

## Reconocimiento de caracteres manuscritos mediante información del proceso inverso en la realización de su trazo

### Character recognition system using the inverse order stroke information

*Karina Toscano\**, Humberto Sossa , Ricardo Barrón, Gabriel Sánchez

Centro de Investigación en Computación–IPN. Av. Juan de Dios Batiz, esquina con Miguel Othón de Mendizábal, Ciudad de México, 07738, México

(Recibido el 9 de septiembre de 2008. Aceptado el 26 de marzo de 2009)

#### Resumen

En este trabajo se describe una metodología para el reconocimiento de caracteres manuscritos mediante información del trazo en orden inverso descrito a través de ondículas. La información para la reconstrucción y reconocimiento del trazo se hace mediante la extracción de los llamados nodos óptimos. En este trabajo se utilizaron 20 nodos. Como función de aproximación se usó la función *spline* natural llamada *slalom*. Los experimentos se realizaron con tres clasificadores: una red neuronal, una máquina de vector soporte y un modelo de mezclas Gaussianas. El sistema se evaluó con una base de datos de siete escritores con 50 trazos por cada carácter del alfabeto inglés. La tasa de reconocimiento global al usar los tres clasificadores oscila entre 98 y 98,7 %.

----- *Palabras clave*: Reconocimiento de caracteres, RNA, SVM, GMM.

#### Abstract

In this paper a novel feature extraction methodology for character recognition by means of the information in inverse order stroke information using wavelets is proposed. The information used to reconstruct and recognize the stroke is done by using the so-called optimal knots. In this paper 20 optimal knots were used. As approximating function we decided to use the slalom natural spline function. The recognition experiments were carried out using the obtained feature vector as the input to some recognition systems based on Neural Network , Support Vector Machines and Gaussian Mixture Models for comparisons purposes. The proposal was evaluated with a database of

---

\* Autor de correspondencia: teléfono: + 52 + 55 + 572 96 000 Ext. 73207, fax: + 52 + 55 + 565 62 058, correo electrónico: likatome@calm-ecac.esimecu.ipn.mx (L. K. Toscano)

seven writers with 50 traces of each English character. The global recognition rate when using the three recognition strategies varied between 98 and 98.7 %.

----- *Keywords:* Character recognition, RNA, SVM, GMM.

## Introducción

La escritura es uno de los medios de comunicación más conocidos. El reconocimiento de escritura en línea gana nuevamente interés, no solo por el número de aplicaciones donde este recurso es utilizado, sino por el aumento de nuevos dispositivos de captura que han sido desarrollados en la actualidad [1]. Las metodologías generales en reconocimiento de patrones y procesamiento de imágenes se analizan en [2], mientras que las empleadas en el reconocimiento de caracteres en línea se revisan en [3,4]. La mayoría de los investigadores han elegido los caracteres numéricos para su investigación [5,6]. Por lo que se puede observar que existe una cierta madurez para el reconocimiento aislado de dígitos. Sin embargo, cuando se habla del reconocimiento de caracteres alfabéticos, el problema llega a ser más complicado. Uno de los problemas que se tiene es el número de las clases que se pueden llegar a tener, en el caso del alfabeto inglés es 52 caracteres, dependiendo si es mayúscula (A-Z) o minúsculas (a-z). Debido a esto, existe un número más grande de caracteres alfabéticos ambiguos que de números. En el reconocimiento de caracteres se presentan otros problemas como: múltiples patrones para representar un solo carácter, la representación intrínseca de la letra cursiva, y el número de trazos para realizar un carácter [7]. Este tema se ha explorado poco por ser complicado, por lo que se dice que el reconocimiento de caracteres es aún un problema de frontera [8].

Existen dos tipos de sistemas de reconocimiento: Los sistemas fuera de línea y los sistemas en línea. En este trabajo se estudia el desempeño de un sistema en línea. Los trazos se capturan durante el proceso de escritura. Esto permite obtener información como: el orden de los trazos, momentos gráficos, presión, velocidad, entre

otras [9-11]. Cuando un escritor realiza el trazo de un carácter: 1) Tiene en mente el carácter que se va a escribir, 2) Conoce el orden de los movimientos del carácter, 3) Sabe la realización del trazo y 4) Tiene una imagen de dicho carácter. El proceso de reconocimiento puede ser realizado en orden inverso al de la generación, es decir del último paso al primero. Sin embargo la realización de este proceso inverso es muy complicada. Los sistemas de reconocimiento de caracteres manuscritos son equivalentes en forma inversa cuando el escritor realiza un carácter (tiene en mente el símbolo que se va a escribir, el orden de los movimientos del carácter y la realización del trazo). Los sistemas de reconocimiento fuera de línea, por otro lado, equivalen al proceso inverso completo es decir además de los tres pasos anteriores se tiene la imagen del carácter. Muchos sistemas para el reconocimiento de caracteres en línea han sido reportados en la literatura. Algunos de estos se describen brevemente a continuación. Namboodiri et. al [9] propusieron un sistema para clasificar palabras y líneas en un documento manuscrito en línea en uno de los seis alfabetos más antiguos: Árabe, Cirílico, Hebreo, Romano, Devnagari (Hindú), Chino. La clasificación se basó en 11 características espaciales y temporales. El sistema propuesto alcanza un porcentaje de clasificación de 87,1 % a nivel palabra para el caso de una base de datos de 13379 palabras. El porcentaje sube a un 95% conforme el número de palabras en la muestra fue aumentado a cinco. El desempeño sube a 95,5% para el caso de líneas de texto completo que consta de una media de siete palabras. Kato et al [12], describen un método para recuperar el orden del trazo de una imagen estática 2D. El trazo se debe realizar en un solo movimiento y puede incluir líneas doblemente trazadas. El método de reconstrucción consiste en 2 fases: en la primera fase se analiza el grafo

construido del esqueleto de la imagen; éste es enseguida etiquetado para determinar los tipos de cada borde. En la segunda fase se traza el grafo desde la primera vértebra hasta la última usando la información de etiquetado. Al recuperar el orden del trazo, se puede obtener la información temporal de una imagen 2D estática. Por lo tanto, este método podría ser usado como puente para el reconocimiento de caracteres cursivos fuera de línea y el de línea. Mezghani et. al [13], proponen un sistema de reconocimiento de caracteres arábigos usando mapas auto-organizados de Kohonen. En este caso de la información dinámica de las letras arábigas, se extraen los coeficientes de Fourier elípticos. Los autores evaluaron el sistema usando 18 letras arábigas, cuando los escritores trazaron cada letra 24 veces obteniendo así 7400 trazos. Los resultados del reconocimiento varían demasiado dependiendo de la letra (desde un 40% hasta un 2% de error) debido a la similitud existentes entre las diferentes letras. El reconocimiento global para solo un escritor es aproximadamente del 88%. Yokobayashi et. al [14, 15], propusieron una nueva técnica de segmentación y reconocimiento de caracteres con una amplia variedad de degradaciones en la imagen y fondos complejos en escenas naturales. Evaluaron el sistema con un total de 698 imágenes de prueba extraídas de ICDAR 2003 OCR. Clasificaron las imágenes en siete grupos según el grado de degradación de imagen y/o la complejidad de fondo. El reconocimiento promedio oscila entre 70,3% y 95,5% para imágenes claras a 24,3% para imágenes de poco contraste.

### Experimentación

El reconocimiento de cualquier tipo de patrones: caracteres ópticos, olores, objetos en el espacio, rostros, voz, etc., queda englobado en el reconocimiento de patrones. La diferencia estriba principalmente en el captor utilizado, así como, en las técnicas empleadas para extraer las características del patrón a reconocer. Para el reconocimiento de caracteres manuscritos específicamente se propone utilizar la arquitectura que se muestra en la figura 1.



Figura 1 Sistema propuesto

Este artículo se enfoca en la realización de los procesos inversos antes mencionados para el caso de sistemas en línea. La realización del trazo se basa en una aproximación mediante la función *spline* natural (SLALOM). A través de esta función es posible obtener el orden del movimiento requerido para realizar el trazo del carácter a partir de los datos capturados por medio de una tarjeta digitalizadora. Estos datos generados se procesan para encontrar características propias de cada carácter manuscrito, se les efectúa un pre-procesamiento y posteriormente, se realiza el reconocimiento usando 3 clasificadores independientes que arrojan cual carácter se trazó. Uno de los objetivos primordiales de este trabajo es comparar y evaluar el desempeño del sistema con 7 escritores y con varios clasificadores. El trabajo más parecido, reportado en [16], prueba el desempeño de sólo clasificador tipo estadístico y un escritor.

### Adquisición de datos

La información de los caracteres se adquiere por medio de una tableta digitalizadora *Intuos 2 Wacom*. Sobre esta tarjeta, los escritores trazan los caracteres. De esta forma se conoce el orden de articulación de cada carácter. La tableta arroja la imagen de lo que se escribe en el monitor, así como los datos según el orden de articulación como se escribe un carácter.

La tableta digitalizadora muestra los datos de lo que se escribe: la posición de la pluma, representada por vectores de datos en forma vertical,  $x(t)$ ,  $y(t)$ ,  $z(t)$ ; donde  $x(t)$  son los valores con respecto al eje  $x$ ,  $y(t)$  son los valores con respecto al eje  $y$

y  $z(t)$  es la información de la presión que ejerce la pluma al escribir sobre la tableta. Estos datos proporcionados por la tableta digitalizadora se conocen como características dinámicas, debido a que se obtienen en el momento que se realiza la escritura.

Con la información que se capturó se genera la forma de onda de los caracteres en  $x(t)$  e  $y(t)$ , así como también la velocidad del trazo, que es el tiempo en que una persona realiza el trazo de un carácter. En este trabajo se utilizó una base de datos de caracteres manuscritos usando la tableta digitalizadora. Se tomaron en cuenta 26 letras o caracteres del alfabeto inglés, las cuales son: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z. Cada carácter fue realizado 50 veces por 7 escritores diferentes, obteniéndose un base de datos de 9100 trazos. De estos, 6370 fueron usados para el entrenamiento (35 veces cada carácter) y 2730 (15 veces cada carácter) fueron utilizados para la prueba de los clasificadores.

### Wavelets

La teoría de las wavelets se basa en una representación local de frecuencias de la señal. Esto permite localizar, e identificar, tal acumulación de ondas pequeñas y ayuda a entender mejor las razones de los fenómenos estadísticos. Además de las propiedades de localización, las wavelets poseen también excelentes propiedades para su suavizado. Como se propuso en [16], en donde se observó que la señal, aún después del filtrado no era tan suave. Por esta razón se le aplicó una *wavelet Daubechies 1* para suavizarla. Se utilizó solo la aproximación para su reconstrucción. La figura 2 muestra el carácter ‘u’ después de su captura en la tableta digital y después de aplicarle la *wavelet db1*.

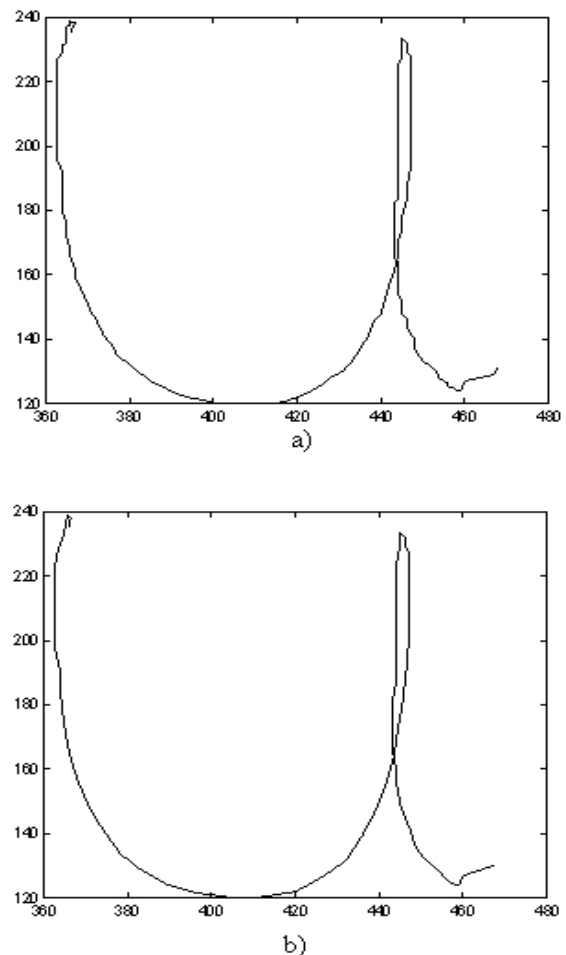
### Pre-procesamiento

La etapa de pre-procesamiento se aplica para reducir la influencia del ruido y algunas deformaciones sobre la señal con el objeto de que los rasgos descriptores extraídos sean más robustos. Además de las variaciones geométricas mencio-

nadas, generalmente los datos capturados contienen componentes de alta frecuencia introducidos por una pequeña vibración de la mano durante el trazado del carácter. Una vez que se tienen los caracteres capturados, se propone el siguiente pre-procesamiento:

- Filtrado, y
- Normalización (posición, tamaño y tiempo).

Asumiendo que el pre-procesamiento se realiza para ambas formas de onda  $x(t)$ ,  $y(t)$ .



**Figura 2** Ejemplo del carácter u a) como se captura con la tarjeta digitalizadora y b) usando *wavelet db1*

### Filtrado

Para esta primera parte del pre-procesamiento se utilizó un filtro para eliminar el ruido que se ge-

nera por el ruido-biomecánico, que se presenta al inicio o al final de la captura del trazo, el filtrado, a veces llamado adelgazamiento, elimina los píxeles de datos duplicados por lo que reduce el número de píxeles del carácter, obteniendo un carácter delineado y sin ruido, como se muestra en la figura 3.

#### Normalización en tamaño

Tomando en cuenta el área de trabajo y la diversificación del tamaño de los caracteres, se define un valor promedio del número de píxeles en el eje  $x$ , y en el eje  $y$ , es decir el ancho y altura del carácter. Para esta investigación este valor es de (215x180) píxeles. Éste es utilizado para llevar a cabo la normalización. Para este pre-procesamiento se obtiene el ancho y la altura de cada carácter, para normalizarlo se toma en cuenta la longitud mayor, ya sea el ancho o la altura, con su valor promedio de píxeles dado respectivamente.

#### Normalización en posición

Para llevar a cabo la normalización en posición se calcula el valor máximo y mínimo en los ejes  $x$ ,  $y$ ; con estos datos se obtiene el promedio de cada uno de los ejes, el cual se resta al eje correspondiente, obteniendo el carácter centrado con valores positivos y negativos para los ejes  $x$ ,  $y$ .

#### Normalización en tiempo

Dependiendo del carácter y la velocidad con que fue escrito, el número de muestreo varía, por lo que hay que unificar el número de muestreo a un valor determinado  $N$ , para esta investigación fue de 146. Se obtiene el número de muestreo, y si éste es mayor a  $N$  se realiza la decimación, es decir, se eliminan datos de manera uniforme y si es menor a  $N$  se realiza la interpolación, se agregan datos de manera uniforme. La figura 3 muestra el carácter  $u$ , normalizados en tamaño, posición y tiempo.

#### Extracción de características

La extracción de características se refiere al proceso que consiste en formar rasgos que permiten

describir la forma del objeto u objetos a reconocer. En esta investigación se extrajeron las siguientes dos características:

- Los nodos óptimos del eje  $x(t)$ .
- Los nodos óptimos del eje  $y(t)$ .

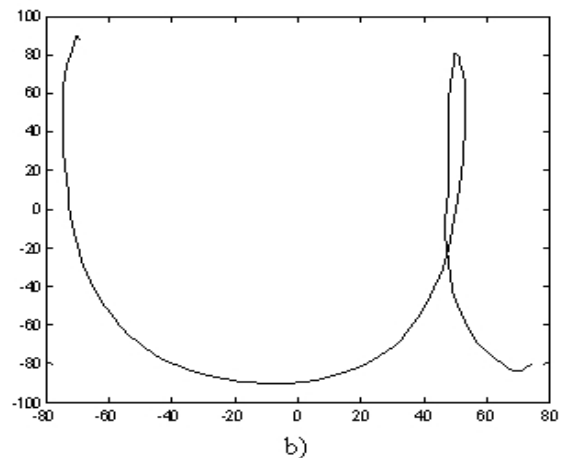
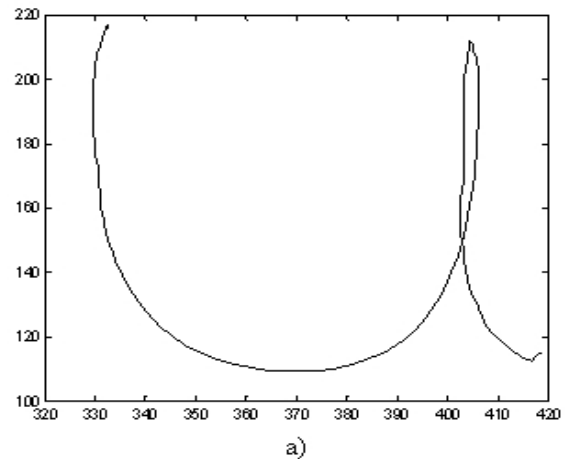


Figura 3 a) Carácter 'u' filtrado b) Carácter 'u' normalizado en tamaño, posición y tiempo

#### Obtención de nodos óptimos

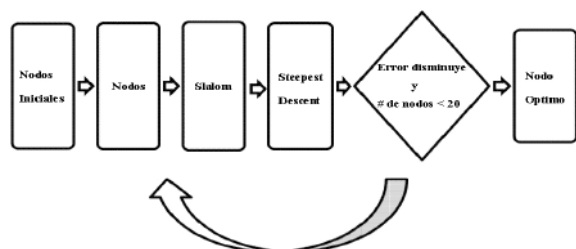
Para obtener los nodos óptimos de cada carácter se realiza el proceso que se muestra en la figura 4.

#### Inicialización de nodos óptimos

Los datos del trazo del carácter son valores positivos y continuos, por lo que se pueden derivar



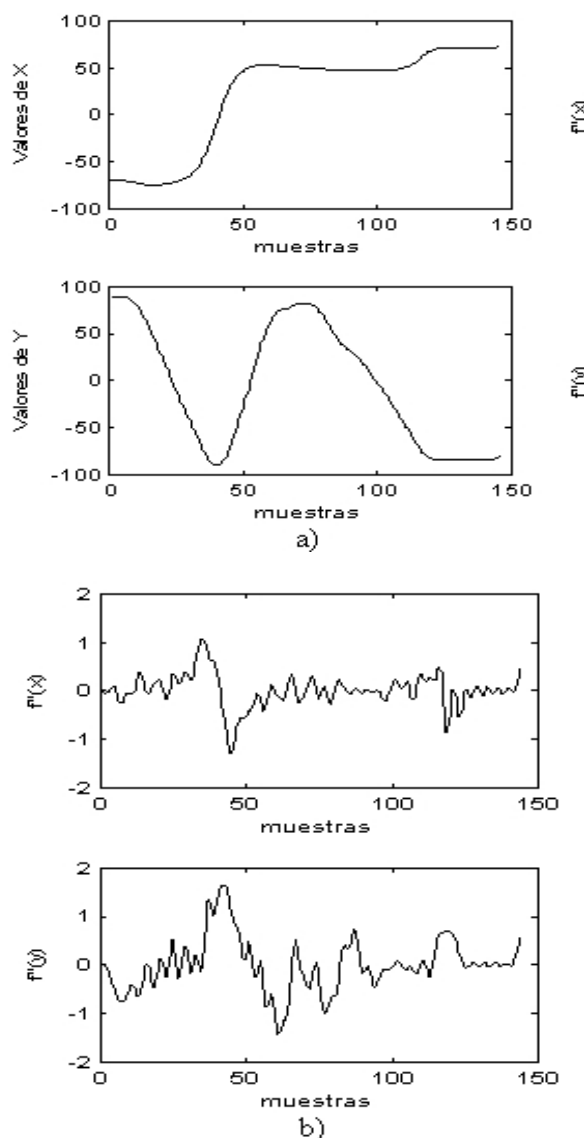
y mantener su continuidad. Para conocer las posiciones donde se cambia la dirección de la escritura de cada carácter (en el eje  $x$  de derecha a izquierda o de izquierda a derecha, en el eje  $y$  de arriba a abajo o de abajo a arriba), se calcula la primera derivada de la señal de cada eje. Los datos de la segunda derivada de las señales del carácter en el eje  $x$  e  $y$  nos da la información acerca de los cambios de la velocidad (aceleración) de la escritura, ya que escribimos con diferentes velocidades, es decir, los valores positivos de la segunda derivada significa que la velocidad de la escritura está aumentando y los valores negativos significa que va disminuyendo. La figura 5 nos muestra las formas de onda del carácter 'u' que son los valores que arroja la tableta  $x(t)$ ,  $y(t)$ . Como podemos observar son datos positivos y continuos; posteriormente se muestra la segunda derivada del carácter u. La señal obtenida de la segunda derivada, muestra pequeños cambios de velocidad que puede ser ruido producido por la vibración de las manos. Para evitar esto y obtener una señal más suave se necesita un filtrado, que se lleva a cabo mediante un filtro pasa-bajas, que se muestra en la figura 5.



**Figura 4** Proceso para la extracción de características

Con la señal filtrada de la segunda derivada se forman vectores con valores positivos y negativos. Cada cambio de signo en los valores se le llama cruce por cero debido a que en cada cambio de signo, ya sea de positivo a negativo o viceversa, se pasa por el cero. Se obtienen los máximos y mínimos de cada cruce por cero, a los que llamaremos nodos locales de cada carácter. La figura 6 muestra la división de la señal del carácter u, y sus nodos locales. Se toma el punto inicial y el

punto final de cada carácter con respecto al tiempo. Estos se toman como el nodo inicial y final de cada carácter al agregar los nodos locales ya obtenidos, obteniéndose así los nodos iniciales. Estos nodos iniciales no son óptimos, ya que los errores entre las señales reconstruidas usando estos nodos y señales originales son considerables.



**Figura 5** a) Señales correspondientes al carácter 'u' después del pre-procesamiento. b) Segunda derivada filtrada por un filtro pasa-bajas correspondientes a las señales a).

### Método de Slalom

Es común que un gran número de datos puedan ajustarse a una única curva suave. La interpolación Spline permite esto. Al usar el método de Slalom y los nodos iniciales se realiza la interpolación obteniéndose como resultado una función suave del carácter, es decir, se reconstruye el carácter. Se calcula el error entre la señal original y la función *Spline* mediante la siguiente ecuación:

$$E_{f,s} = \frac{1}{N} \sum_{i=1}^N (f(x_i) - s(x_i))^2 \quad (1)$$

donde:

$E_{f,s}$  es el error cuadrático medio entre la función *Spline*  $s(x)$  y la señal original  $f(x)$

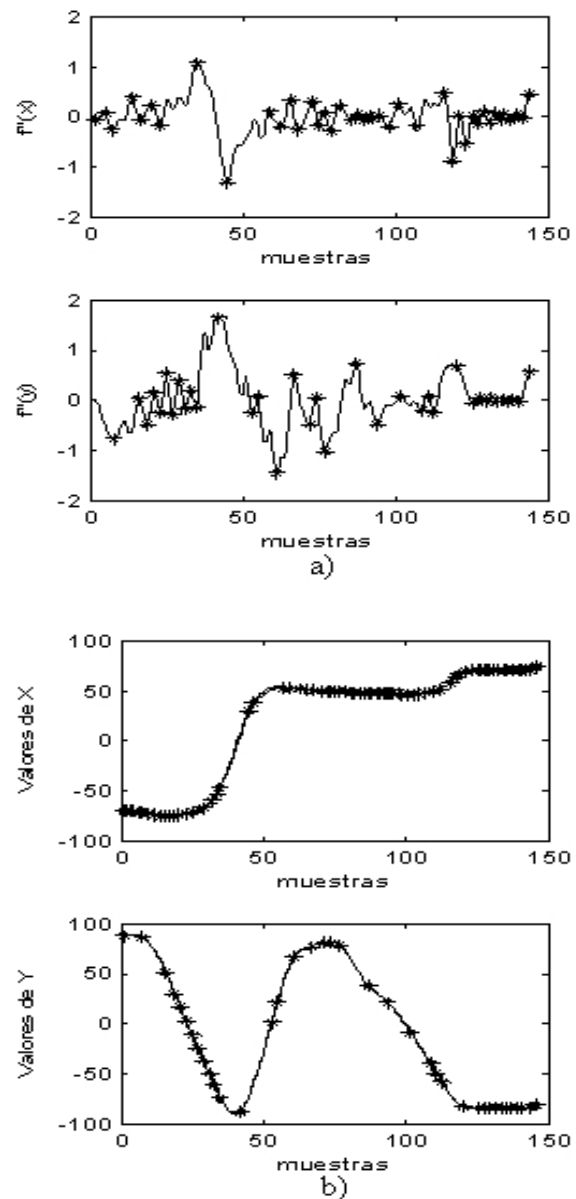
$N$  es el número de datos del carácter manuscrito.

En la figura 6 se muestra la señal original y la señal suave (SLALOM), así como también los nodos iniciales (\*) de la interpolación, en ambos ejes.

En la figura 6 se observa que hay nodos muy cercanos, a los que llamaremos nodos redundantes, se eliminan dichos nodos, tomando el parámetro de que deben haber 6 muestras mínimas entre cada nodo. Por pruebas a los diferentes caracteres se observó que después de 6 muestras, el nodo puede ser considerado como significativo. Al aplicar este criterio obtenemos la figura 7. En esta figura se puede observar que los nodos redundantes se eliminan y se vuelven a interpolar los nodos (slalom). Se calcula el error, y a partir de estos nodos se buscan los nodos óptimos.

### Método de búsqueda de gradiente descendente

Para obtener los nodos óptimos se aplica el método de búsqueda de gradiente descendente. Este método como se sabe permite ajustar los nodos locales antes obtenidos. En la figura 7 se muestran los nodos ya ajustados con el método de búsqueda de gradiente descendente, y la interpolación de los nodos (slalom).



**Figura 6** a) División de la señal y nodos locales del carácter u. b) Señal original y la señal de slalom, con nodos locales del carácter u

### Nodos óptimos

Se obtuvieron los nodos óptimos para cada carácter que para la mayoría de los caracteres en cantidad son menores de los nodos iniciales (nodos en x, nodos en y, valor de los nodos de x y valores de los nodos de y). En trabajos anteriores utilizába-

mos el MDL [16] que nos indicaba cuando detener el proceso y nos daba automáticamente el número de nodos óptimo para cada carácter, los nodos óptimos oscilaban de 18 a 23 nodos, pero como el número de nodos óptimos variaba, no se podían utilizar fácilmente ningún tipo de clasificador, por lo que se decidió fijar el número de nodos óptimos. Buscando que el error siempre disminuyera, se decidió que se obtuvieran siempre 20 nodos óptimos en cada eje de cada carácter que es el 13,7% de la información de la muestra. Al obtener los nodos óptimos del carácter reconstruimos el carácter ‘u’ y ‘o’, como se muestra en la figura 8, donde ‘o’ significa los nodos óptimos en el eje x y ‘\*’ significa los nodos óptimos en el eje y.

### Reconocimiento de caracteres manuscritos

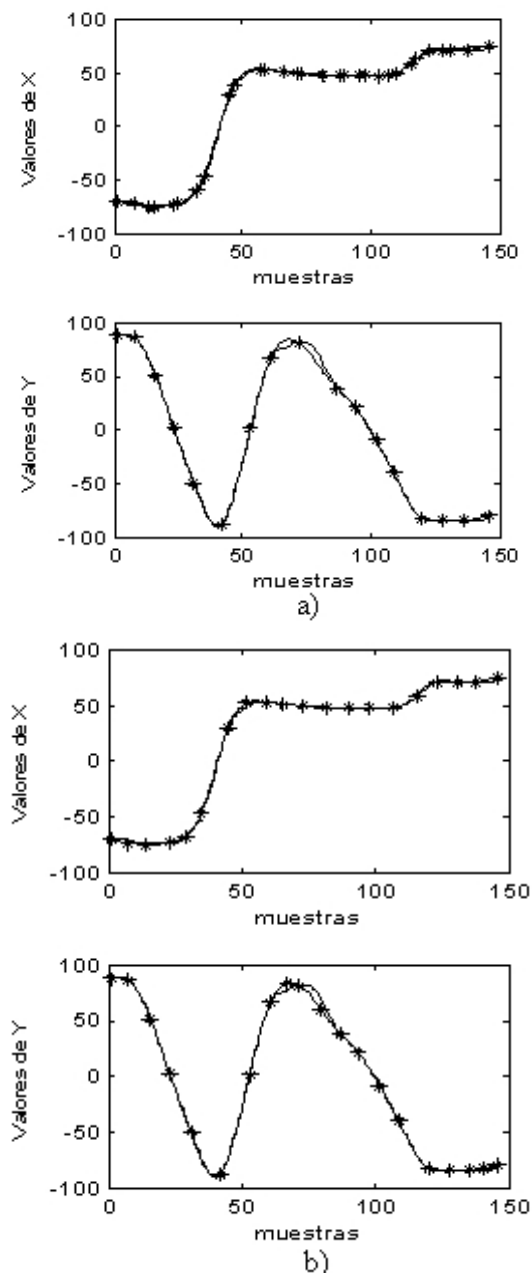
En la etapa de reconocimiento el sistema propuesto de esta investigación se evalúa con tres clasificadores: red neuronal entrenada con regla de propagación hacia atrás (BNN), máquina de vector soporte (MVS) y el modelo de mezclas gaussianas (MMG).

#### Red Neuronal con propagación hacia atrás

En esta investigación se usa la red neuronal con regla de aprendizaje tipo propagación hacia atrás. Esta red, como se sabe, es una de las más confiables para el reconocimiento de caracteres, esto debido a su capacidad de generalización. La regla de propagación del error hacia atrás es una regla de aprendizaje que se puede aplicar en modelos de redes con más de dos capas de células. Una característica importante de este algoritmo es la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las células para conseguir cualquier correspondencia entre la entrada y la salida de la red. La estructura de la BNN utilizada se muestra en la figura 9.

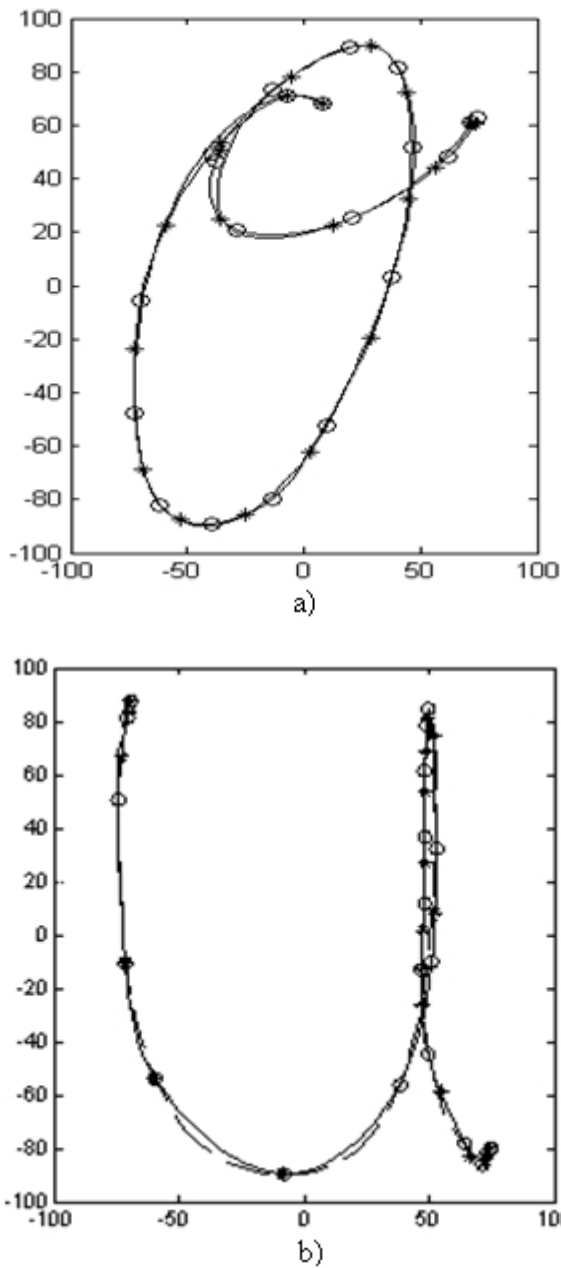
El número de neuronas de entrada se fijó en 40 neuronas (20 por X y 20 por y), mientras que el número en la capa oculta fue de 48 neuronas. Este número fue el mejor que se encontró por prueba y error, ya que como se sabe todavía no existe un método sistemático para conocer el número óptimo

de neuronas en la capa oculta. El número de neuronas en la capa de salida fue de 26, ya que es el número de caracteres del alfabeto inglés. Se usó un factor de convergencia de 0.005 y un error mínimo de 0.01.



**Figura 7** a) Señal original y señal de slalom con sus nodos locales sin nodos redundantes del carácter u. b) Nodos óptimos ajustados por el método de Búsqueda de Gradiente del carácter u

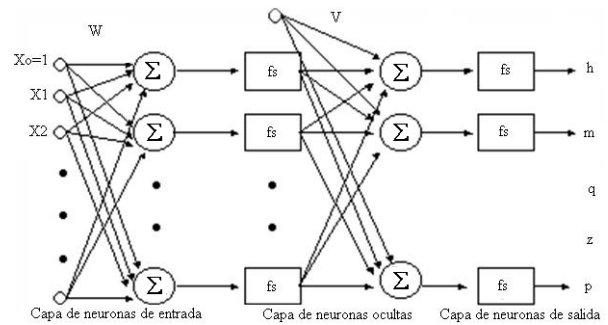




**Figura 8** Trazos originales y trazos contruidos a partir de nodos óptimos. “\*” indica nodos óptimos en eje-x y ‘o’ indica nodos óptimos en eje-y

La teoría de las Máquinas de Vector Soporte (SVM por su nombre en inglés Support Vector Machine) es una técnica de clasificación y ha tomado mucha atención en años recientes [17, 18]. La teoría de las MVS se basa en la idea de minimización de riesgo estructural (SRM) [19].

En muchas aplicaciones, las MVS han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales [18] y han sido introducidas como herramientas poderosas de clasificación. Una MVS primero mapea los puntos de entrada a un espacio de características de una dimension mayor y encuentra un hiperplano que los separe y maximice el margen  $m$  entre las clases en este espacio. Maximizar el margen  $m$  es un problema de programación cuadrática (QP) y puede ser resuelto por su problema dual introduciendo multiplicadores de Lagrange. Sin ningún conocimiento de mapeo, la MVS encuentra el hiperplano óptimo al utilizar el producto punto con funciones en el espacio de características que son llamadas *kernels*. La solución del hiperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son vectores de soporte.



**Figura 9** Estructura de la BNN

*MVS (Máquinas de Vector Soporte)*

*Modelo de Mezclas Gaussianas (MMG)*

Una densidad de mezclas *Gaussianas* es una suma ponderada de  $M$  componentes de densidad, esto se muestra en la figura 10 y se obtiene por la siguiente ecuación:

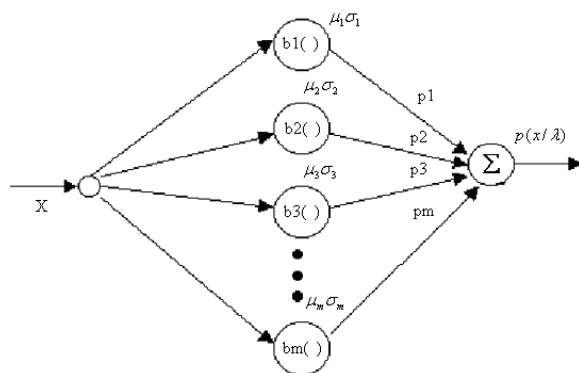
$$p(\bar{x} / \lambda) = \sum_{i=1}^M p_i b_i(\bar{x}) \quad (2)$$

donde  $\bar{x}$  es un vector  $N$ -dimensional,  $b_i(\bar{x}), i=1,2,\dots,M$ , son los componentes de densidad y  $p_i, i=1,2,\dots,M$ , son los pesos de las mezclas. Cada

componente de densidad es una D-variación de la función Gaussiana de la forma:

$$b_i(\bar{x}) = \frac{1}{(2\pi)^{D/2} |\sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\bar{x} - \mu_i)' \sigma_i^{-1} (\bar{x} - \mu_i)\right\} \quad (3)$$

donde ( )' denota el vector transpuesto,  $\mu_i$  es el vector de medias de dimensión y  $\sigma_i$  es la matriz de covarianzas. De esta manera, cada carácter estará representado por un modelo de mezclas Gaussianas que denotaremos  $\lambda = \{p_i, \mu_i, \sigma_i\} \quad i = 1, 2, \dots, M$ .



**Figura 10** Modelo de Mezclas Gaussianas, MMG

El entrenamiento de los MMG's se puede hacer de distintas maneras. La manera tradicional suele realizarse mediante estimaciones de máxima verosimilitud (Maximum Likelihood) a través del algoritmo en dos pasos estimación-maximización (Expectation Maximization), en el que de manera iterativa se refinan los parámetros del MMG para que aumente la probabilidad de generar el vector de características X dado el modelo, o lo que es lo mismo que para las iteraciones r y r+1 se cumpla que:

$$p(X / \lambda(r)) \geq p(X / \lambda(r-1)) \quad (4)$$

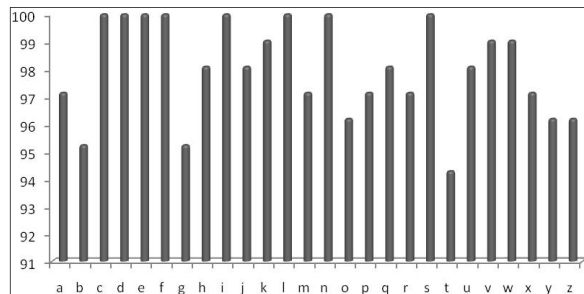
Cada T elementos se deben de actualizar los parámetros del modelo como en [20]. Durante la fase de prueba se estima la probabilidad de que un carácter bajo análisis corresponda al modelo dado que es  $P_r(X / \lambda)$ , por lo que usamos el teorema de Bayes, obteniendo:

$$\hat{R} = \sum_{t=1}^T \log_{10} (p(X_t / \lambda)) \quad (5)$$

donde:  $p(X_t / \lambda)$  es la probabilidad condicional de el carácter X por el modelo  $\lambda$ .

## Resultados

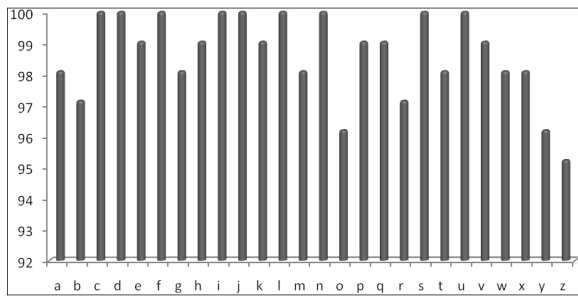
Para el primer caso del clasificador BNN en la fase de entrenamiento se utilizaron 6370 caracteres obteniendo un 99,95% de reconocimiento, mientras que en la fase de prueba en donde se utilizaron 2730 se obtuvo un reconocimiento del 98,01%. En la figura 11 se muestra detalladamente el porcentaje de reconocimiento de cada carácter.



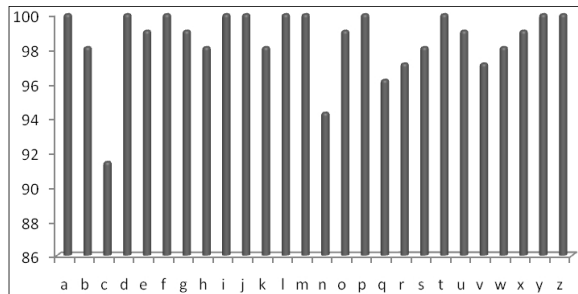
**Figura 11** Reconocimiento del alfabeto inglés mediante BNN

Para el segundo clasificador, las Máquinas de Vector Soporte (MVS) se utilizó un kernel RBF (Radial Basis function), entrenado la MVS con el algoritmo SMO (Sequential Minimal Optimizer). Para la fase de entrenamiento se utilizaron 6370 caracteres obteniéndose un 99,97% de reconocimiento, mientras que en la fase de prueba al utilizarse 2730 se obtuvo un reconocimiento del 98,64%. La figura 12 muestra detalladamente el porcentaje de reconocimiento para cada carácter.

Para el tercer clasificador MMG se utilizó la misma base de datos usada con la BNN y la MVS. En la fase de entrenamiento se utilizaron 6370 caracteres obteniendo un 99,97% de reconocimiento y en la fase de prueba en donde se utilizaron 2730 se obtuvo un reconocimiento del 98,67%. La figura 13 muestra detalladamente el porcentaje de reconocimiento de cada carácter.

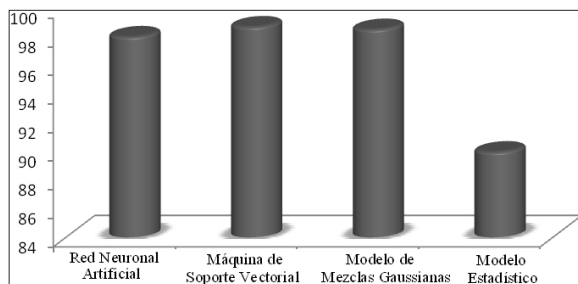


**Figura 12** Reconocimiento del alfabeto inglés usando Máquinas de Vector Soporte (MVS)



**Figura 13** Reconocimiento del alfabeto inglés usando el Modelo de Mezclas *Gaussianas* (MMG)

La tabla 1 compara los resultados obtenidos con cada clasificador para cada carácter, también se realiza la clasificación con el Método Estadístico (ME) utilizado en [16], pero se evalúa ahora con la base de datos actual, es decir, la de 7 escritores, en donde se observa un rendimiento bajo comparado con los otros 3 clasificadores. La figura 14 muestra los resultados globales de cada clasificador, demostrándose que el mejor clasificador para la base de datos utilizada es el MVS.



**Figura 14** Reconocimiento global del alfabeto inglés usando 4 clasificadores

**Tabla 1** Reconocimiento del alfabeto inglés con los 4 clasificadores

Carácter	BNN	MVS	MMG	ME
a	97,14	98,09	100	90,47
b	95,23	97,14	98,09	93,33
c	100	100	91,42	93,33
d	100	100	100	85,71
e	100	99,04	99,04	81,90
f	100	100	100	89,52
g	95,23	98,09	99,04	90,47
h	98,09	99,04	98,09	89,52
i	100	100	100	95,23
j	98,09	100	100	90,47
k	99,04	99,04	99,04	91,42
l	100	100	100	90,47
m	97,14	98,09	100	90,47
n	100	100	94,28	94,28
o	96,19	96,19	100	90,47
p	97,14	99,04	100	80,95
q	98,09	99,04	96,19	83,80
r	97,14	97,14	97,14	90,47
s	100	100	98,09	90,47
t	94,28	98,09	100	93,33
u	98,09	100	99,04	90,47
v	99,04	99,04	97,14	94,28
w	99,04	98,09	98,09	82,85
x	97,14	98,09	99,04	93,33
y	96,19	96,19	100	90,47
z	96,19	95,23	100	91,42

## Conclusiones

En este trabajo se describió con detalle una metodología para el reconocimiento de caracteres manuscritos para letras de tipo cursiva. Las *wavelet* se usaron para mejorar los problemas de suavizado de el trazo. El método de SLALOM se utilizó para obtener los nodos óptimos de cada carácter. Estos nodos óptimos se consideran como las características que describen a cada carácter, que son utilizadas como un vector de entrada en los 3 clasificadores (BNN, MVS, MMG). Los resultados de la evaluación demuestran que el sistema propuesto proporciona una buena taza de reconocimiento, para los tres clasificadores. Estos resultados pueden ser considerados como muy buenos, pensando que los caracteres reconocidos son cursivos y tienen cierto grado de deformación. El porcentaje del reconocimiento de este sistema propuesto es bastante bueno contra los porcentajes reportados en la literatura que oscilan entre el 85% y el 98% para caracteres manuscritos de diferentes idiomas.

## Referencias

- 1 M. Morita, R. Sabourin, F. Bortolozzi, C. Y. Suen. "A Recognition and Verification Strategy for Handwritten Word Recognition". *Proceedings ICDAR'03*. Edinburgh-Scotland. 2003. pp. 482-486.
- 2 J. Mantas. "An Overview of Character Recognition Methodologies". *Pattern Recognition*. Vol. 19. 1986. pp. 425-430.
- 3 F. Nouboud, Plamondon. "On-Line Recognition of Handprinted Characters: Survey and Beta Tests". *Pattern Recognition*. Vol. 23. 1990. pp. 1031-1044.
- 4 P. Rejean, N. Sargur, N. Srihari. "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey". *IEEE Transactions on PAMI*. Vol. 22. 2000. pp. 63-84.
- 5 A. Lemieux, C. Gagne, M. Parizeau. "Genetical Engineering of Handwriting Representations". *Proc. of the International Workshop on Frontiers in Handwriting Recognition (IWFHR)*. Ontario (Canadá). 2002. pp. 145-150.
- 6 H. Mitoma, S. Uchida, H. Sakoe. "Online character recognition based on elastic matching and quadratic discrimination". *Proceedings of 8th International Conference on Document Analysis and Recognition*. Vol. 1. 2005. pp. 36-40.
- 7 L. Koerich. *Large Vocabulary Off-Line Handwritten Word Recognition*. PhD thesis, École de Technologie Supérieure, Montreal-Canada. 2002. pp. 17-34.
- 8 F. Bortolozzi, A. Souza, L. S. Oliveira, M. Morita, *Recent Advances in Handwritten Recognition*, Document analysis. U. Pal, S. K. Parui, B. B. Chaudhuri (editors). Montreal. 2005. pp. 1-31.
- 9 A. M. Namboodiri, A. K. Jain. "On-line Handwritten Script Recognition". *IEEE Trans. PAMI*. Vol. 26. 2004. pp. 124-130.
- 10 A. L. Koerich, R. Sabourin, C.Y. Suen. *Large vocabulary off-line handwriting recognition: A survey*. *Pattern Anal Applic*. Vol. 6. 2003. pp. 97-121.
- 11 L. Cheng-Lin, S. Jaeger, M. Nakagawa. Online Recognition of Chinese Characters: The State-of-the-Art". *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol. 26. 2004. pp.198-203.
- 12 Y. Kato, M. Yasuhara. "Recovery of Drawing Order from single-Stroke Handwriting Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22. 2000. pp. 938-949.
- 13 N. Mezghani, A. Mitiche, M. Cheriet. "On-Line Recognition of Handwritten Arabic Characters using a Kohonen Network". *Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition*. Ontario (Canadá). 2002. pp. 490-495.
- 14 M. Yokobayashi, T. Wakahara. "Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation". *Eighth International Conference on Document Analysis and Recognition ICDAR'05*. Seoul. Vol. 1. 2005. pp.167-171.
- 15 M. Yokobayashi, T. Wakahara. "Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation". *18th International Conference on Pattern Recognition ICPR 2006*. Hong Kong. Vol. 2. 2006. pp. 885-888.
- 16 K. Toscano, G. Sánchez, M. Nakano, H. Pérez, M. Yasuhara. "Cursive Character Recognition System". *CERMA 2006*. Cuernavaca. Vol. II. 2006. pp. 62-67.
- 17 C. Burges B. Schölkopf, A. Smola. *Advances in kernel methods: Support vector machines*. Cambridge, MA: MIT Press. 1999. pp.327-352.
- 18 C. Burges. "A tutorial on support vector machines for pattern recognition". *Data Mining and Knowledge Discovery*. Kluwer Academic Publishers. Boston. Vol. 2. 1998. pp. 121-167.
- 19 V. Ñ. Vapnik. *The nature of statistical learning theory*. New York: Springer-Verlag, 1995. pp. 138-216.
- 20 D. Reynolds, R. C. Rose. "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models". *IEEE Trans. Speech and audio Proc*. Vol. 3. 1995. pp. 72-83.