el propósito de poder comparar las firmas en condiciones análogas, se reduce la dependencia de los datos con respecto de la posición e inclinación de la firma, aplicando la transformada Hotelling a las secuencias de tiempo X, Y originales. Esto es:

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \end{bmatrix} = A \begin{bmatrix} X - \mu_X \\ Y - \mu_y \end{bmatrix} \qquad (3.3.2.1)$$

Donde A representa la matriz de los eigenvectores normalizados de la matriz de covarianza de los valores de X e Y [19]. Haciendo uso de esta transformación es posible centrar la firma sobre el origen y orientarla sobe su diagonal principal.

Con base en [16] se pudo apreciar la ventaja de tratar la información de posición en x e y de la firma como un número complejo $Z = \hat{X} + i\hat{Y}$. Por medio de esta transformación es posible representar a una función de dos variables espaciales x e y como un número complejo de la forma:

$$\boldsymbol{Z} = \boldsymbol{Re^{j\theta}} \qquad (3.3.2.2)$$

Como primer conjunto de datos para analizar se emplean los correspondientes a los elementos de magnitud de este número complejo, centrados sobre su media y normalizados para obtener una varianza igual a 1. Esto es:

$$\boldsymbol{M} = \frac{|\boldsymbol{Z}| - \mu(|\boldsymbol{Z}|)}{\sigma(|\boldsymbol{Z}|)} \qquad (3.3.2.3)$$

El proceso anterior se realiza para obtener una comparación independiente del tamaño de la firma.

Para el caso de la fase fue necesario realizar una modificación adicional. Con el propósito de evitar transiciones abruptas de la información de fase con respecto al cambio de cuadrantes cuando la firma se encuentra centrada en el origen, los puntos de la firma son limitados al primer cuadrante, de la forma siguiente:

$$\boldsymbol{F_f} = \boldsymbol{tan^{-1}}\left(\frac{Im(\boldsymbol{Z}) - \min\,(Im(\boldsymbol{Z})}{Re(\boldsymbol{Z}) - \min\,(Re(\boldsymbol{Z})}\right) \qquad (3.3.2.4)$$

Lo anterior, limita los valores de $\boldsymbol{F_f}$ al rango $[0, \frac{\pi}{2}]$.

Finalmente, se centran los valores de $\boldsymbol{F_f}$ sobre su media, de forma que:

$$\boldsymbol{F} = \boldsymbol{F_f} - \mu(\boldsymbol{F_f})$$

Ambas funciones descritas anteriormente (**M** y **F**) serán utilizadas en el extractor de características, sus valores serán ponderados y sumados para obtener el valor final. En el presente trabajo el valor de la ponderación para cada parámetro es de 0.5, es decir, el valor de salida es el valor obtenido en el extractor de características para la función **M** multiplicado por 0.5 más el valor obtenido en el extractor de características para la función **F** multiplicado por 0.5.

3.3.3 Extractor de características

Como se ha mencionado con anterioridad, este bloque es el único que presenta variaciones con respecto al esquema general, ya que su configuración puede ser alterada para realizar distintos tipos de análisis. La figura 14 representa una vista interna del bloque extractor de características con sus respectivos subsistemas:

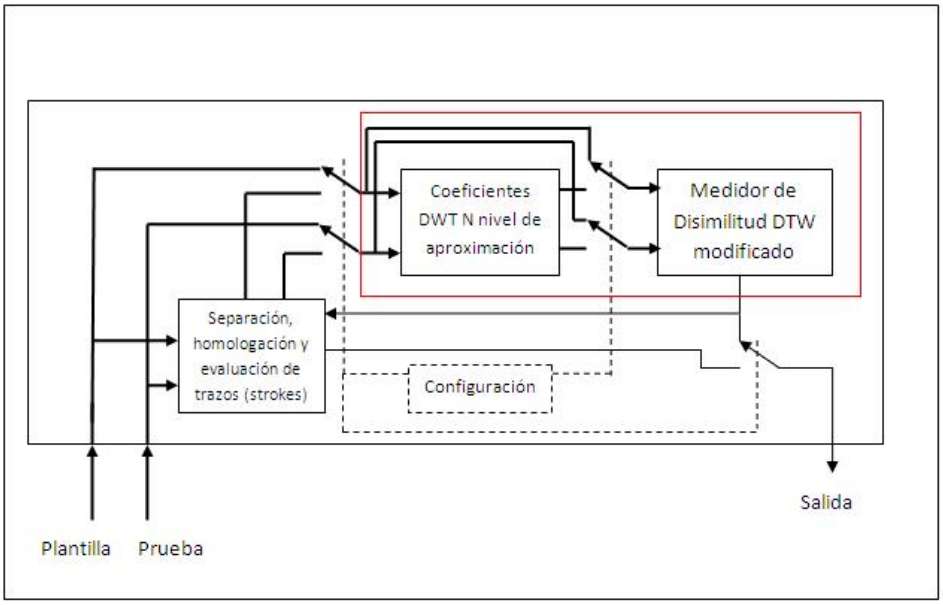

Figura 14. Diagrama de bloques del extractor de características

En esta figura podemos apreciar que el bloque tiene dos entradas, una para la secuencia de tiempos correspondiente a los datos preprocesados de la plantilla y la otra para recibir

la secuencia de tiempo que contiene los datos preprocesados de la firma bajo prueba. En la realización del presente trabajo, para comparar una firma, se procedió a realizar la prueba de disimilitud a las secuencias de tiempo con la información de magnitud M y posteriormente a las secuencias de tiempo correspondientes a la fase F, sumando los resultados de los valores de disimilitud obtenidos por cada prueba y dividiendo el resultado entre dos. La salida del sistema es simplemente un número cuyo valor va del 0 al 2. Cuando este valor de salida es 0, existe una correspondencia máxima entre la firma de la plantilla y la de prueba (de hecho esta situación indica que ambas firmas son iguales), mientras que un valor numérico más alto es indicación de un mayor número de disimilitudes entre ambas firmas.

En la figura podemos apreciar la existencia de tres subsistemas. Los interruptores son simbólicos y son empleados para indicar un arreglo de sub-bloques específico. Como puede observarse el comportamiento del bloque depende de la variable de configuración. Más adelante se explicarán las configuraciones posibles del sistema.

3.3.3.1 Subsistemas del bloque Extractor de características

<u>Sub bloque Medidor de Disimilitud DTW modificado</u>

Este bloque recibe como entradas dos secuencias de tiempo y su salida principal es un número que indica la similitud del alineamiento y normalización de ambas secuencias. De esta forma, si las secuencias de entrada son idénticas, el bloque devuelve un valor de 0, y este valor se incrementa a medida que las secuencias de entrada pierden similitud. A su vez, este sub bloque también es capaz de devolver un segundo valor de salida que indica la cantidad de pasos necesarios para lograr el alineamiento óptimo.

Para determinar el primer valor de salida del bloque, se ha empleado como referencia la medida de disimilitud en base a Dynamic Time Warping (DTW) con restricción de trayectoria local tipo 1 [11] y se ha normalizado con respecto a la longitud de la trayectoria de warping (T) resultante de esta operación, esto es:

Para las secuencias de tiempo de entrada $X = x\{i\}\ con\ \ i = 1,2, \dots T_x$ y $Y = y\{k\}\ con\ k = 1,2, \dots T_y$:

$$D_v = \frac{D(T_x, T_y)}{T} \qquad (3.3.3.1.1)$$

Otro criterio empleado en el presente trabajo para la evaluación de la disimilitud entre las secuencias de tiempo indicadas anteriormente corresponde a la deformación de la línea de trayectoria óptima generada por el DTW. Considerando dos secuencias idénticas ingresadas al DTW obtendremos que esta trayectoria se trata de una relación uno a uno de los puntos de las secuencias, y su gráfica describirá una diagonal sobre el cuadrado generado por su matriz de distancia acumulada. La figura 15 muestra la trayectoria diagonal generada por dos secuencias idénticas sobre una imagen de la matriz de distancia acumulada donde el color negro representa valores pequeños:

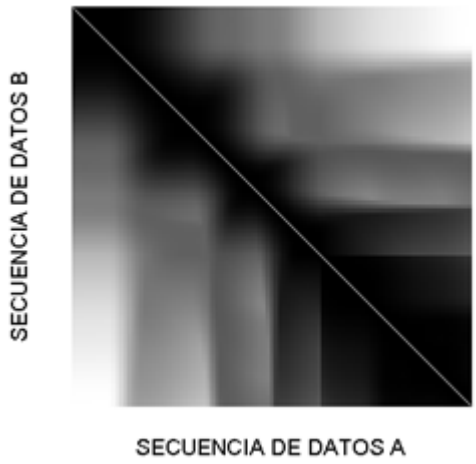

Figura 15. Diagonal de trayectoria DTW en secuencias iguales

A medida que la disimilitud de las secuencias se incrementa, la trayectoria se desvía de esta diagonal. Analizando las restricciones globales indicadas en [11] es posible determinar que cualquier trayectoria con restricción local tipo 1 tendrá como peor caso un número total de $T_x + T_y$ pasos. Este valor se emplea para normalizar este parámetro. La ecuación siguiente nos permite obtener el valor del segundo criterio:

$$D_N = \frac{T}{T_x + T_y} \qquad (3.3.3.1.2)$$

En conjunto, estos dos criterios conforman una forma de DTW normalizado.

Finalmente, en base a una descripción encontrada en [2] se decidió emplear un tercer criterio. En la literatura indicada, se les denomina DMP's, y son descritos a continuación:

Un DMP de T con respecto a R^i esta identificado como un punto que tiene un acoplamiento uno a uno con un punto de R^i. En otras palabras sea T_p un punto de T acoplado con un punto R_q^i de R^i, T_p es un DMP de T con respecto a R^i si y solo sí:

1) $\forall \underline{p} = 1, \dots, N_T, \underline{p} \neq p$, resulta que $T_{\underline{p}}$ no está acoplado con R_q^i.
2) $\forall \underline{q} = 1, \dots, N_T, \underline{q} \neq q$, resulta que $R_{\underline{q}}^i$ no está acoplado con T_p.

En la figura siguiente se muestran de forma gráfica los DMP's.

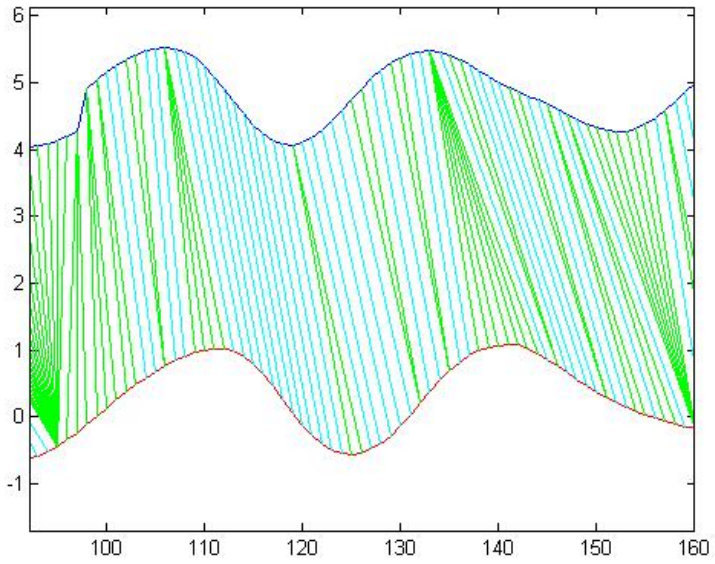

Figura 16. Ilustración gráfica de los DMP's

La figura 16 muestra una sección de dos secuencias de tiempo a las que se les ha aplicado la operación de DTW, indicando en color verde los puntos que se han relacionado a través de la técnica. En color cian se muestran los DMP´s mencionados.

Nuevamente, en el caso de tener dos secuencias idénticas, el número de DMP's sería igual al número de puntos que contienen las secuencias de warping (que también será igual al número de puntos que tengan las secuencias de entrada), por lo que para normalizar el valor de este coeficiente de interés se ha planteado la siguiente ecuación:

$$C_{dmp} = 1 - \frac{\sum dmp}{T} \qquad (3.3.3.1.3)$$

La salida del sub bloque correspondiente a la medida de disimilitud, es el producto de los valores determinados por cada criterio anteriormente descrito. Esto es:

$$MV_{DTW} = (D_v)(D_N)(C_{dmp}) = \left(\frac{D(T_x, T_y)}{T}\right)\left(\frac{T}{T_x + T_y}\right)\left(1 - \frac{\sum dmp}{T}\right)$$
$$= \frac{\left(D(T_x, T_y)\right)(1 - \sum dmp)}{T(T_x + T_y)}$$

Una segunda salida del sistema es simplemente la cantidad de pasos para obtener la trayectoria óptima de alineamiento. De esta forma:

$$DTWMP = T$$

El parámetro MV_{DTW} es empleado como la medida de disimilitud entre las secuencias de tiempo comparadas, en el que un valor cercano a 0 indica una gran similitud entre estas secuencias tanto en la duración como en sus valores. Este valor empleado en las tareas de reconocimiento y verificación para determinar la similitud de las firmas de entrada.

El parámetro $DTWMP$ es usado por el algoritmo de separación de trazos para determinar el tamaño relativo de la comparación de un trazo con respecto de toda la firma y determinar la ponderación que le será dada al momento de calcular la disimilitud.

3.3.3.2 Sub Bloque de coeficientes wavelet de aproximación de nivel N.

Como su nombre lo indica, este bloque tiene como objetivo obtener los coeficientes de aproximación de la transformada wavelet discreta de nivel N. La función wavelet madre

así como el nivel N de aproximación empleados pueden ser configurados por el experimentador. De los fundamentos teóricos se puede deducir que mientras N sea mayor, la cantidad de coeficientes de aproximación es menor pero la función resultante de la síntesis de estos coeficientes es una versión muy burda de la función original. Se decidió hacer configurables los parámetros mencionados con el objeto de poder realizar distintas pruebas y determinar cuál de las funciones wavelet y en qué nivel se obtienen los mejores resultados. Este bloque, en unión con el bloque Medidor de Disimilitud Modificado, conforman el núcleo principal del bloque Extractor de características que se encuentra marcado en rojo en el esquema. Este sub-bloque puede ser usado o no antes de realizar la medida de disimilitud de las funciones de tiempo de acuerdo con la configuración indicada por el experimentador, es decir, la prueba de disimilitud puede realizarse entre los datos de las secuencias de tiempo o entre los coeficientes de aproximación wavelet correspondientes a dichas secuencias.

3.3.3.3 Separación, homologación y evaluación de trazos ("strokes")

Dos de los análisis realizados por el sistema tienen la capacidad de comparar las firmas separándolas en trazos individuales y emplear cada trazo como si se tratara de una firma completa con respecto a los trazos de la firma plantilla. Los datos de la etapa de adquisición demostraron que aunque un mismo firmante plasmó su firma en forma consecutiva, su número de trazos tendió a cambiar dentro de un rango de uno a dos trazos en promedio. Lo anterior fue considerado debido a que para realizar las pruebas establecidas es necesario contar con un número idéntico de trazos. Por lo que este sub bloque se desarrolló con el propósito de realizar la separación de los trazos, hacer que el número de trazos a comparar sea idéntico y evaluar la disimilitud de cada trazo con respecto a los trazos correspondientes de la plantilla en el núcleo descrito en la sección anterior. La salida de este sub bloque es un valor numérico que indica la disimilitud entre los datos preprocesados de la firma plantilla y los datos preprocesados de la firma de prueba a nivel de los trazos.

<u>Separación de trazos</u>

Se debe recordar que en este punto del sistema la firma ya no existe como tal (ya no es una función de dos componentes x e y sino de una magnitud M y una fase F), así que la información correspondiente a los trazos se obtiene desde la sección previa al preprocesamiento, de los datos entregados por la etapa de adquisición. En este conjunto de datos existe una columna que marca los puntos en donde la pluma de la tableta digitalizadora es levantada. Esta es la información empleada en la etapa de separación de trazos.

Para separar los trazos, basta localizar las marcas que identifican el levantamiento de la pluma (1 en esta secuencia de datos) y definir el trazo como todos los puntos previos a esta marca. Esta misma operación se repite para todos los trazos subsecuentes.

Homologación y evaluación de trazos

Una vez que se han separado los trazos se procede a realizar la homologación (hacer coincidir el número de trazos de la firma de prueba y la de plantilla) y a su evaluación. La entrada a esta parte del sistema consiste en dos conjuntos de secuencias de tiempo, cada secuencia de dichos conjuntos representa a un trazo. La idea de esta sección es, tomando como base el conjunto con menor número de elementos, comparar cada elemento de este conjunto con uniones consecutivas de elementos del conjunto restante, es decir, unir trazos de la firma con mayor número de ellos y compararlos con los trazos individuales de la firma que contiene menos trazos.

Se ha impuesto una restricción con respecto a la diferencia de trazos entre una firma y la otra. Con base en el trabajo de [3], se decidió limitar esta diferencia a 2. Si una de las firmas contiene 2 trazos más que la otra, por defecto se entrega en la salida de esta sección un valor numérico igual a 2 (Máxima disimilitud).

En el caso de que el número de trazos se encuentre dentro del rango establecido, se procede a realizar la comparación, siendo esta efectuada por medio del Medidor de disimilitud DTW modificado y, según el tipo de análisis se intercala o no el sub-bloque para obtener los Coeficientes DWT para el tercer nivel de aproximación.

Primero es realizada la comparación del primer trazo de la firma con menor número de ellos, en contra del primer trazo de la otra firma y es obtenido el valor numérico de la disimilitud de esta comparación. Posteriormente, se repite la operación pero ahora con la unión del primer y segundo trazo de la firma con mayor número de trazos y se obtiene un

segundo valor de disimilitud y, en caso de que la diferencia en el número de trazos sea igual a 2, también es realizada una comparación con la unión del primer, segundo y tercer trazo. Se determina el mínimo de estos tres valores, se obtiene el número de pasos de la trayectoria óptima y el número de trazos que se unieron. Este último número será restado del número de comparaciones que se efectúa en la siguiente iteración. En la última de las iteraciones, el trazo final de la firma con menor número de ellos es comparada contra los que restan de la firma con mayor número de trazos y se obtienen los parámetros descritos.

Una vez obtenidos estos parámetros se procede a calcular el valor de disimilitud en la siguiente forma:

$$salidas = \frac{\sum_{i=1}^{N}(d_i T_i)}{\sum_{i=1}^{N}(T_i)} \qquad (3.3.3.3.1)$$

donde T_i representa el número de pasos de la trayectoria óptima resultante del acoplamiento de menor disimilitud i-ésimo y d_i, el valor de su disimilitud y N el número de trazos de la firma con menor número de ellos. La ecuación indicada busca dar mayor peso a las disimilitudes encontradas entre los trazos de mayor duración.

Se puede resumir el proceso anterior por medio del diagrama de flujo descrito en la siguiente figura:

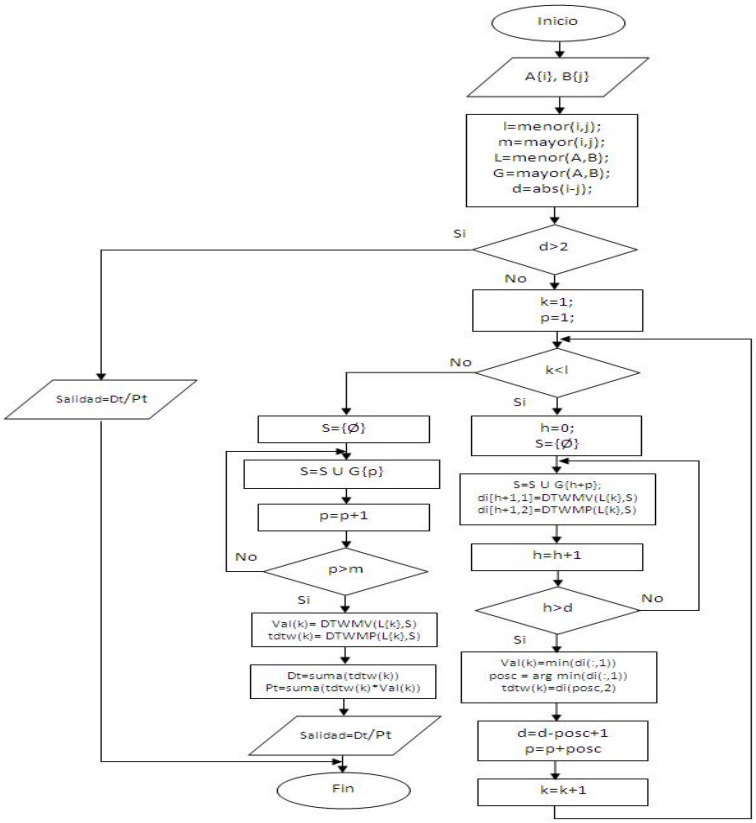

Figura 17. Diagrama de flujo del algoritmo homologador y evaluador de trazos

3.3.4 Bloque de plantillas

En este bloque se encuentra la base de datos que contiene las secuencias de tiempo correspondientes a todas las firmas plantilla además de los valores de media y desviación estándar descritos para cada una. Estas medidas estadísticas serán obtenidas del proceso de comparación de un número determinado de firmas seleccionadas para el entrenamiento del sistema. La información de la secuencia de tiempos correspondiente a la firma plantilla es empleada en el proceso Extractor de Características, mientras que los dos datos restantes se emplean en el bloque Verificador.

3.3.4.1 Obtención de plantillas

Para generar la firma plantilla que representa a cierto firmante en el proceso se desarrollan los siguientes pasos:

1. Se toman N muestras de firmas del mismo firmante. Para el presente trabajo N tiene un valor de 5, aunque el programa está diseñado para configurar cualquier valor.
2. Se toma la muestra número 1 y se compara con las restantes. Este proceso hace uso del extractor de características descrito anteriormente. El valor de salida de cada comparación es guardado. Posteriormente se obtienen la media y la varianza de los valores guardados.
3. Se repite el procedimiento para las firmas restantes.
4. Una vez obtenidos los datos anteriores, la firma elegida como plantilla es determinada por aquella muestra que haya registrado la media más pequeña.
5. En la base de datos correspondiente a la plantilla se guardan la secuencia de datos de la firma elegida como plantilla, su media y su desviación estándar.

3.3.5 Bloque Verificador

Este bloque tiene como propósito determinar cuándo una firma de prueba ha coincidido con una plantilla. Este bloque toma la información de media y de desviación estándar de la base de datos de plantillas y haciendo uso de estos datos genera un nivel de disparo, que se calcula de la forma siguiente:

$$th = \mu_i + k\sigma_i$$

donde μ_i y σ_i representan la media y la desviación estándar de la plantilla i-esima respectivamente, k es un factor de peso configurable por el usuario. El valor de k se usa principalmente para dar mayor o menor flexibilidad en las tareas de reconocimiento y verificación de firmas, pero también es empleado para determinar la FAR y FRR. El parámetro Rec determina si el reconocimiento de la firma bajo prueba ha sido positivo o negativo. Este parámetro indica cuando existe coincidencia entre la firma de prueba y la plantilla e indica un reconocimiento exitoso. Se calcula en la siguiente forma:

$$Rec = \begin{cases} positivo, & si\ MV_{DTW} \leq th \\ negativo, & si\ MV_{DTW} > th \end{cases}$$

3.4 Tipos de análisis implementados en el sistema para la realización de pruebas.

De acuerdo con los bloques descritos, el sistema permite llevar a cabo cuatro distintos tipos de pruebas en el proceso de análisis de la firma:

3.4.1 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma completa.

En esta prueba son introducidas de forma directa las secuencias de tiempo preprocesadas de la plantilla y de la firma a comprobar, en el sub bloque Medidor de Disimilitud DTW modificado. El bloque Extractor de características adquiere la siguiente forma:

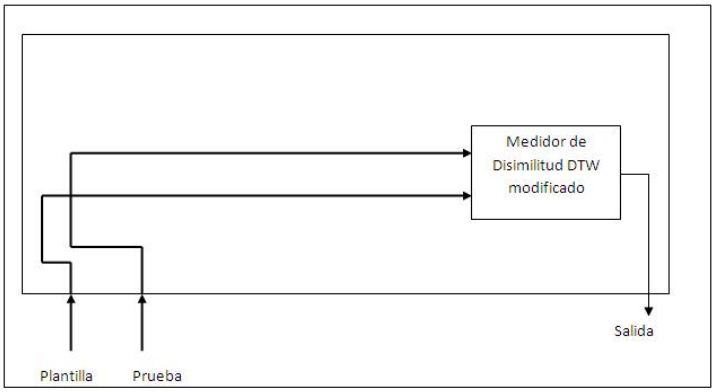

Figura 18. Bloque extractor de características para prueba tipo 1

3.4.2 Comparación de los coeficientes de aproximación de firma plantilla y firma de prueba considerando la firma completa.

En esta prueba las secuencias de tiempo preprocesadas de la plantilla y de la firma a comprobar ingresan al sub bloque de coeficientes wavelet de aproximación de nivel N. Una vez obtenidos dichos coeficientes, estos ingresan en el sub bloque Medidor de Disimilitud DTW modificado. El bloque Extractor de características adquiere la siguiente forma:

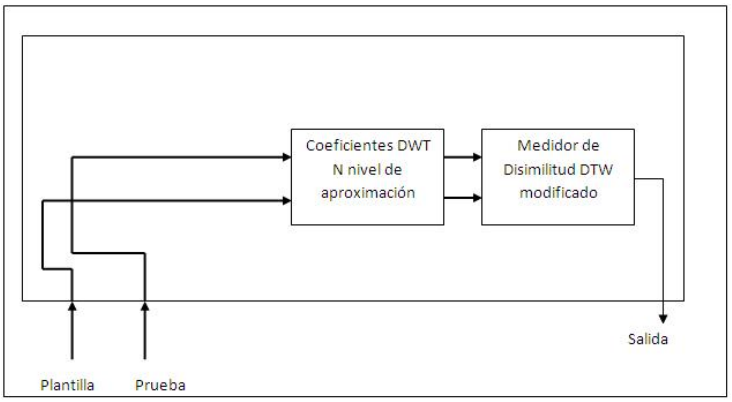

Figura 19. Bloque extractor de características para prueba tipo 2

3.4.3 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma dividida en trazos.

En esta prueba, las secuencias de tiempo preprocesadas de la plantilla y de la firma a comprobar son ingresadas en el sub bloque de Separación, homologación y evaluación de trazos. Este sub bloque emplea de forma directa el Medidor de Disimilitud DTW modificado para evaluar los trazos. Para esta prueba el bloque Extractor de características adquiere la forma:

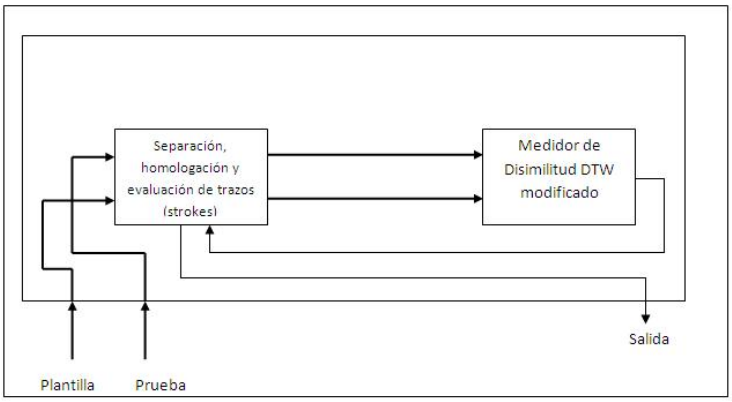

Figura 20. Bloque extractor de características para prueba tipo 3

3.4.4 Comparación de las secuencias de firma plantilla y firma de prueba considerando la firma dividida en trazos.

La prueba es similar a la anterior, con la diferencia de que en la evaluación de los trazos, primero se obtienen los coeficientes wavelet de aproximación de nivel N y posteriormente se envían estos coeficientes al sub bloque Medidor de Disimilitud DTW modificado. El bloque Extractor de características adquiere la siguiente forma:

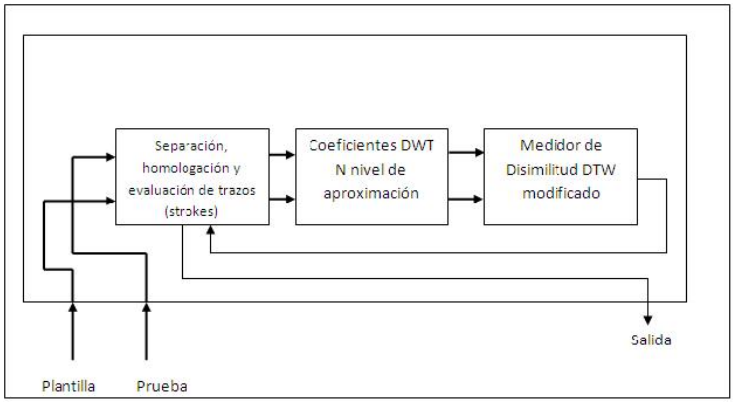

Figura 21. Bloque extractor de características para prueba tipo 4

3.5 Herramienta de procesamiento desarrollada en MATLAB

Con el propósito de desarrollar la presente investigación, fue elaborada una herramienta que permite realizar los análisis descritos. Para el desarrollo de esta herramienta fue usado el software MATLAB, debido a que cuenta con algoritmos matemáticos de gran utilidad, tales como la transformada discreta Wavelet.

La herramienta desarrollada permite organizar la información obtenida de la etapa de adquisición y que se encuentra en archivos guardados en distintas carpetas, realizar los análisis descritos en la sección anterior y evaluar el comportamiento del sistema para un tipo de análisis dado, por medio de una aplicación de verificación y otra de reconocimiento. Todas estas funciones se han integrado en una interface gráfica. La descripción de la interface así como los detalles de su funcionamiento pueden consultarse en el apéndice A3.

CAPITULO 4 RESULTADOS Y CONCLUSIONES

4.1 Resultados

Como ya se dijo, se obtuvieron un total de 20 firmas de 50 personas para conformar una base de datos de 1000 firmas en total. De este número de firmas, para cada firmante 5 firmas son empleadas para el entrenamiento y las 15 restantes en la validación.

En la tabla 1 se enlistan algunas características relevantes sobre el espacio muestral de la base de firmas obtenida:

Tabla 1. Características del espacio muestral

Característica	Valor
Número máximo de trazos (strokes) en una firma	18
Número promedio de trazos (strokes) por firma	5
Tiempo máximo de firmado	11 segundos
Tiempo promedio de firmado	4.33 segundos
Tiempo mínimo de firmado	1.14 segundos

La figura 22 muestra un histograma de la distribución del número de trazos en la muestra:

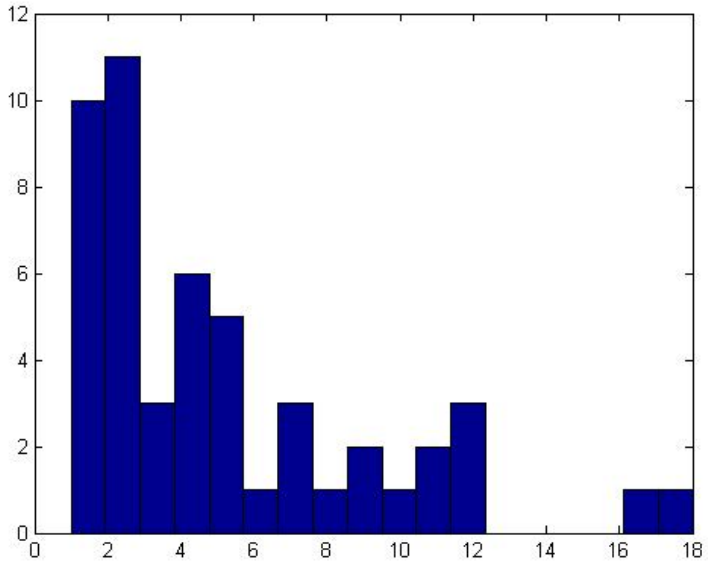

Figura 22. Histograma del número de trazos de la muestra

La figura 23 muestra el histograma del tiempo de firmado:

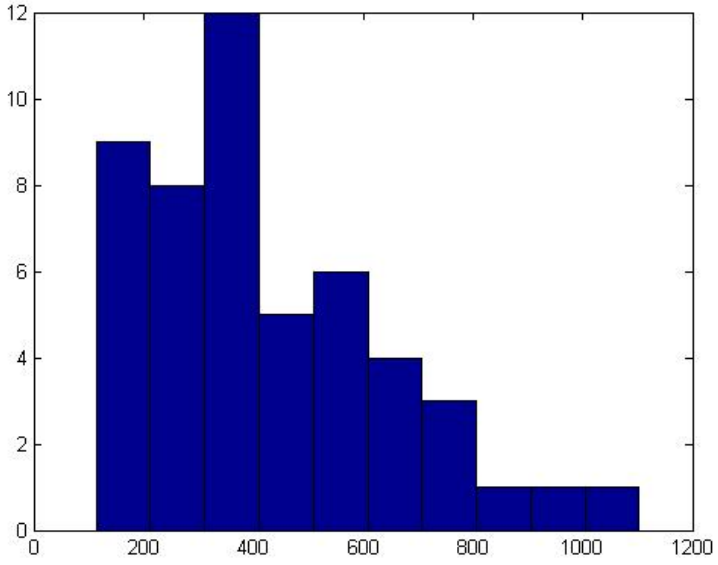

Figura 23. Histograma del tiempo de firmado de la muestra

Condiciones del experimento

- Para la elaboración de la plantilla se emplearon los ejemplares 2, 7, 11, 13 y 18 de cada firmante. Se emplearon estos números de muestra con el propósito de considerar firmas realizadas al inicio, en la parte media y al final de la sesión de firmado.
- Se emplearon los resultados correspondientes a las firmas falsificadas entrenadas, siendo cotejada la plantilla con cada una de las falsificaciones para determinar la FAR y cotejando la plantilla con el resto de firmas pertenecientes al mismo firmante de la plantilla en la determinación de la FRR.
- Se probaron distintos niveles del nivel de ponderación de la desviación estándar empleada en el verificador, modificando el nivel de disparo para conformar las gráficas de FAR y FRR y poder determinar el nivel de EER. El eje de las abscisas corresponde con el valor de ponderación indicado.

Prueba 1

Prueba comparativa entre resultados haciendo uso de la información de X e Y y magnitud y fase, mediante la evaluación tipo 1 (DTW modificado para la firma completa).

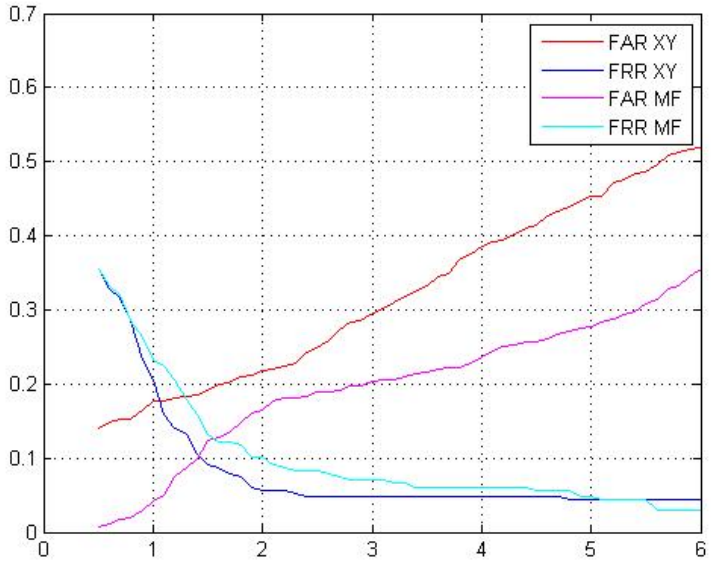

Figura 24 Gráfica FAR/FRR del empleo de las funciones X,Y vs Magnitud/Fase

El resultado puede resumirse mediante la siguiente tabla:

Tabla 2. Resultados EER del empleo de las funciones X,Y vs Magnitud/Fase

Prueba	EER
DTW modificado para firma completa, funciones posición X y posición Y	17.71%
DTW modificado para firma completa, funciones Magnitud y Fase	12.57%

De acuerdo con esta prueba, se puede observar que el empleo de información de magnitud y fase reduce la EER. Con base en este resultado, estos parámetros serán empleados para la elaboración de las siguientes pruebas.

Prueba 2

Prueba comparativa entre el método DTW normal y el DTW modificado para la firma completa haciendo uso de las funciones de magnitud y fase.

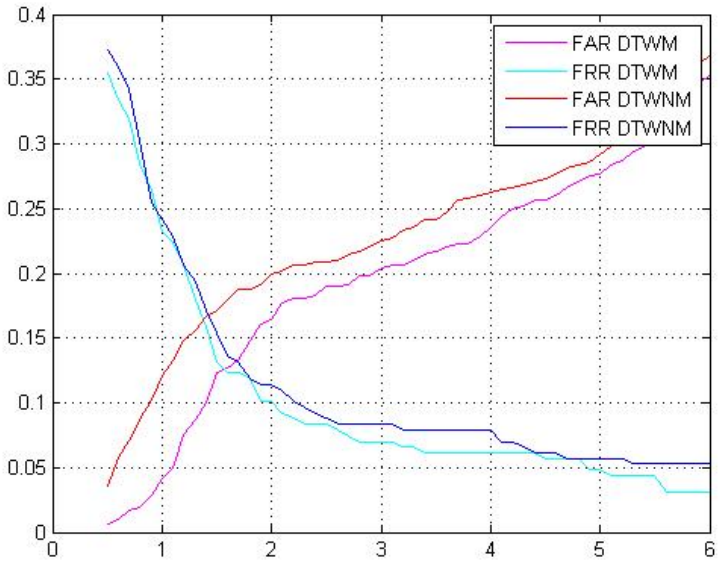

Figura 25 Gráfica FAR/FRR algoritmo DTW modificado vs algoritmo DTW normal

La siguiente tabla resume el resultado de esta prueba

Tabla 3 Resultados EER algoritmo DTW modificado vs DTW normal

Prueba	EER
DTW no modificado para firma completa	16.75%
DTW modificado para firma completa.	12.57%

Los resultados anteriores muestran que la modificación del algoritmo de DTW para considerar los DMP's ha reducido la EER. Debido a esto, para las subsecuentes pruebas serán empleadas estas condiciones.

Prueba 3

Prueba comparativa entre los distintos tipos de análisis. Cabe mencionar que estos son:

- Tipo 1: DTW para firma completa.
- Tipo 2: DTW de coeficientes Wavelet para firma completa.
- Tipo 3: DTW para firma a nivel de trazos.
- Tipo 4: DTW para coeficientes Wavelet de trazos de la firma.

La gráfica siguiente muestra los resultados obtenidos en cada caso:

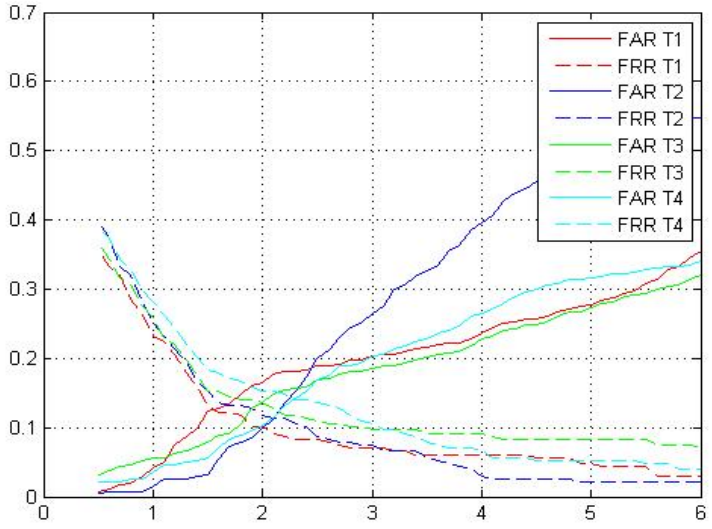

Figura 26 Gráfica comparativa FAR/FRR para distintos tipos de análisis implementados

La siguiente tabla resume los resultados anteriores considerando los puntos de EER. También se incluye información con respecto al tiempo promedio empleado en cada caso. Con respecto a los análisis de tipo 2 y tipo 4, la DWT empleada es la Haar, coeficientes de aproximación de nivel 3.

Tabla 4. Resultados EER para distintos tipos de análisis implementados

Prueba	EER	Tiempo promedio de procesamiento
Tipo 1	12.57%	0.7318 segundos
Tipo 2	11.40%	0.0211 segundos
Tipo 3	13.58%	0.4895 segundos
Tipo 4	14.09%	0.0314 segundos

Se observa que el análisis del tipo 2 es el que lleva a mejores resultados con respecto al EER. También se observa que este tipo de análisis reduce de forma significativa el tiempo de procesamiento.

Prueba 4

Prueba comparativa haciendo uso de distintos tipos de DWT para el caso de los análisis del tipo 2.

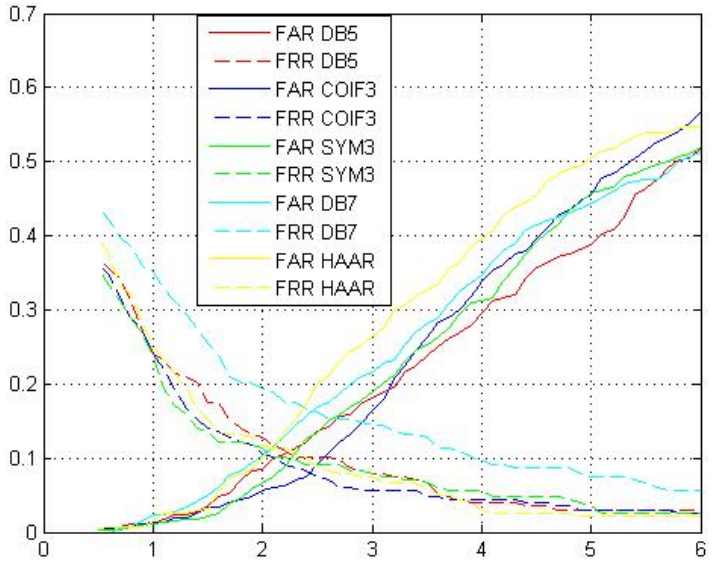

Figura 27 Gráfica comparativa FAR/FRR análisis tipo 2 para diferente DWT

La siguiente tabla resume estos resultados:

Tabla 5 Resultados EER para distintas DWT análisis tipo 2

DWT empleada	EER	Tiempo promedio de procesamiento
Db5	10.79%	0.0274 segundos
Coif3	8.05%	0.0322 segundos
Sym3	9.88%	0.0246 segundos
Db7	16.40%	0.0297 segundos
Haar	11.40%	0.0211 segundos

Se observa que la transformada discreta wavelet Coif3, entrega el menor valor de EER.

Prueba 5

Prueba comparativa con respecto al nivel de aproximación de los coeficientes wavelet para la prueba tipo 2 con DWT Coif3.

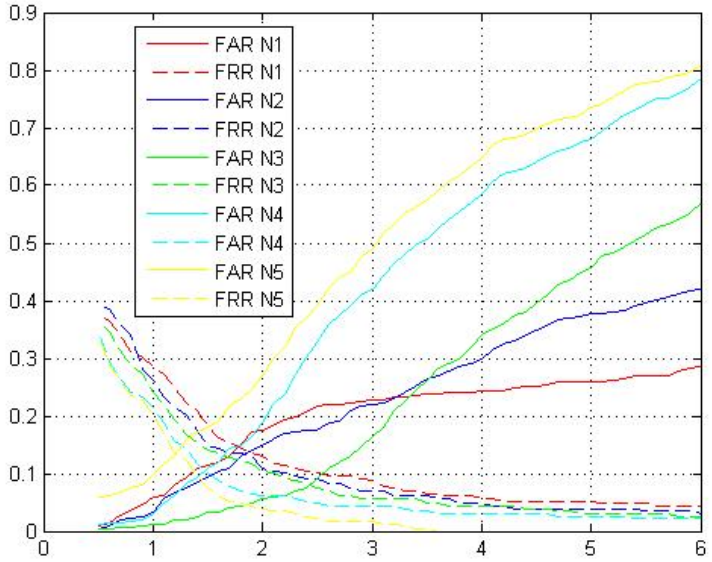

Figura 28 Gráfica comparativa análisis tipo 2 Wavelet Coiflet3 distintos niveles de aproximación

La tabla siguiente muestra un resumen de los resultados:

Tabla 6 EER para análisis tipo 2 Wavelet Coiflet3 distintos niveles de aproximación.

Nivel de coeficientes de aproximación coif3	EER	Tiempo promedio de procesamiento
1	14.68%	0.2118 segundos
2	13.43%	0.0743 segundos
3	8.05%	0.0326 segundos
4	10.58%	0.0217 segundos
5	13.44%	0.0202 segundos

Se observa que el nivel 3 de coeficientes de aproximación transformada wavelet discreta coiflet3 entrega el menor valor de EER con un 8.05%

4.2 Conclusiones

Con base en los resultados obtenidos de las pruebas descritas en la sección anterior para el sistema estudiado, es posible obtener las siguientes conclusiones:

El empleo de la información de magnitud y fase permitió reducir la EER en las pruebas de tipo 1 en un 5.14% en relación con la información correspondiente a la posición X e Y para la misma prueba.

La modificación del algoritmo DTW que incluye la información de los DMP's permitió reducir el EER en el análisis con firma completa en un 4.18%.

Los análisis de tipo 2 (Coeficientes de aproximación wavelet para la firma completa) haciendo uso de la DWT coiflet3 nivel 3 obtuvieron los mejores resultados en cuanto a la EER con un 8.05% para firmas falsificadas entrenadas.

El empleo de los coeficientes de aproximación wavelet como parámetros (análisis tipo 2) permitió reducir el tiempo promedio empleado en el análisis 22 veces y reducir la EER en un 4.52% comparado con el método que emplea todos los datos de la firma con el mismo propósito (análisis tipo 1).

En general, los análisis tipo 2 y tipo 4 que emplean la información de los trazos demostraron tener un desempeño más pobre que los que hacen uso de la firma completa.

4.3 Trabajo futuro

- Realizar cambios al algoritmo DTW propuesto con el propósito de que, además de la información actual, permita obtener información sobre las deformaciones temporales localizadas de la firma, pues en base al trabajo realizado se pudo comprobar que la diferencia entre una firma original y una falsificación está fuertemente marcada en las distorsiones existentes en áreas específicas de la firma así como en tiempo empleado para firmar, y se pudo comprobar que el método de DTW es una buena elección para determinar este tipo de parámetros debido a que es capaz de encontrar distorsiones en el alineamiento de las secuencias mediante puntos que no son DMP's.
- Modificar el sistema de verificación. Realizar pruebas con verificadores que hagan uso de redes neuronales de forma que el sistema presente un comportamiento más robusto.
- Construir un generador de plantillas que permita incorporar información sobe la wavelet madre así como el nivel de resolución de los coeficientes empleados durante el análisis. La idea es poder personalizar y mejorar el desempeño del sistema debido a que en la creación de la plantilla se determina que wavelet madre emplear, y esta información es empleada por el sistema cuando una firma de prueba es evaluada. Debido a que cada firma tiene características distintas con respecto a los trazos (líneas curvas, líneas oblicuas, múltiples líneas, etc.), el uso de una sola wavelet madre para obtener información de ellas resulta inapropiado, por lo que cada firma, una vez analizada contendrá información sobre la wavelet madre con la que obtuvo mejores resultados en la etapa de entrenamiento.
- Modificar el algoritmo de homologación de trazos, de forma que pueda cotejar de forma más eficiente los mismos, y generar una mejor comparación. En estas pruebas se pudo observar que el algoritmo empleado para obtener los "strokes" rechazaba firmas auténticas debido a que existen firmantes que tienden a levantar el bolígrafo en ciertas áreas donde sus trazos se vuelven más tenues, obteniendo una diferencia de más de 2 "strokes", lo que se incrementa la FRR y , por tanto, la EER. Se pretende diseñar un algoritmo que permita obtener la variabilidad de trazos para cada firmante en base a las firmas de entrenamiento.

APENDICE

A1 Programación dinámica

La siguiente sección es un breve resumen de la información contenida en [11] para ilustrar los fundamentos teóricos de la programación dinámica.

La programación dinámica es una herramienta ampliamente utilizada en operaciones de investigación para resolver problemas de decisión secuencial. Existen dos problemas representativos en los que esta técnica es utilizada: problema de decisión secuencial asíncrona y problema de decisión síncrona.

Suponiendo la existencia de una serie de puntos 1…N y trayectorias de interconexión entre ellos asociadas a una función de costo $\varsigma(i,j)$ que resulta de viajar directamente del punto i al punto j en un paso, el problema de decisión secuencial asíncrona consiste en encontrar un costo mínimo así como la secuencia de pasos necesarios para lograrlo, con el propósito de desplazarse de un punto, digamos el número 1 hasta el punto i en el número de pasos que sea necesario.

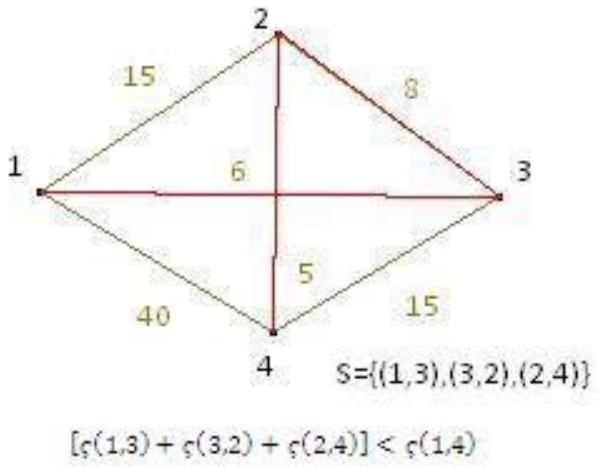

Figura 29. Ejemplo del problema de decisión secuencial asíncrona

La figura 29 ilustra un ejemplo de este problema. Las líneas que unen los puntos representan un posible movimiento de un punto a otro en un solo paso. Los números al

lado de las líneas representan su función de costo. En este ejemplo en particular se debe encontrar la trayectoria que minimiza la función de costo desde el punto 1 hasta el punto número 4. La solución al problema es el conjunto de pasos S y la trayectoria mostrada en color rojo.

El problema de decisión secuencial síncrono es similar al problema descrito anteriormente, teniendo como diferencia la regularidad del proceso de decisión. En este problema, el objetivo es encontrar la secuencia óptima de un número determinado de M movimientos para lograr un desplazamiento del punto i al punto j.

Con el objetivo de simplificar esta tarea se emplea una estructura de celosía ("trellis structure"), en la cual se disponen el número de puntos existentes N en forma vertical y se duplican en forma horizontal de acuerdo con el número de movimientos M que se hayan planeado (figura 30).

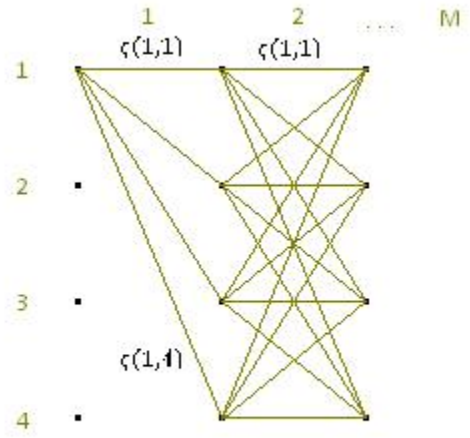

Figura 30. Estructura de celosía para el problema de decisión secuencial síncrona

Continuando con el ejemplo anterior, se pude plantear la siguiente pregunta: ¿Cual es el camino óptimo para ir del punto 1 al punto 4 en 2 pasos? La respuesta puede encontrarse mediante la estructura mencionada. Una opción para desplazarse del punto 1 al punto 4 sería mediante las trayectorias marcadas en color azul en la figura 31, pero tomando en

cuenta las funciones de costo de cada trayectoria se puede determinar que los movimientos marcados en rojo representan la trayectoria óptima.

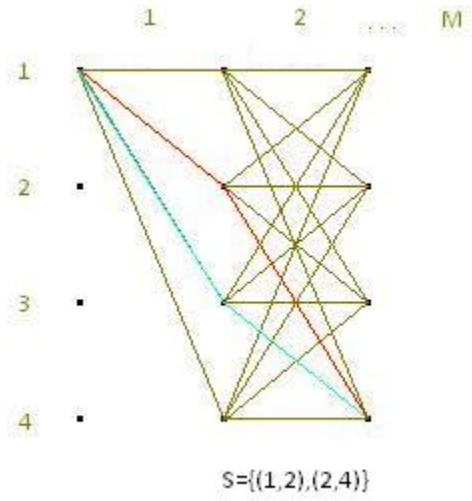

Figura 31. Solución al problema de ejemplo de decisión secuencial síncrona

Una política es la terminología tradicional empleada para referirse a una regla para determinar el próximo punto, después del punto i, a ser visitado. La pregunta que la programación dinámica trata de resolver es, ¿cuál de todas las posibles políticas lleva al mínimo costo desde el punto 1 al punto i? Esta función de costo mínima es llamada $\varphi(1, i)$.

La respuesta a esta pregunta, que representa la solución de los problemas expuestos anteriormente, puede encontrarse haciendo uso del principio de optimación, que de acuerdo con Bellman [20], indica:

“Una política óptima tiene la propiedad de que, sin importar el estado y la decisión inicial, las decisiones restantes deberán constituir una política óptima con respecto al estado resultante de la primera decisión.”

Tomando como base el principio de optimación de Bellman en el desarrollo de un algoritmo, primero se considera un movimiento del punto inicial 1 a un punto intermedio j en uno o más pasos. Por otra parte se denota $\varphi(1, i)$ la función de costo mínima que resulta de desplazarse del punto inicial 1 a un punto final i . Debido a que moverse de un

punto i a un punto j en un paso incurre en un costo $\varsigma(j,i)$, la política óptima que determina por cual punto intermedio se debe de pasar(debe de existir alguno) satisface la siguiente ecuación:

$$\varphi(1,i) = \min_{j}[\varphi(1,j) + \varsigma(j,i)] \qquad (A1.1)$$

Generalizando la ecuación al caso donde interesa la secuencia óptima de movimientos y la función de costo mínima asociada de para trasladarse desde un punto i cualquiera a otro punto j, obtenemos:

$$\varphi(i,j) = \min_{l}[\varphi(i,l) + \varphi(l,j)] \qquad (A1.2)$$

Para determinar la trayectoria de costo mínimo entre dos puntos i y j en cualquier número de pasos, se emplea el siguiente programa dinámico:

$$\varphi_1(i,l) = \varsigma(i,l) \qquad l = 1,2,\dots.N$$

$$\varphi_2(i,l) = \min_{k}[\varphi_1(i,k) + \varsigma(k,l)] \qquad \begin{matrix} l = 1,2,\dots.N \\ k = 1,2,\dots.N \end{matrix}$$

$$\varphi_3(i,l) = \min_{k}[\varphi_2(i,k) + \varsigma(k,l)] \qquad \begin{matrix} l = 1,2,\dots.N \\ k = 1,2,\dots.N \end{matrix}$$

$$\begin{matrix} . \\ . \\ . \end{matrix}$$

$$\varphi_S(i,l) = \min_{k}[\varphi_{S-1}(i,k) + \varsigma(k,l)] \qquad \begin{matrix} l = 1,2,\dots.N \\ k = 1,2,\dots.N \end{matrix}$$

$$\varphi(i,j) = \min_{1\leq s\leq S} \varphi_s(i,j)$$

donde N es el número total de puntos, $\varphi_s(i,j)$ es la s-esima mejor trayectoria del punto *i* al punto *l* y S el número máximo de pasos encontrados.

Para el ejemplo de la figura 2, podemos emplear el algoritmo recursivo anterior como sigue:

$$\varphi_1(1,2) = \varsigma(1,2) = 15$$
$$\varphi_1(1,3) = \varsigma(1,3) = 6$$
$$\varphi_1(1,4) = \varsigma(1,4) = 40$$

$$\varphi_2(1,2) = min\{\varphi_1(1,3) + \varsigma(3,2)\} = 14$$
$$\varphi_2(1,3) = min\{\varphi_1(1,2) + \varsigma(2,3)\} = 23$$
$$\varphi_2(1,4) = min\begin{Bmatrix}\varphi_1(1,2) + \varsigma(2,4)\\ \varphi_1(1,3) + \varsigma(3,4)\end{Bmatrix} = min\begin{Bmatrix}20\\ 21\end{Bmatrix} = 20$$

$$\varphi_3(1,4) = min\begin{Bmatrix}\varphi_2(1,2) + \varsigma(2,4)\\ \varphi_2(1,3) + \varsigma(3,4)\end{Bmatrix} = min\begin{Bmatrix}19\\ 38\end{Bmatrix} = 19$$

Por tanto, la trayectoria optima entre los puntos 1 y 4 será:

$$\varphi(1,4) = \min_{1\leq s\leq 3} \varphi_s(1,4) = \varphi_3(1,4) = 19$$

Haciendo uso de este método se puede observar que los resultados coinciden con la trayectoria mínima encontrada.

Finalmente, para el caso de encontrar la solución al problema de decisión secuencial síncrona, el principio de optimación sigue siendo aplicable. Después del movimiento m con m<M, donde M es el número de pasos máximo preestablecido, la trayectoria puede finalizar en cualquier punto l, l =1,2…,N con la función de costo asociada $\varphi(i,l)$. Suponiendo que en el movimiento m+1 se realiza un desplazamiento hacia el punto n, la función $\varphi_{m+1}(i,n)$ satisface:

$$\varphi_{m+1}(i,n) = \min_{l}[\varphi_m(i,l) + \varsigma(l,n)] \qquad (A1.3)$$

El algoritmo para la solución del problema de decisión secuencial síncrono puede resumirse en los siguientes pasos:

1. Inicialización

$$\varphi_1(i,n) = \varsigma(i,n)$$
$$\xi_1(n) = i$$
$$para\ \ n = 1,2 \dots, N$$

2. Recursión

$$\varphi_{m+1}(i,n) = \min_{1\leq l\leq N}[\varphi_m(i,l) + \varsigma(l,n)]$$
$$\xi_{m+1}(n) = \arg\min_{1\leq l\leq N}[\varphi_m(i,l) + \varsigma(l,n)]$$
$$para\ \ n = 1,2\ldots, N\ y\ m = 1,2,\ldots, M-2$$

3. Terminación

$$\varphi_M(i,j) = \min_{1\leq l\leq N}[\varphi_{M-1}(i,l) + \varsigma(l,n)]$$
$$\xi_M(j) = \arg\min_{1\leq l\leq N}[\varphi_{M-1}(i,l) + \varsigma(l,n)]$$

4. Retroceso de trayectoria

Trayectoria óptima = $(i, i_1, i_2, \ldots, i_{m-1}, j)$

donde:

$$i_m = \xi_{m+1}(m+1) \quad m = M-1, M-2, \ldots, 1$$

A2 Particularidades de la aplicación de adquisición de firmas

El software para la adquisición de las firmas se desarrollo haciendo uso de Visual Basic 6.0 de Microsoft. Con el propósito de poder emplear la tableta digitalizadora mencionada se hizo uso de la librería “Microsoft Tablet PC Type Library versión 1.0” incluida en el SDK Windows XP Tablet PC [14].

Dentro de la librería se incluye una clase denominada “InkCollector”; esta es la clase de interés que se empleó para el desarrollo de la aplicación. El objeto construido en base a esta clase permitió acceder a la información entregada por la tableta digitalizadora.

Una característica muy importante de esta clase es que los objetos creados con ella pueden enlazar su controlador de ventana con los objetos existentes en Visual Basic. Su importancia radica en que cualquier objeto de la ventana de la aplicación puede ser empleada como un “colector de tinta”, es decir, una zona en la que se puede emplear la tableta digitalizadora para escribir. En el caso de esta herramienta se optó por construir el “AREA DE FIRMADO” haciendo uso de un control normalmente empleado para la presentación de imágenes (“Picture Box”).

En Visual Basic las acciones vinculadas a algún objeto son ejecutadas cuando se produce algún tipo de evento. Nuestros objetos generados por la clase “InkCollector” no son la excepción, y para el caso de nuestra aplicación en particular fueron empleados dos de ellos:

- Evento “NewPackets”: Ocurre cuando llega un nuevo conjunto de datos provenientes de la tableta digitalizadora. Debido a que nuestra tableta es capaz de tomar un total de 100 puntos por segundo, este evento se produce con la misma frecuencia.
- Evento “Stroke”: Ocurre cuando se completa un trazo de la firma. De hecho, para el caso de esta librería un trazo completo consiste en la acción de bajar la pluma, obtener datos y el levantamiento de la pluma, sin embargo el evento se dispara en este último instante.

A continuación se presenta un resumen de las acciones ejecutadas en los eventos más significativos del programa construido:

Evento “NewPackets”

Es el evento principal que se presenta en el software. Ocurre cuando la tableta digitalizadora envía datos. Durante este evento, los datos correspondientes a la posición en x, la posición en y, y la presión son guardados en un arreglo de memoria.

Evento “Stroke”

Ocurre cuando se levanta la pluma de la tableta digitalizadora (“pen up”). Cuando se produce, se coloca una marca (1) en los datos del arreglo de memoria correspondiente a la información de “stroke”.

Evento Botón Guardar

Cuando se presiona este botón, se muestra el cuadro de dialogo guardar, donde el usuario indica el nombre del archivo a guardar. Una vez determinada esta información, se obtienen los valores máximo y mínimo de los datos sobre posición en x e y. Con esta información se procede a corregir los datos del eje y (el cero de este eje en la pantalla corresponde al margen superior del recuadro de captura) y a colocar el valor mínimo de ambas posiciones en cero. Una vez realizado esto se guardan los datos con el formato .csv [15] en un archivo con el nombre asignado y la misma extensión.

Evento Botón Limpiar

Durante este evento se limpia el control “InkCollector” ubicado en el cuadro de imagen y se borra el contenido del arreglo de datos.

A3 Descripción de la interface gráfica creada con MATLAB

La interface gráfica de la herramienta consta de una ventana principal que es la encargada de mostrar los controles para cada operación realizada con la herramienta.

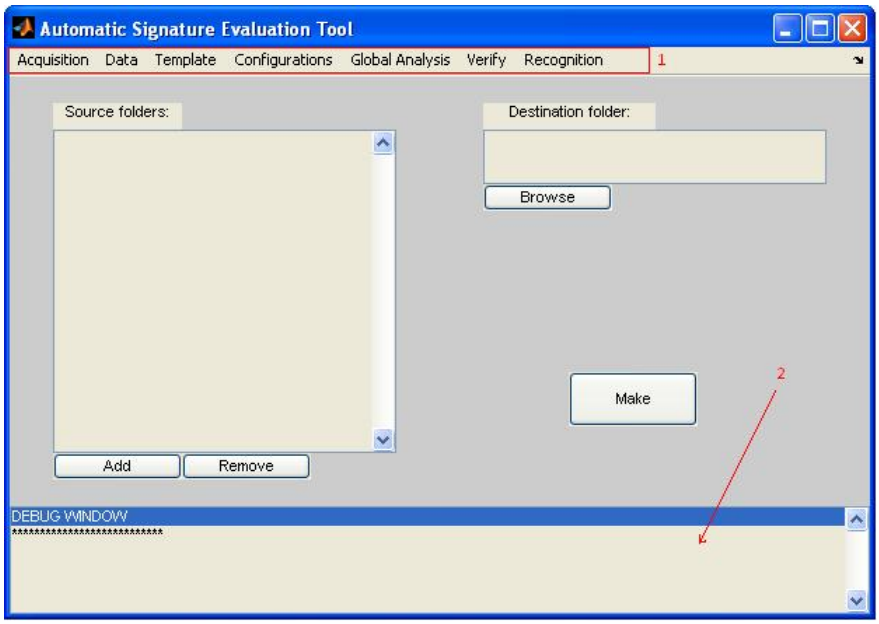

Figura 32. Interface gráfica desarrollada

La ventana principal cuenta con dos elementos comunes que aparecen durante toda la sesión de uso. Estas son:

1. Menú de tareas: Son las pestañas que se muestran en la parte superior de la ventana, por medio de la selección de alguna de ellas con el puntero del ratón se puede cambiar la tarea que se está realizando con la herramienta.
2. Ventana de depuración (DEBUG WINDOW): En esta ventana se muestran errores y mensajes de interés para el usuario con respecto a la tarea que se está realizando.

A3.1 Tareas implementadas

El sistema es capaz de llevar a cabo 7 distintas tareas implicadas en el procesamiento de las firmas. En las siguientes líneas se describen en detalle estas tareas.

A3.1.1 Tarea de Adquisición ("Acquisition")

Por medio de esta tarea se pude generar una carpeta que contenga la totalidad de las firmas organizadas de forma que las otras tareas implementadas en la interfaz puedan realizar las evaluaciones en forma automática.

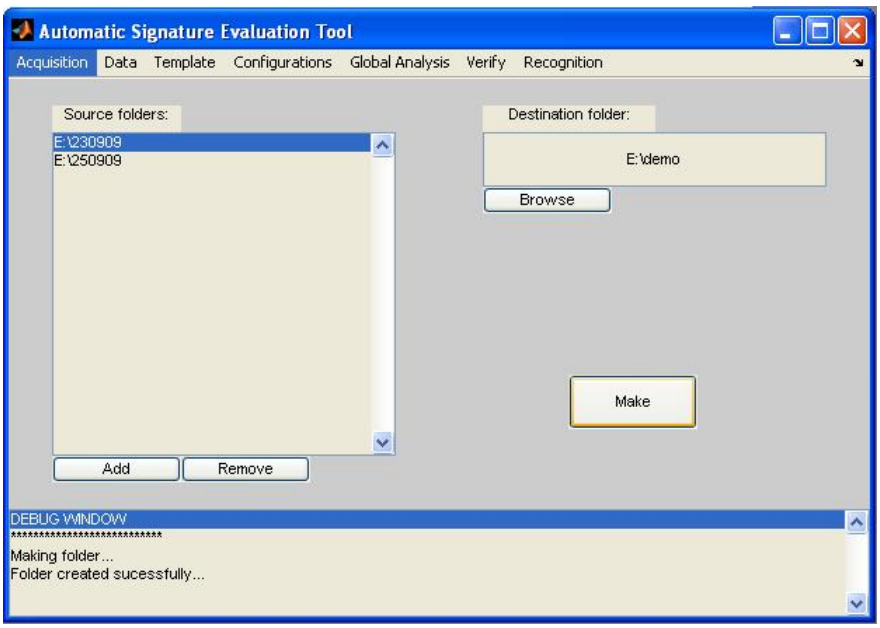

Figura 33. Ventana correspondiente a la tarea de adquisición

Durante la recopilación de firmas por medio de la tableta digitalizadora se nombró a los archivos haciendo uso de una letra para identificar al firmante y un número para identificar el número de firma. Debido a que se generaron varias carpetas fue necesario que un sistema automático tomara los archivos de estas carpetas y los almacenara en una sola carpeta. Ya que el número de firmantes total es mayor de 26 que es el número de letras en el alfabeto inglés, fue necesario cambiar los nombres de archivo, de forma que se identifica a un firmante con dos letras y el número de firma por un número (por ejemplo AA12 firmante 1 firma 12). Este es el objetivo de esta tarea, organizar y renombrar los archivos. Por defecto el número de firmas por firmante es de 20.

Para emplear esta tarea solamente es necesario indicar las carpetas donde se han guardado los archivos que han sido creados con la herramienta de adquisición respetando los nombres de archivo, en el campo marcado como "Source folders". Para agregar más

carpetas se emplea el botón “Add” y en caso de haber colocado una carpeta incorrecta esta se puede remover por medio del botón “Remove”.

En el campo “Destination folder” debe colocarse la ruta de la carpeta donde se colocaran los archivos renombrados. Para buscar dicha ruta se hace uso del botón “Browse”.

Finalmente para generar la carpeta organizada se debe hacer uso del botón “Make”

A3.1.2 Tarea de obtención de datos (“Data”)

Por medio de esta tarea de la interface se cargan los datos de las firmas en la memoria del programa de evaluación de firmas.

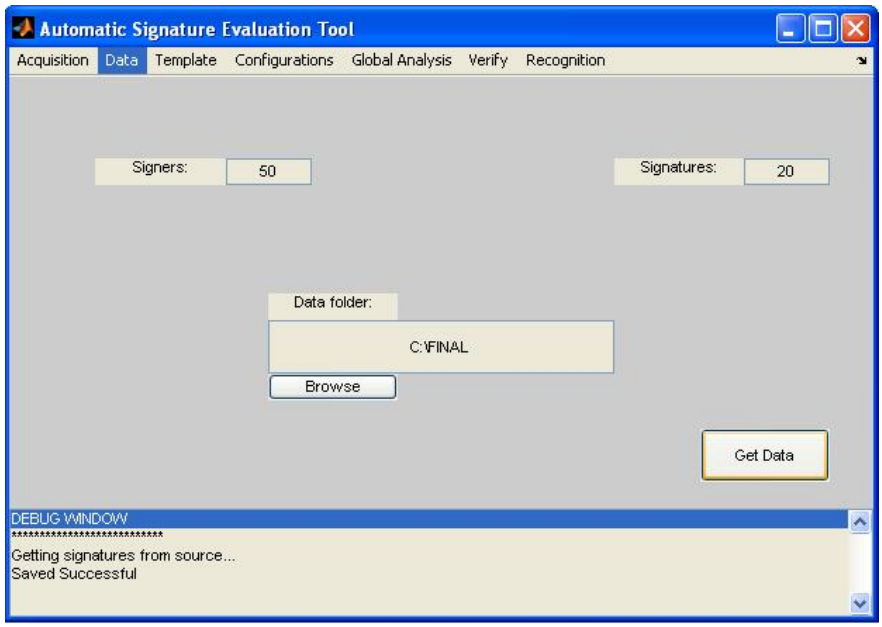

Figura 34. Ventana de la tarea de obtención de datos

Para emplear esta tarea solamente es necesario indicar el número de firmantes y de firmas que serán cargadas en la memoria. Esto se logra indicando dichos números en los campos “Signers” y “Signatures” respectivamente. En el campo “Data folder” debe indicarse la ruta donde los archivos se encuentran. Estos deben estar nombrados en la forma descrita en la tarea anterior.

Por medio del botón “Get data” se indica al programa que puede proceder a cargar los datos.

A3.1.3 Tarea de generación de plantillas ("Template")

Por medio de esta tarea se generan las plantillas para cada firmante siguiendo el procedimiento descrito en la sección "Obtención de plantillas" de la sección 3.3.4.1 El usuario puede indicar que firmas del conjunto que fue guardado en memoria serán usadas como candidatas para la selección de la plantilla.

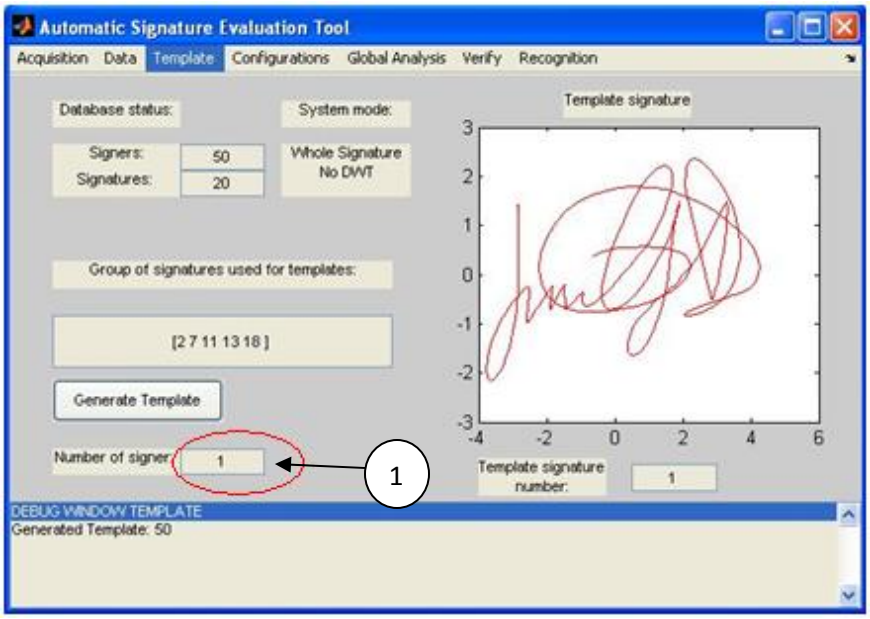

Figura 35. Ventana de la tarea para generación de plantillas

Para usar esta tarea basta indicar el número de las firmas que se han propuesto para generar la plantilla en el campo "Group of signatures used for templates". Una vez indicado esto se pueden generar las plantillas haciendo uso del botón "Get Template".

En la parte inferior del botón existe un campo, marcado como (1) en la figura 35, con el cual se puede visualizar la plantilla para un firmante dado, solo basta ingresar el número del firmante del cual se desea conocer la plantilla.

Es necesario aclarar que antes de proceder a la creación de las plantillas es necesario configurar el tipo de prueba y sus parámetros para obtener las plantillas correctas. Lo anterior puede ser logrado mediante la siguiente tarea.

A3.1.4 Tarea de ajuste de configuraciones ("Configurations")

Con esta tarea se pueden ajustar las configuraciones de prueba. Solo basta con llenar cada campo con los datos deseados para la prueba. El primer campo corresponde con el tipo de análisis. Este campo solo puede contener los números del 1 al 4 que corresponden con los tipos descritos en la sección 3.4. Por defecto este campo está marcado con 1. En el segundo campo se debe indicar la Wavelet madre que será empleada para obtener los coeficientes de las secuencias de tiempo en las pruebas que así lo requieran. Por defecto en este campo se encuentra marcado db5 (Wavelet madre Daubechies con 5 momentos evanescentes). En el tercer campo se indica el nivel de aproximación de los coeficientes, inicializado por defecto en 3. El último campo es usado en la determinación del nivel de disparo y corresponde al factor de ponderación de la desviación estándar indicado en la sección 3.3.5.

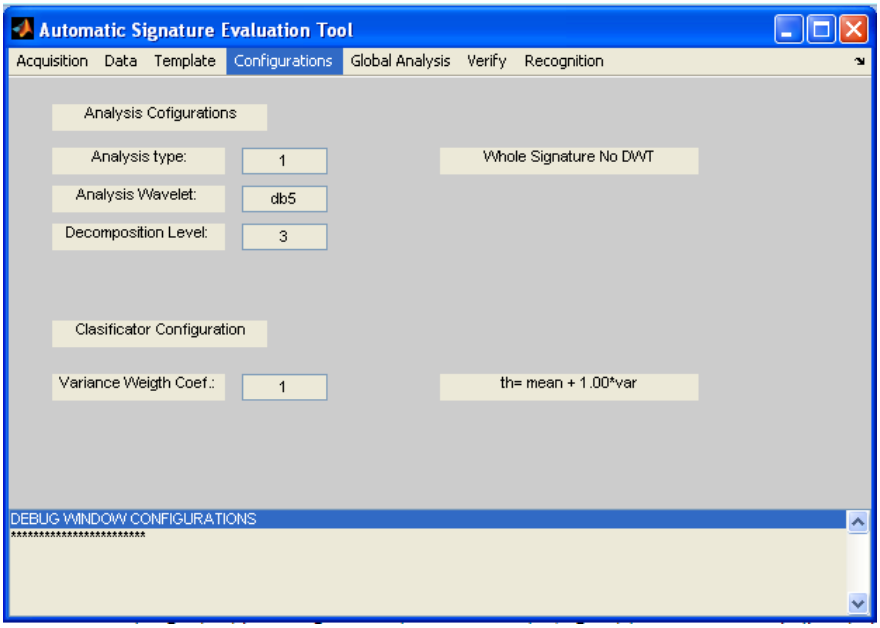

Figura 36. Ventana de la tarea de ajuste de configuraciones

A3.1.5 Tarea de análisis global ("Global Analysis")

Una vez efectuados los procesos de guardado en memoria de firmas y los firmantes, la configuración de las pruebas y la generación de plantillas es posible hacer un análisis

global entre todas estas firmas. Para ello basta, solo en necesario usar el botón marcado como “Go”.

Este análisis realiza todas las pruebas indicadas en la sección de Metodología (1.5).

Cuando el sistema finaliza su ejecución indica al usuario si desea guardar los resultados. En caso de ser así, el sistema presenta un cuadro de dialogo en el cual el usuario guarda el nombre del archivo. Estos archivos son guardados con extensión .mat, lo que permite emplearlos de forma directa en MATLAB.

Cuando es cargado en MATLAB, el archivo indicado genera una variable llamada statis, que contiene una matriz de K filas, donde K representa al número de firmantes con 6 columnas, las cuales contienen la siguiente información:

Numero de firmante	Firmas falsas reconocidas como verdaderas	Total de comparaciones para la prueba de falsa aceptación	Firmas verdaderas reconocidas como verdaderas	Total de comparaciones para la prueba de falso rechazo	Tiempo promedio de procesamiento para cada firmante

Debido a que algunos procesos pueden tardar demasiado, en la parte derecha de la ventana existe un visualizador que permite determinar que firma se encuentra bajo prueba y su número aparece en la ventana de depuración:

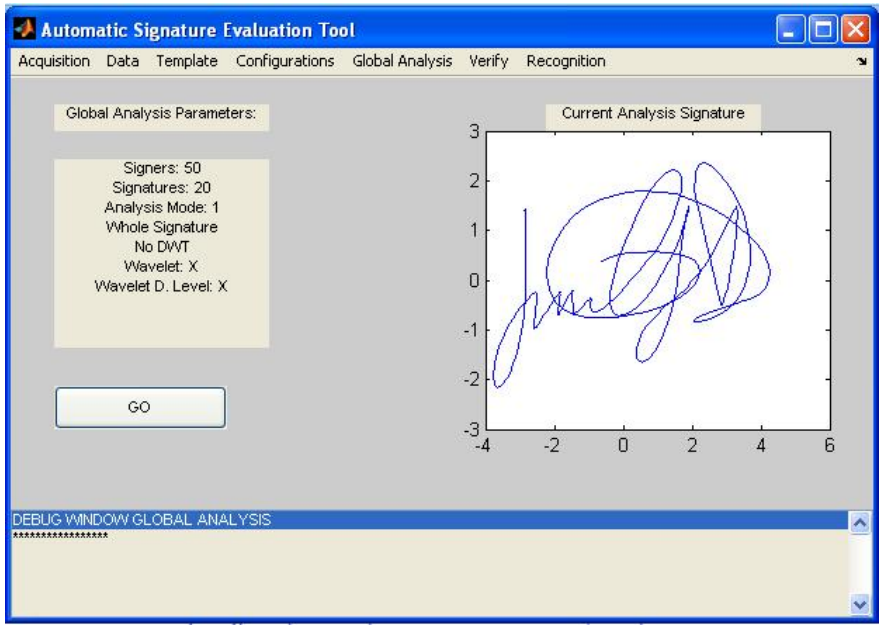

Figura 37. Ventana de la tarea de análisis global

A3.1.6 Tarea de verificación ("Verify")

Esta tarea se trata de un ejemplo de aplicación del sistema desarrollado. Para emplearla es necesario haber realizado los mismos pasos que en el proceso de análisis global, esto es:

1. Guardar el número de firmas y firmantes en la memoria, que serán considerados en el proceso. Esto se logra por medio de la tarea "Data".
2. Configurar el tipo de prueba a realizar y sus parámetros por medio de la tarea "Configurations".
3. Generar las plantillas, por medio de la tarea "Template".

La ventana de la Tarea de verificación cuenta con cuatro campos que deben ser llenados por el usuario. Los dos primeros corresponden con el número de firmante y de firma que se emplea como firma de prueba, indicándose estos valores en los campos marcados como "Signer" y "Signature" respectivamente. También es posible obtener la firma de prueba desde un archivo, siempre y cuando este cumpla con las características indicadas

en la sección 3.2.4.1. Para obtener la firma de prueba desde un archivo se emplea el botón “Browse”.

En el campo “Claim. Signer” se indica el número de firmante del cual se usará la plantilla en el proceso de verificación.

Finalmente en el campo “Std. Weight” se puede indicar el valor de ponderación de la desviación estándar para fijar el nivel de disparo (sección 3.3.5).

Para iniciar el proceso de verificación se emplea el botón “verify”. Al final del proceso, se indicara el resultado de la comparación en el cuadro que se encuentra al lado del botón y las estadísticas podrán observarse en la barra de depuración.

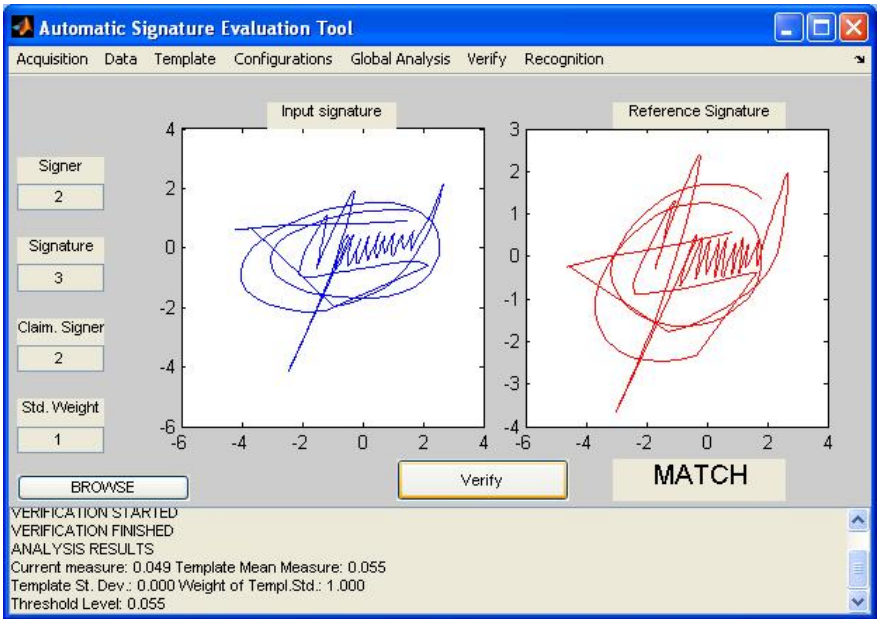

Figura 38. Ventana de la tarea de verificación

A3.1.7 *Tarea de reconocimiento (“Recognition”)*

Esta tarea es similar a la anteriormente descrita, con la diferencia de que el sistema buscara de forma automática entre las plantillas enlistadas con el objetivo de encontrar una correspondencia. De ser así, al igual que en la tarea anterior, esta indicación aparecerá en el cuadro que se encuentra a la derecha del botón “Recognize” indicando

cual ha sido la correspondencia y las estadísticas podrán observarse en la ventana de depuración.

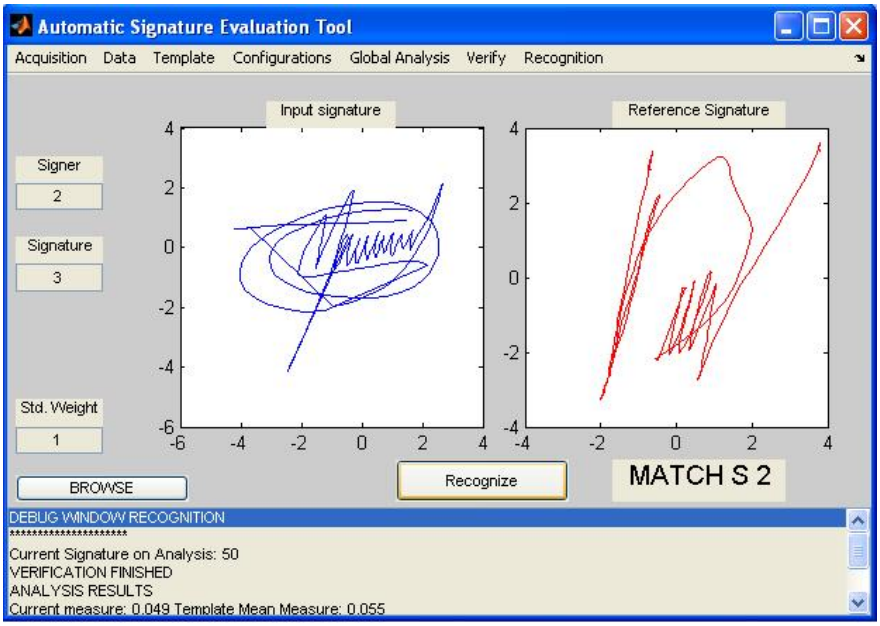

Figura 39. Ventana de la tarea de reconocimiento

REFERENCIAS

[1] Shoniregun C. A, Stephen C.(2008), Securing Biometric Applications. Londres: Springer

[2] D. Impedovo, G. Pirlo, "Automatic signature verification: The state of the art", *IEEE Trans. Syst. Man, Cybern. C.* Vol. 38, no. 5 Sep. 2008

[3] I. Nakanishi, N.Nishiguchi, Y. itoh, Y. fukui, "On line signature verification utilizing feature extraction based on DWT", *Proc. 2003 Symp. Circuit Syst.* (ISCAS'03), May, vol 4. Pp 73-76

[4] L. Bovino, S. Impedovo, G. Pirlo y L. Sarcinella, "Multiexpert verification of handwriten signatures", *Proc. 7th. Intl. Conf. Doc. Anal. Recognit.*(ICDAR-7), IEEE Comput. Soc. Edimburg, U.K. Aug. 2003. Pp. 932-936

[5] K. Huang, H. Yan. "Stability and style variation modeling for on line signature verification", *Pattern. Recognit*. vol. 36, no. 10 pp 2253 -2270, Oct. 2003.

[6] National Check Fraud Center, "Credit card Fraud Prevention Techniques and information" http://www.ckfraud.org/credit_card.html

[7]The shop.org community, Abril 2008, "On line sales to climb despite struggling economy according shop.org/ Forrester Research Study" http://www.shop.org/c/journal_articles/view_article_content?groupId=1&articleId=702&version=1.0

[8] BBC News, "Plastic Card fraud goes back up", http://news.bbc.co.uk/2/hi/business/7289856.stm

[9]Fortuna Negocios y Finanzas, Febrero 2008, "Fraude en tarjetas de crédito por 100 millones de dolares", http://revistafortuna.com.mx/contenido/index.php/2008/02/15/fraude-en-tarjetas-de-credito-por-100-millones-de-dolares/

[10] Dan Ellis, "Dynamic Time Warp in Matlab", http://labrosa.ee.columbia.edu/matlab/dtw/

[11] Rabiner L., Juang B. (1993), Fundamentals of Speech Recognition. EUA: Prentice Hall, 200-229

[12] Lokenath D., Mikusinski P. (2005), Introduction to Hilbert Spaces with applications, EUA: Elsevier Academic Press.

[13] Daubechies I. (1992), Ten lectures on wavelets, EUA: SIAM

[14] Microsoft Corporation, "Página de descarga de Windows XP Tablet PC SDK", http://www.microsoft.com/downloads/details.aspx?FamilyId=B46D4B83-A821-40BC-AA85-C9EE3D6E9699&displaylang=en

[15] The Internet Engineering Task Force (IETF), "Common Format and MIME Type for Comma-Separated Values Files", http://tools.ietf.org/html/rfc4180.

[16] A. Vergara Da Silva, D. Santana de Freitas, "Wavelet –Based compared to Function - Based On line Signature Verification", *proc. XV Brazilian, Symp. Comput. Graph. Image Process* (SIBGRAPI -XV), Fortaleza Brazil, Oct. 2002 p. 218

[17] V.Dilecce, G. Dimauro, A. Guerreiro, S. Impedovo, G. Pirlo and A. Salzo, "A Multiexpert system for dynamic signature verification," in Proc. 1st Int. Workshop, Multiple Classifier Syst. (MCS2000) (Lecture Notes in Computer Science), vol. 1857, J. Kittler and F. Roli. Eds. Berlin, Germany: Springer-Verlag, Jun.2008, pp. 320-329.

[18] Berrin Yanikoglu, Alisher Kholmatov, "On Line Signature Verification Using Fourier Descriptors", in EURASIP Journal on Advances in Signal Processing, vol. 2009, Oct. 2008.

[19] C. Gonzalez, E. Woods (2002), Digital Image Processing. EUA: Prentice Hall

[20] Bellman R.E. (1957), Dynamic Programming, EUA: Princeton University Press.

[21] H. Saoke, S. Chiba, "Dynamic programming optimization for spoken word recognition", IEEE Trans. Acoustics, Speech, Signal Proc.,ASSP-26(1): 43-49, Febraury, 1978.