

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

*“ANÁLISIS Y EVALUACIÓN DE ALGORITMOS PARA EL RECONOCIMIENTO
DE CARACTERES MANUSCRITOS”*

T E S I S

**Que para obtener los títulos de:
Ingeniero Eléctrico Electrónico**

Presenta:

**Israel Delgado Guerra
e Ingeniero en Telecomunicaciones**

Presentan:

**Haydee Bravo Zúñiga
Nancy Rangel Galán**

Asesor: Dr. Rogelio Alcántara Silva

MÉXICO D. F. 2005



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

En primer lugar quiero agradecer su apoyo en todo sentido y forma a mis papás y a mi hermano, porque como siempre se dice y es verdad, yo no hubiera llegado hasta aquí ni sería lo que soy sin ustedes.

Quiero agradecer también la enorme generosidad que la Universidad Nacional Autónoma de México tuvo al permitirme estudiar en ella, desde el CCH Vallejo aprender a aprender, hasta la Facultad de Ingeniería, donde también se aprende a aprender, pero también se aprenden cosas que te permiten ser de utilidad para uno mismo y para los demás. Y la generosidad de la UNAM fue tal, que en ella también encontré amistad, amor, compañerismo y una enorme diversidad de ideas y de formas de ser y pensar en toda la gente que conocí estando en ella.

Esto me lleva a agradecer a mis amigos CCHeros Berenice, Juan Carlos, César y Javier, porque ustedes tienen parte de la culpa de que yo esté aquí, no lo olviden. (Sé que corro el riesgo de no mencionar a alguien, pero espero comprendan que la omisión es a causa de la emoción y de las muchas personas a quienes debo agradecer). Gracias a mis amigos ingenieros Gabriel Chávez, Carmen Salgado, Gabriel Hernández, Carlos Hernández, por todas las anécdotas, la compañía y otras circunstancias que los hace importantes para mí. Gracias siempre a los telerines, Paty, Yoel, Bernardo, David, José Luis, Oscar, José Luis Reyes, Rangel, porque pasamos juntas tantas horas de clase, angustias, prisas, desesperaciones, gracias por enseñarme tantas maneras de reír a pesar de las dificultades, gracias también por su amistad.

Gracias a los que estuvieron y ya no están, gracias a los que llegaron y a los que reaparecieron, gracias por lo que en su momento pasamos juntos.

Gracias a mis maestros, a todos, desde la que me enseñó a leer, a escribir y a sumar (muy importante!), hasta el último profesor de la Facultad, gracias a todos porque de cada uno aprendí algo, de los malos, que hay otras formas de aprender, enseñar y ser, y de los buenos, los conocimientos y las actitudes ante los problemas, no sólo de ingeniería. Gracias muy especialmente al Dr. Rogelio Alcántara Silva por el tiempo y esfuerzo que le dedicó a nuestro trabajo, gracias por toda su ayuda y por los conocimientos compartidos con nosotros.

Gracias a Nancy e Israel por haberme invitado a ser parte de este proyecto, gracias por permitirme compartir con ustedes este largo tiempo y porque finalmente superamos las dificultades y ya estamos aquí!

Ya te mencioné Bere, pero gracias siempre por tu amistad a prueba de tiempo y distancia, gracias por superar juntas tantas cosas y seguir aquí después de tanto tiempo, gracias por tu apoyo también en todo sentido y forma.

Gracias al Ing. Gabriel Chávez por su enorme ayuda en la programación de algunos algoritmos de esta tesis (no me vas a cobrar regalías verdad?), gracias Gabriel por tu amistad y por la infinita paciencia que siempre has tenido conmigo.

Y también ya sabes que, medio en clave pero ya te mencioné, y también sabes bien por qué.

Haydee

Agradecimientos

A mis padres quienes me han dado todo, y sobre todo la libertad de elegir mi camino, la libertad de equivocarme, la libertad de vivir.

A ese ser extraño que adivina el alma o que recuerda acaso de otra vida, que bebe conmigo el cáliz seco del ensueño en la realidad.

A la Universidad Nacional Autónoma de México que me otorgo el honor de estudiar en sus aulas de la escuela nacional preparatoria 6, así como de la facultad de Ingeniería.

Israel Delgado Guerra

Agradecimientos

A mis papás, abuelos y hermanos por su apoyo. En especial a mi Abuelo, aunque hubiese querido que fuera antes, se que te alegrará.

A mis amigos y compañeros de tesis.

A nuestro profesor director y a todos aquellos que han participado en nuestra formación.

Al esfuerzo de todos.

Nancy Rangel Galán

Índice

<i>Índice</i>	<i>i</i>
<i>Introducción</i>	<i>iv</i>
<i>Resumen</i>	<i>vi</i>
<i>Objetivos</i>	<i>vii</i>
1. Generalidades del Reconocimiento de Caracteres Manuscritos (RCM)	1
1.1 Arquitectura de Reconocimiento de Caracteres Manuscritos	1
1.2 Captura de la Escritura Manuscrita	2
1.2.1 Herramientas hardware de captura	2
1.2.2 Herramientas software de captura	6
1.2.3 Compresión y formatos de imagen	8
1.3 Pre-procesamiento de la Escritura Manuscrita	10
1.3.1 Pre-procesamiento a nivel pixel	10
1.3.2 Pre-procesamiento a nivel imagen	12
1.3.3 Segmentación	16
1.4 Extracción de Características de los Caracteres Manuscritos	18
1.4.1 Generalidades	18
1.4.2 Métodos para la extracción de características	19
1.5 Clasificación de Características	22
1.5.1 Selección de características	22
1.5.2 Clasificación de características	22
1.5.3 Bases de datos	23
1.5.4 Comparación y reconocimiento	25

2. Captura	27
2.1 Hardware de captura	27
2.2 Software de captura	29
2.3 Interfaz de captura	33
2.4 Interfaz gráfica del sistema	36
3. Pre-procesamiento y Segmentación	38
3.1 Caracteres	39
3.2 Palabras	42
3.2.1 Segmentación gráfica	43
3.2.2 Segmentación por cortes	44
3.2.3 Segmentación por localización de extremos superiores	47
3.2.4 Selección del algoritmo de segmentación	50
4. Extracción de Características	54
4.1 Método por envolventes	55
4.1.1 Extracción de envolventes	56
4.1.2 Extracción de puntos de corte	59
4.2 Método por trayectorias	60
4.3 Método por vectores	68
4.3.1 Conceptos básicos	69
4.3.2 Métodos de extracción	71
4.4 Bases de datos	75
4.4.1 Base de datos de imágenes	76
4.4.2 Base de datos de vectores (MRTO)	78
4.4.3 Base de datos de vectores (MRV)	79
5. Comparación y reconocimiento	82
5.1 Reconocimiento de caracteres aislados	82
5.1.1 Comparación y reconocimiento por envolventes (RE)	83
5.1.2 Comparación y reconocimiento por trayectorias (RT)	85
5.1.2.1 Reconocimiento por trayectorias optimizado (RTO)	88
5.1.3 Comparación y reconocimiento por vectores (RV)	88

5.2 Reconocimiento de palabras	90
5.2.1 Reconocimiento de palabras utilizando RV	92
6. Aplicación específica del RCM	96
7. Pruebas y Resultados	105
7.1 Pruebas	105
7.2 Resultados	110
7.2.1 Reconocimiento de caracteres	111
7.2.2 Segmentación	112
7.2.3 Reconocimiento de palabras	113
8. Conclusiones y Perspectivas	115
Referencias y Bibliografía	126

Introducción

En la actualidad el desarrollo de sistemas de reconocimiento de caracteres es un problema del ámbito comercial, que se resuelve de forma compleja, pues se requiere del uso de estilos predeterminados para que dicho reconocimiento sea válido, volviendo la aplicación de éstos sistemas muy poco natural.

Con lo anterior se deduce que son varios los problemas de reconocimiento, basta con escribir nuestro nombre dos veces para comprobar que hay un sinnúmero de parámetros que varían y que es preciso considerar para resolver el problema. Es así como se vuelve necesario involucrar diversos algoritmos de reconocimiento, sin embargo esto mismo ocasiona un incremento considerable en el trabajo y en la experiencia requerida para darle solución.

Este tema no es reciente, ya que se han realizado trabajos desde los años setenta, el trabajo por hacer es sumamente extenso, y alrededor del mundo hay muchas personas dedicadas a investigar un pequeño aspecto sobre el reconocimiento de caracteres, ya sea segmentación, reconocimiento de patrones, contexto, semántica, entre muchas otras áreas, que pueden a su vez enfocarse hacia alfabetos de distintos idiomas. En el idioma español se han desarrollado pocos trabajos de este tipo, ya que la gran mayoría de las soluciones disponibles se encuentra en inglés.

La gran extensión de este tema, que comprende la captura y el análisis de datos, la programación de interfaces, la validación y comprobación de resultados, considerando que hay un sinnúmero de métodos para realizar cada parte del proceso, requiere sin duda del trabajo en equipo de personas de diversas disciplinas especializadas en las áreas requeridas.

En este trabajo se presenta una serie de algoritmos originales y sencillos para el reconocimiento de caracteres manuscritos cursivos, en particular trece letras y diez palabras cortas de nuestro alfabeto. Los algoritmos se basan en estrategias geométricas en el dominio de la imagen. Los resultados de la evaluación de los algoritmos propuestos, sobre dos bases de datos, mostraron tasas de eficiencia superiores al 90% en el reconocimiento de caracteres aislados, y de 80% en el reconocimiento de palabras. Una parte muy importante en el análisis del desempeño cuantitativo de los algoritmos, son los resultados de la segmentación de palabras, la cual obtuvo un porcentaje de eficiencia superior al 97%.

En el análisis cualitativo de los algoritmos, además de sus altas tasas de eficiencia, éstos presentan un reducido tiempo de ejecución, así como modularidad y sencillez en los métodos de programación empleados. Por otro lado, la interfaz desarrollada permite una presentación clara del funcionamiento de los diferentes algoritmos que conforman al sistema de RCM, además de que su uso es sencillo y comprensible para los usuarios finales. El proceso completo en sí mismo es sistematizado, eficiente, rápido y de bajo costo.

En el Capítulo 1 se abordarán los conceptos teóricos necesarios para ilustrar los elementos requeridos en un sistema de reconocimiento de escritura. A partir del Capítulo 2, el cual habla sobre la metodología empleada para capturar la escritura cursiva que analizaremos, describiremos nuestra propuesta del sistema de reconocimiento de caracteres manuscritos que cumpla con los objetivos que nos hemos planteado. En el Capítulo 3 se describen los algoritmos de pre-procesamiento y segmentación aplicados sobre los caracteres y palabras de nuestro estudio. Por otro lado, el Capítulo 4 trata sobre la extracción de las características de la escritura que serán útiles para su posterior reconocimiento.

En el Capítulo 5, se describen los algoritmos de reconocimiento, tanto de palabras como de caracteres, partiendo de las características obtenidas y de su comparación con caracteres prototipo. En el Capítulo 6 tenemos la aplicación específica de todos los algoritmos que intervienen en el reconocimiento de caracteres y palabras, a través de un sistema de interfaces gráficas. En el Capítulo 7 tenemos las pruebas, validación y resultados de los algoritmos que intervienen en todos los procesos, y finalmente, en el Capítulo 8 tenemos las conclusiones obtenidas y las perspectivas de trabajos y aplicaciones futuras de esta área del procesamiento de imágenes.

Con la finalidad de situar los algoritmos de reconocimiento de caracteres y palabras propuestos dentro del contexto de la arquitectura general de un sistema de reconocimiento de caracteres manuscritos, se hace una descripción de las funciones y de las tareas que realiza cada uno de los elementos que conforman dicha arquitectura. El diseño e implementación de los algoritmos, así como los resultados de la evaluación de su desempeño, se presentan de manera detallada.

Resumen

En este trabajo se presenta un conjunto de algoritmos originales y sencillos para el Reconocimiento de Escritura Manuscrita Cursiva (REMC), en particular trece letras y diez palabras cortas de nuestro alfabeto. Dichos algoritmos se basan en estrategias geométricas en el dominio de la imagen. La captura de la escritura se realiza en tiempo real, mientras que el procesamiento y reconocimiento es fuera de línea.

Los resultados de la evaluación de los algoritmos propuestos, sobre dos bases de datos, mostraron tasas de eficiencia superiores al 90% en el reconocimiento de caracteres aislados, y de 80% en el reconocimiento de palabras. Igualmente, se evalúan los resultados de la segmentación de palabras, la cual obtuvo un porcentaje de eficiencia superior al 97%. Dichos algoritmos presentan un reducido tiempo de ejecución, así como modularidad y sencillez de los métodos de programación empleados. Por otro lado, a través de una interfaz gráfica se analizan y simulan los diferentes procesos que conforman al sistema de REMC propuesto.

Con la finalidad de situar los algoritmos de reconocimiento de caracteres y palabras propuestos dentro del contexto de la arquitectura general de un sistema de reconocimiento de caracteres manuscritos, se hace una descripción de las tareas que realiza cada uno de los elementos que conforman dicha arquitectura. El diseño e implementación de los algoritmos, así como los resultados de la evaluación de su desempeño, se presentan de manera detallada.

Objetivos

Objetivos generales

- Estudiar la arquitectura de procesamiento para el reconocimiento de escritura manuscrita.
- Explorar y evaluar distintas técnicas de procesamiento digital de señales, para la obtención de las características útiles en la clasificación y reconocimiento de escritura manuscrita.
- Analizar y simular algoritmos en la secuencia de procesamiento para el reconocimiento de caracteres manuscritos tales como letras y palabras cortas.

Objetivos particulares

- Reconocer las palabras manuscritas cursivas de un diccionario específico: los nombres de los números cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, escritos con minúsculas.
- Reconocer los caracteres manuscritos cursivos que forman los nombres de dichos números, los cuales son las letras minúsculas a, e, i, o, u, c d, h, n, r, s, t, v.
- Implementar algoritmos para la segmentación de las palabras del diccionario problema.
- Desarrollar un ambiente de simulación (interfaz gráfica) de los algoritmos más representativos del proceso de reconocimiento, incluyendo la captura de la escritura, a fin de facilitar la evaluación de su desempeño.

Capítulo 1

Generalidades del Reconocimiento de Caracteres Manuscritos (RCM)

1.1 Arquitectura de Reconocimiento de Caracteres Manuscritos

La arquitectura de un sistema de reconocimiento de escritura manuscrita comprende, desde la captura de la imagen inicial de la escritura, hasta el reconocimiento de la misma. El sistema se puede dividir de manera general en tres secciones: captura y pre-procesamiento, extracción de características y finalmente la sección de comparación y reconocimiento; tal como se encuentran divididos la mayoría de los sistemas de reconocimiento de escritura manuscrita, en la literatura sobre el tema. El diagrama a bloques de la arquitectura general de un sistema para el reconocimiento de la escritura manuscrita se muestra en la Figura 1.1 [16], [20].

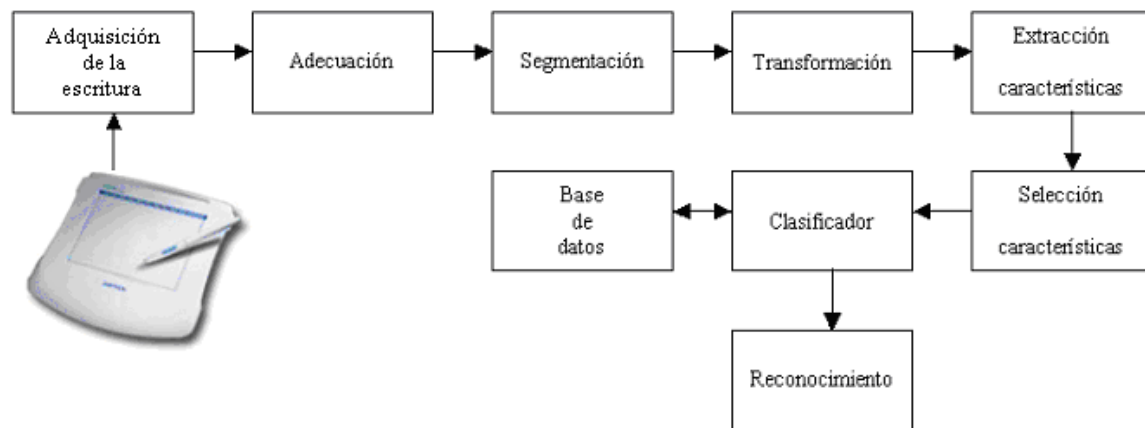


Figura 1.1. Diagrama a bloques del sistema

El proceso de captura inicia con la adquisición de los datos mediante algún dispositivo de captura de escritura, principalmente con una tableta digitalizadora y su interfaz de captura, o bien mediante sistemas ópticos, como los escáneres, almacenando la información en algún formato de imagen o en otros métodos de almacenamiento.

El siguiente paso del proceso es la adecuación de la señal, la que puede constar de varias etapas. De esta manera, la señal de información podría ser amplificada y filtrada para corregir

defectos tales como la atenuación o el ruido. También podría existir otra etapa de adecuación a nivel pixel, en donde se pueden incluir tratamientos tales como el adelgazamiento (thinning), suavizado (smoothing), esqueletización (skeletonizing), vectorización (vectorization), entre otros [1]. La aplicación de estos procesos se considera parte del pre-procesamiento de la escritura, y dependiendo de la calidad de la información capturada pueden considerarse todos o solo algunos de estos procesos.

La segmentación de la escritura se realiza antes de la aplicación de los algoritmos de extracción de características y puede ser de dos tipos: externa e interna. En la segmentación externa se realiza la separación de la imagen capturada y pre-procesada, en unidades de reconocimiento, tales como caracteres. Si se requiere una nueva segmentación ahora de la unidad de reconocimiento (carácter) se utilizará una segmentación interna, que consiste en la separación de ciertos trazos de la unidad de reconocimiento para la extracción de ciertas características [16].

En el bloque de transformación se aplica algún proceso al dibujo de la escritura ya adecuado y segmentado, de tal manera que la información obtenida contenga características diferentes a las originales, por ejemplo, como resultado de este proceso puede obtenerse una función cuyo dominio es el del tiempo o la frecuencia [1]. Lo anterior con la finalidad de poder analizar determinadas características de la escritura según el tipo de reconocimiento a utilizar, las que de otra manera no podrían ser estudiadas ni extraídas para su utilización en el dominio de la imagen.

Una vez que los caracteres se encuentran aislados y en el dominio correspondiente, el proceso de reconocimiento continúa con la extracción de características previamente definidas que identifican al carácter o a la palabra y que la distinguen de todas las demás. Estas características se seleccionan y clasifican para ser guardadas en una base de datos, en donde se encontrarán todas las características extraídas de los prototipos, para compararlas posteriormente con las obtenidas para los caracteres de prueba. La comparación consiste en la revisión de las características de las bases de datos a fin de encontrar las características más parecidas a las obtenidas para la prueba [20]. Una vez hallado ese carácter, es asignado como carácter reconocido. Para concluir con el reconocimiento se representa al carácter o a la palabra original mediante alguna fuente establecida previamente.

1.2 Captura de la Escritura Manuscrita

1.2.1 Herramientas Hardware de Captura

En la actualidad existen diversas formas de obtener información digitalizada mediante hardware, como son los escáneres (lectores ópticos), tabletas electrónicas, cámaras tanto de video como fotográficas, etc. Los medios que más se utilizan para la captura de escritura son los escáneres y las tabletas electrónicas (o digitalizadoras), los cuales serán descritos a continuación [16].

El escáner es un dispositivo que puede capturar texto o ilustraciones impresas en papel y trasladar la información a un formato que la computadora pueda usar. Un escáner trabaja digitalizando una imagen dividiéndola en un arreglo de cuadros y representando cada cuadro con un cero o un uno dependiendo de cómo el cuadro esté lleno. Para colores y escalas de gris, se aplica el mismo principio, pero cada cuadro es representado por hasta 24 bits. El resultado es una matriz de bits, llamada mapa de bits, que puede ser grabada en un archivo, desplegada en pantalla, y manipulada por diversos programas [1].

Un factor determinante en la calidad del escáner es la resolución o densidad del mapa de bits. Normalmente los escáneres soportan resoluciones desde 72 a 600 dpi (dots per inch: puntos por pulgada). Algunos fabricantes mencionan dos tipos de resolución: óptica e interpolada. La resolución óptica se refiere a la cantidad real de píxeles registrados, mientras que la interpolada incrementa por medio de cálculos la cantidad de puntos en la imagen. Es más confiable evaluar un escáner a partir de su resolución óptica, que se refiere a la capacidad del dispositivo, mientras que la resolución interpolada se relaciona con el programa que el escáner utiliza [1], [2].

Los escáneres difieren unos de otros en las siguientes características.

Tecnología de escaneo

- CCD (Charge-Coupled Device): muchos escáneres usan arreglos de dispositivos CCD, los cuales consisten en filas empaquetadas de receptores de luz que pueden detectar variaciones en su intensidad y frecuencia. La calidad del arreglo de CCDs es probablemente el factor que más afecta la calidad del escáner.
- PMT (tubo fotomultiplicador): los más potentes escáneres de tambor de uso industrial utilizan una tecnología diferente, la cual se basa en un PMT, sin embargo este tipo de escáner es mucho más caro que los escáneres basados en tecnología CCD.

Tamaño y forma

- De mano: algunos escáneres son pequeños dispositivos de mano que pueden moverse sobre el papel. Estos escáneres de mano también se llaman escáneres de media página, porque solo pueden escanear de 2 a 5 pulgadas a la vez. Los escáneres de mano son adecuados para fotos, pero son difíciles de usar si se necesita escanear una página entera o una gráfica.
- De cama plana: tiene una base horizontal, similar a la de una copiadora; son adecuados para capturar documentos en hojas de papel. Usualmente se utilizan junto con un alimentador automático para procesar grandes volúmenes de documentos. No son los más adecuados para capturar imágenes. El tamaño del área de captura permite obtener imágenes de tamaño carta como máximo.

Por otro lado, las tabletas gráficas, también conocidas como tabletas digitalizadoras, son digitalizadores electromagnéticos, esto es, mediante campos electromagnéticos siguen la traza de la pluma, a través de la presión captada en los ejes X y Y. Estos dispositivos pueden trabajar en línea o fuera de línea, esto significa que la información puede ser procesada a medida que se va generando, o se puede guardar hasta que se acaba de escribir para después procesarla.

Las tabletas digitalizadoras han existido por más de tres décadas. Estas tabletas pueden ser usadas para una gran variedad de tareas interactivas. Seis tipos de tareas han sido clasificados para digitalizadores y otros dispositivos de apuntamiento: selección, posición, orientación, ruta, cuantificación, y entrada de texto. Aquí nos interesa el uso de los digitalizadores para la captura en tiempo real de escritura manuscrita de diferentes tipos.

Un cierto número de tecnologías está disponible para las tabletas digitalizadoras. Hasta ahora, los dos tipos principales son las tabletas electromagnéticas / electrostáticas y de presión sensitiva. Las tabletas electromagnéticas / electrostáticas tienen mallas de conductores en los ejes x y y, espaciados

de 0.1 a 0.5 cm en adelante en la tableta y una espiral de alambre en la punta de la pluma incorporada a la tableta. La posición de la punta de la pluma se determina cuando la malla es excitada con un pulso electromagnético, o por el cierre de una malla de detección de voltaje inducido por la pluma, y con esto se realiza una interpolación para obtener la posición de la pluma en la tableta [16].

Las tabletas sensibles a la presión tienen capas de materiales conductores y resistivos con un espaciado mecánico entre las capas. Un potencial eléctrico se aplica a través de una de las capas resistivas, ya sea en la dirección x o y , para colocar un gradiente de voltaje que corresponda a la posición. La presión de la punta de la pluma en un punto ocasiona que la capa conductora seleccione el voltaje (y por lo tanto la posición) proveniente de la capa resistiva. La tecnología sensible a la presión tiene la ventaja de que no requiere el uso de una pluma especial [16].

La precisión en la medición de las tabletas digitalizadoras se caracteriza por la resolución, exactitud, y tasa de muestreo. Para capturar los detalles de la escritura normal, los requerimientos de la tarjeta son estrictos: una resolución de al menos 200 dpi y una tasa de muestreo de al menos 100 muestras por segundo.

Una tableta es un dispositivo alternativo de entrada, que puede ser usado en lugar del ratón, o conjuntamente con él. La tableta consiste de dos partes principalmente, las cuales ya hemos mencionado: una superficie plana (tableta), y una pluma, que se configura para trabajar con la tableta. Algunos sistemas incorporan un ratón inalámbrico que funciona con la superficie de la tableta; muchos usuarios eligen utilizar una tableta, ya que es más ergonómica, y reduce la probabilidad de desarrollar alguna malformación en las manos por la repetición de la presión. A continuación comentaremos algunas de las características y especificaciones más comunes de las tabletas gráficas [3], [4].

Tamaño

El tamaño es uno de los principales factores que se considera para elegir una tableta. Para los usuarios profesionales y los aficionados caseros, los tamaños más comunes son los de 4x5 pulgadas y de 6x8 pulgadas. Los diseñadores gráficos y los ilustradores técnicos pueden desear un área superficial más grande, pero el precio se extiende a medida que el tamaño aumenta. Un factor importante a saber sobre el tamaño de la tableta es que las dimensiones dadas se refieren casi siempre al área superficial de la entrada de la tableta. El tamaño real de la tableta puede ser tanto como 4 a 5 pulgadas más grande que el área activa: por ejemplo, una tableta de 6 pulgadas de área activa tiene un tamaño total de 10 a 12 pulgadas.

Interfaz

La interfaz es la manera como la tableta se conecta a la computadora; la mayoría de las tarjetas actualmente se conectan mediante interfaz USB, y esta es claramente la mejor manera de conexión. En computadoras viejas que no cuentan con este tipo de puertos, se elige una tableta con una interfaz en puerto serie. Cabe mencionar que una interfaz USB consigue su energía de la conexión a la computadora, mientras que el puerto serial requiere una conexión de energía separada.

Plumas y accesorios

La pluma conectada a la tableta debe sentirse cómoda y natural en la mano; existen plumas

que utilizan pilas, lo que hace no solo que se requiera de pilas de reemplazo, sino que también la hará más pesada. La pluma también puede ser libre o conectada. Muchas plumas también tienen un interruptor o botones en el cuerpo de la pluma, y algunas plumas tienen un extremo que borra, esto es una característica excelente porque los botones se pueden programar para las funciones específicas tales como un click derecho o un doble click, y la extremidad que borra puede realizar una función de cancelación. Algunos fabricantes ofrecen plumas adicionales que se pueden programar independientemente.

Presión sensitiva

El nivel de presión sensitiva se refiere a la respuesta a la presión que ofrece la tableta. La mayoría de las tabletas tienen 256, 512, ó 1024 niveles de presión. La presión sensitiva puede controlar el grueso de la línea, la transparencia, y / o el color; cuanto más alta sea la respuesta a la presión sensitiva, la tableta responderá de manera más natural, y el control será mejor.

Software del controlador

El controlador maneja el funcionamiento de la tableta; algunas de las tabletas más caras ofrecen capacidades muy avanzadas debido al software del controlador.

Software asociado

El software asociado puede agregar valor a la tableta. La mayoría de las tabletas vienen con un programa de pintura, y algunas incluirán utilidades adicionales para aprovechar la tableta. Los programas Adobe Photoshop LE, Corel, y Corel Art Dabbler son los títulos asociados más comunes. Si se encuentra una tableta que no ofrezca ningún software asociado, se debe considerar que para explotar la tableta al máximo se necesitará realizar compras posteriores.

Precio

Las tabletas gráficas pueden ser muy costosas, inclusive en cientos de dólares, sin embargo, hay muchos fabricantes que ofrecen las tabletas como valor agregado a usuarios caseros. Estas tabletas se tasan generalmente alrededor de \$200 pesos o menos, aunque carecen de algunas de las características profesionales de las tabletas más costosas. La tableta a elegir depende del tipo de trabajo a realizar, ya sea para diseño de páginas Web, diseño gráfico, fotografía, etc. Por último, los fabricantes más reconocidos del ramo de las tabletas digitalizadoras en nuestro país son: Genius, Acecat, Dynalink, Abera Systems, Aiptek, CalComp Technology y MediaForte.

Por otro lado, respecto a la información que se obtiene de ambos dispositivos, tanto los escáneres como las tabletas digitalizadoras entregan imágenes en formato raster. Este tipo de archivo de imagen es la base de cualquier formato de archivo, de ahí que cada programa lo transforme a otro tipo de archivo para poder utilizarlo, lo cual será explicado más adelante [3], [4].

Un aspecto muy importante es la manera como recibimos la información de los dispositivos de captura; por lo general es por medio de puertos de comunicación. La comunicación entre computadoras se ha incrementado notoriamente, actualmente el correo electrónico está desplazando tanto al correo normal como al teléfono, así como también cada día se usan más y más bases de datos

electrónicas. Estos tipos de comunicación vía computadora se realizan por medio de puertos, ya sea paralelos, ethernet, o seriales; de estos últimos hablaremos a continuación.

En la llamada conexión dura dos dispositivos de comunicación se interconectan para intercambiar señales eléctricas, y mientras que uno recibe la información (receptor), el otro la transmite (transmisor). Esta comunicación puede ser directa o indirecta: en el primer caso se conectan los sistemas mediante cables que van de computadora a computadora, y en la segunda interviene un medio, el cual puede ser una línea telefónica, fibra óptica, e incluso el aire, para sistemas de radiocomunicaciones.

Los equipos de comunicaciones conectados en serie se pueden dividir en simplex, half-duplex y full-duplex. En las comunicaciones de una vía (simplex) se transmite en un solo sentido, esto es, que no se intercambian los sistemas de transmisor a receptor, y de receptor a transmisor. La comunicación half-duplex significa que los datos pueden ser enviados en ambas direcciones entre dos sistemas, pero en una sola dirección al mismo tiempo. En una transmisión full-duplex cada sistema puede enviar y recibir datos al mismo tiempo.

Los dispositivos se conectan principalmente por puertos serie convencionales, aunque también pueden ser conectados por puertos seriales USB. Para una conexión serial existe un solo tipo de señal, que es la eléctrica, y esta tiene dos estados permitidos: un espacio, que corresponde a un cero binario, cuando existe un voltaje positivo, y una marca, que corresponde a un uno binario, cuando existe un voltaje negativo.

El puerto serie de la computadora es un dispositivo asíncrono; en este tipo de sistemas, un bit identifica el comienzo de la transmisión y 1 o 2 bits identifican su final, y no es necesario ningún bit de sincronía. Los bits de datos son enviados al receptor después del bit de comienzo. El bit menos significativo es transmitido primero. Un carácter de datos suele consistir de 7 u 8 bits. Dependiendo de la configuración de la transmisión un bit de paridad es enviado después de cada bit de datos. Este se utiliza para corregir errores en los caracteres de datos. Finalmente son enviados 1 o 2 bits de parada.

1.2.2 Herramientas Software de Captura

Respecto al software utilizado en un sistema de reconocimiento de escritura, este es muy vasto, ya que incluye desde el software para controlar el dispositivo de captura de datos, mediante el puerto de comunicación utilizado para tal función, así como el software de la interfaz gráfica, la cual es la manera como se presentará el sistema al usuario. En general se utiliza un ambiente tipo ventana, desarrollado ya sea por medio de programación en C, Matlab, Java, etc. La razón de incluir una interfaz es que sea sencillo para el operador (usuario final) hacer uso del sistema.

La interfaz gráfica proporciona el realce de una función y adiciona una imagen atractiva a cualquier aplicación informática. Un ejemplo de una GUI (Graphical User Interface) comúnmente usado sería decir que Microsoft Windows 95 es un GUI complejo para MS-DOS. Las interfaces que utilizan gráficos son hoy el paradigma dominante para intercambiar información entre las computadoras y las personas [30].

Anteriormente, desarrollar un GUI era un proceso muy largo que requería de un equipo de desarrollo de software, pero a menudo el producto final no resultaba el mejor para su GUI. Hoy las

cosas no podrían ser más diferentes: centrado en la codificación y la funcionalidad de las características finales del producto, el diseño del GUI es una parte mucho más integral del proceso de diseño. La interfaz no solamente debe ser estética, sino que también debe satisfacer las necesidades de los usuarios. La interfaz es la parte del sistema informático que permite que el usuario interaccione con la computadora. Sería justo precisar que el propósito de una interfaz es hacer al sistema informático utilizable por el usuario. En otras palabras, la calidad de una interfaz depende de su utilidad.

La utilidad implica el adaptar el sistema informático a las personas que lo utilizan. Esto implica que los diseñadores del software puedan cambiar su propia perspectiva, viendo el problema desde la perspectiva del usuario final y sus propósitos, ya que si no cumple con los propósitos del usuario, entonces el sistema tiene un problema de utilidad. La Figura 1.2 es un organigrama a detalle del proceso ideal del desarrollo del GUI [30].

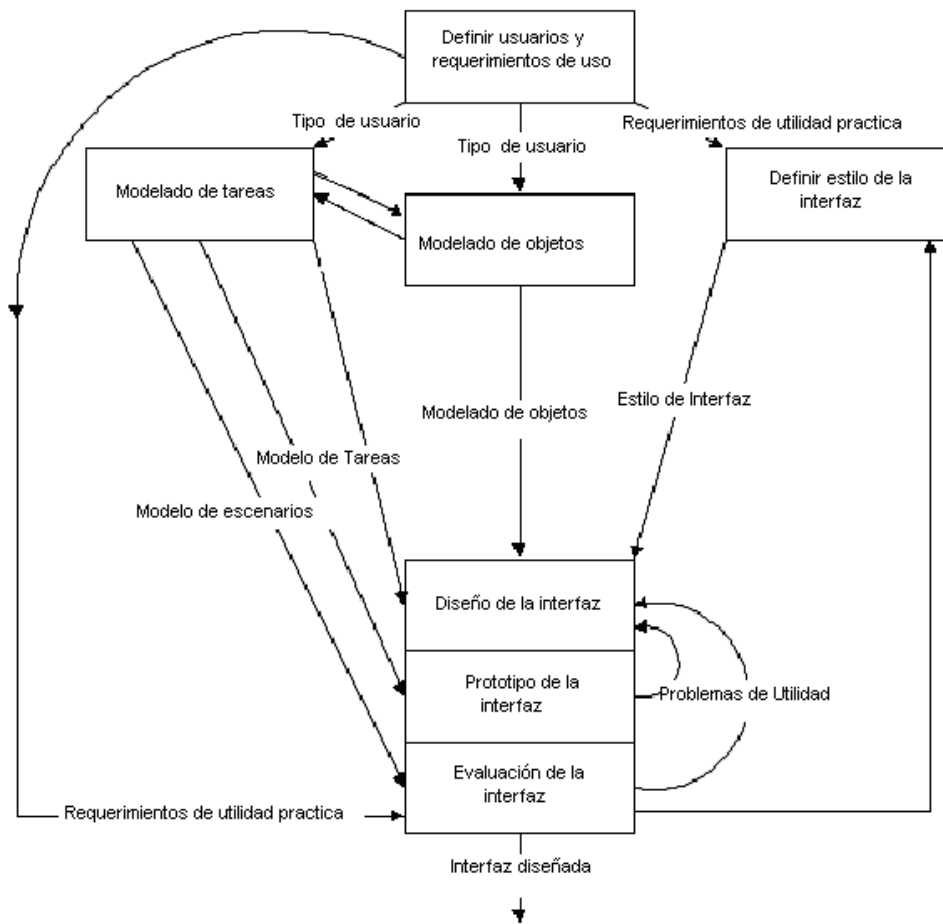


Figura 1.2. Esquema de desarrollo de una GUI

Los aspectos más importantes de este proceso son el diseño, prototipo y evaluación de los procesos del GUI.

1.2.3 Compresión y Formatos de Imagen

Las imágenes se clasifican de acuerdo a la forma en que fueron generadas, y pueden ser de dos tipos: vectoriales y de mapa de bits.

Las imágenes vectoriales definen a una imagen como una lista de objetos gráficos, llamados primitivos. Algunos ejemplos de primitivos son las líneas, rectángulos, circunferencias, etc. Una característica fundamental de este tipo de imágenes es que no presentan problemas de escalamiento o transformación, ya que esto se resuelve realizando una transformada numérica [5], [6].

Las imágenes de mapa de bits (bitmapped) se obtienen a través de escáneres, tabletas electrónicas y otros medios ópticos. Su dimensión y estructura está definida por el número de píxeles y de colores que se puede representar: a mayor número de píxeles mejor calidad de la imagen. Como no tiene un tamaño definido físicamente, existen problemas para realizar su escalamiento si la escala no es un número entero de su dimensión; además de esta desventaja, las imágenes bitmapped ocupan gran cantidad de memoria. Es por ello que al momento de generar un archivo se utilizan códigos de compresión para reducir su tamaño [7], [8].

La estructura de los archivos de imágenes consiste de un encabezado y de la imagen comprimida. La información del encabezado varía dependiendo del tipo de archivo, que básicamente consiste de:

- Tamaño de la imagen
- Colores
- Tipo de compresión
- Programa con el cual fue creado [9].

Existen además archivos que son flexibles, esto es que se les puede agregar información práctica en el encabezado.

A continuación mencionaremos algunos de los tipos de archivos más usados para el reconocimiento de caracteres manuscritos.

- Tipo BMP: este tipo de archivo es bitmapped y tiene la particularidad de que nunca se guarda comprimido, lo que nos da la oportunidad de aplicarle distintos tipos de compresores [7], [8].
- Tipo PCX: este es un archivo generado por Paint Brush, su compresión es run length (explicada más adelante), de 2, 4, 8, 16 y 256 colores, y es un tipo de archivo bitmapped [8].
- Tipo WPG: puede ser de tipo bitmapped y/o vectorial, entre sus primitivos se encuentran líneas, curvas, elipses, circunferencias, y se comprime con un run length [7].
- Tipo TIFF: es un archivo flexible de etiquetas, es independiente de la máquina en que fue generado, su compresión es pack bit, la cual es muy similar a run length. Este tipo de archivos tiene aplicaciones de circuitería electrónica, y en un principio fue la base para los escáneres [10], [11].

- Tipo TGA: no tiene límites de dimensiones, es un archivo flexible, se puede generar comprimido o no. Este es el tipo de archivo más difundido en 24 bits, y lo utilizan la mayoría de los escáneres a color [7], [8].

La meta de la compresión de la imagen es alcanzar un alto nivel en la reducción del número de bytes para representar la imagen original con una degradación aceptable, para la reducción en tiempo de la transmisión de la imagen, el espacio de almacenaje, y el tiempo de respuesta del sistema. La aplicación de la compresión de imágenes se da en las videoconferencias, multimedia, videoteléfonos, video interactivo, cámaras digitales, impresoras, distribución de TV por cable, mail multimedia, operaciones remotas (telemedicina, teleradiología), etc. [12].

Un par de conceptos importantes en este tema son la entropía y la redundancia. La entropía, definida como:

$$H = f(p_1, p_2, \dots, p_n),$$

asume que cada mensaje es independiente del resto de los mensajes, es decir, cada pixel en una imagen es estadísticamente independiente del resto de los pixeles. Cuanto más alta es la entropía, más es la información que una imagen contiene; cuanto mayor es el cociente bits/pixel, menor es el cociente de la compresión [13], [5].

Por otro lado, la redundancia se define como:

$$\text{redundancia} = \log(2M) - H$$

donde el número M es el nivel de cuantización. No es posible codificar una fuente discreta en menos segmentos que H , el cual es el límite superior para la compresión sin que la información se pierda [2].

Los tipos de compresión sin pérdida de información de la imagen son los siguientes.

- Codificación de Entropía: codificación de Huffman, codificación de Shannon-Fano, codificación modificada de Huffman.
- Codificación Estimativa: Modulación de Código de Pulso Diferenciado (DPCM), predicción lineal y predicción polinómica.
- Codificación de Función-Longitud: codificación del pixel, codificación del bitplane, codificación adaptada de la función-longitud [5], [12], [14].

Los criterios de evaluación de la compresión de la imagen son los siguientes:

1. Cociente previsto de la compresión.
2. El algoritmo funciona igualmente bien en imágenes provenientes de diversas fuentes de proyección de la imagen.
3. Susceptibilidad del método a los errores de almacenaje y transmisión.

4. Cómo se implementa en hardware.
5. Reconstrucción de imágenes a una velocidad aceptable.
6. Medida de la fidelidad de la imagen: si se pierde información importante.

La calidad de la imagen del objeto se mide mediante:

- Error cuadrático medio (MSE).
- Promedio normalizado del error cuadrático medio (NMSE).
- Porcentaje del promedio del error cuadrático medio (PMSE).
- Relación señal/ruido (SNR) [5], [13], [14].

El código run length es un caso especial de la predicción polinómica, es decir, con una predicción de orden cero, y con una tolerancia de cero. Cualquier secuencia de palabras codificadas idénticas se considera un corrimiento, si el valor de las palabras codificadas se conoce, entonces solamente es necesario especificar un punto y una longitud de partida para definir totalmente el corrimiento, además es necesario determinar el número de bits que serán asignados para cada uno.

Asimismo, debe considerarse el valor para elegir el código: código de longitud variable (entropía) o campo fijo de longitud, y la longitud: si la estadística del funcionamiento se conoce o puede ser estimada, un código de entropía puede ser utilizado [14].

En la codificación adaptativa de la función de corrimiento (run length), para adaptarse a la estadística local de una imagen, se necesita un algoritmo para ajustar continuamente el número de bits asignados, para que haya información de la codificación de la longitud, o para utilizar la línea anterior para estimar las longitudes del corrimiento en la línea actual.

La codificación de planos de bits (bitplane) desglosa una imagen de n-bits en imágenes binarias separadas en n planos de bits, todos los planos de bits abarcan unos y ceros, y en la mayoría de los casos la codificación del valor del funcionamiento no es necesaria, cada plano de bits varía en su compresibilidad [15].

1.3 Pre-procesamiento de la Escritura Manuscrita

1.3.1 Pre-procesamiento a nivel pixel

El procesamiento previo al reconocimiento consiste de dos partes principales: normalización y representación. La primera de ellas tiende a remover en las imágenes aquellas variaciones que no afectan la identidad de la palabra, y la segunda expresa la información saliente en la imagen de un modo conciso, conveniente para el procesamiento en un sistema de reconocimiento de patrones.

El pre-procesamiento de caracteres manuscritos se realiza antes de la aplicación de los algoritmos de reconocimiento de patrones. Además de la segmentación, usualmente involucra la limpieza y suavizado de trazos (smoothing). Otras tareas de pre-procesamiento a nivel pixel son el adelgazamiento (thinning), la aplicación de códigos de cadena (chain codes), así como la vectorización (vectorization). Dichos procesos serán descritos a continuación.

Adelgazamiento

El adelgazamiento (thinning) es una operación de procesamiento de imágenes en el cual regiones de imágenes binarias son reducidas a líneas que aproximan las líneas centrales, o esqueletos, de las regiones. El propósito del adelgazamiento es reducir los componentes de la imagen a su información esencial para facilitar el proceso de análisis y reconocimiento. Por ejemplo, las mismas palabras pueden ser escritas a mano con diferentes plumas, dando diferentes grosores a los trazos, pero la información literal de las palabras es la misma. Para muchos métodos de análisis y reconocimiento de trazos, es más rápido y fácil trazar líneas de sólo un pixel de ancho que trazar líneas más gruesas. El adelgazamiento es comúnmente usado en el estado de pre-procesamiento de análisis de documentos como mapa de procesamiento [1].

Los requerimientos del adelgazamiento se establecen como sigue:

1. Regiones de imágenes conectadas se deben adelgazar hasta ser estructuras de líneas conectadas.
2. El resultado adelgazado debe ser de mínimo una región 8-conectada (cuya definición veremos más adelante).
3. La localización de los extremos de línea debe ser mantenida.
4. Los resultados del adelgazamiento deben aproximarse a las líneas medias de las regiones originales.
5. Las pequeñas líneas sobrantes causadas por el adelgazamiento deben minimizarse.

Es esencial que los resultados del adelgazamiento mantengan la conectividad entre las líneas, lo que garantiza que el número de estructuras de líneas adelgazadas conectadas sea igual al número de regiones conectadas de la imagen original. Para el requerimiento 2, las líneas resultantes deben contener un número mínimo de pixeles que mantengan la 8-conexión de pixeles (un pixel se considera 8-conectado a otro pixel si el segundo pixel es uno de los ocho vecinos más cercanos a él). Respecto al requerimiento 3, el adelgazamiento puede lograrse por la remoción iterativa de los pixeles de las fronteras exteriores, sin remover los pixeles interiores de una línea. El cuarto requerimiento no siempre puede cumplirse, ya que si se tiene una línea de 2 pixeles de ancho, la línea media quedará entre dos pixeles, y la línea media correrá a un lado de la línea media real. Con respecto al quinto requerimiento, es obvio que el ruido debe ser minimizado, pero a veces es difícil distinguir qué es ruido y qué no lo es. Aunque existen algoritmos que tienen parámetros para remover espurias, puede ser más conveniente que el adelgazamiento y la reducción de ruido sean dos procesos separados.

Códigos de cadena

Cuando los caracteres son descritos por sus esqueletos o contornos, pueden ser representados más eficientemente que por ceros y unos en una imagen raster. Un modo común de hacer esto es mediante códigos de cadena (chain codes), donde los unos son representados como secuencias de pixeles vecinos conectados a lo largo de líneas y curvas. En vez de almacenar la localización absoluta de cada pixel encendido, se almacena la dirección de su vecino codificado previamente. Un vecino es cualquiera de los pixeles adyacentes en la vecindad de 3x3 pixeles alrededor del pixel central.

Hay dos ventajas al codificar la dirección en lugar de las coordenadas absolutas de la localización. Una de ellas es la eficiencia de almacenamiento. Para imágenes mayores de 256x256, las coordenadas de un pixel encendido son representadas usualmente por dos palabras de 16 bits; en contraste, para códigos de cadena con ocho posibles direcciones para un pixel, cada pixel encendido puede almacenarse en un byte, o incluso en tres bits. Una ventaja más importante es que dado que los códigos de cadena contienen información en conexión dentro del código, esto puede facilitar el procesamiento posterior, tal como suavizado (smoothing) de curvas continuas, y análisis tal como detección de características de líneas rectas.

Un código de cadena muy usado es el de las primitivas o PCC (Primitives Chain Code), que codifica las siguientes características: extremos de líneas, bifurcaciones y uniones, y quebrantamientos que indican el inicio de código dentro de un contorno cerrado. Con estas características adicionales los pasos subsecuentes del reconocimiento de patrones se facilitan. Este código proporciona también una compresión más alta que otros códigos, porque limita eficientemente el número de palabras de código al número de posibilidades de 8-conexión, y las empaqueta eficientemente.

Vectorización

Una alternativa al adelgazamiento y códigos de cadena es representar líneas de imágenes por segmentos de líneas rectas que pueden ser dibujados dentro de las líneas originales. Esta aproximación se llama vectorización (vectorization), y una de sus ventajas es que, dado que para este proceso se buscan líneas largas, hay menos espurias de las que resultan de los procesos de adelgazamiento y códigos de cadena [9].

1.3.2 Pre-procesamiento a nivel imagen

La escritura cursiva varía de diversas maneras. Además de las particularidades de la escritura del autor, que significan que un escritor puede ser identificado entre miles, están las particularidades de escribir en diversas situaciones, con diversos medios y para diversos propósitos. En la tarea del reconocimiento de caracteres, toda esta variación es relevante y sirve solamente para confundir las identidades de las palabras, aunque en otros usos, tales como verificación de firmas, este "ruido" puede ser del mayor interés [20].

Una forma de adquirir la variación de algunos de estos procesos debidos al ruido es identificar ciertos parámetros de la escritura a mano que pueden variar entre diversas escrituras de una misma palabra. Entonces, se debe determinar un procedimiento para estimar cada uno de los valores del parámetro de la palabra muestra (o de varias) y finalmente, se debe encontrar otro procedimiento para quitar los efectos de dicho parámetro sobre la palabra.

Algunos de los parámetros a considerar son los que se menciona a continuación.

- **Altura:** la altura de las letras variará entre los autores para la misma letra y para un mismo autor para diversas letras.
- **Inclinación:** la inclinación es la desviación de los trazos de la vertical, variando entre las palabras y entre los autores.
- **Pendiente:** es el ángulo con la línea base de una palabra, si no se escribe horizontalmente.
- **Ancho del trazo:** esto depende de factores tales como el instrumento de escritura utilizado, la presión aplicada, y del ángulo del instrumento de escritura.
- **Rotación:** en el caso de la captura por medio de un escáner, si la página se sesga en él, entonces todas las palabras serán rotadas. Se desea que la rotación sea pequeña y que pueda removerse con una combinación de transformadas de inclinación y corrección de pendiente.

Un buen sistema debe incorporar la normalización de cada uno de estos factores, reduciendo cada imagen a una consistente de letras representadas por trazos de un pixel de ancho, verticales, de altura uniforme, y ubicadas sobre una línea base horizontal [19].

Altura

La altura del carácter es determinada encontrando las líneas importantes que cruzan la totalidad de la letra (desde arriba hasta abajo) en las letras minúsculas. Con estas líneas, los trazos ascendentes y descendentes que usamos los humanos para determinar la forma de una palabra pueden ser identificados.

Línea base o renglón de la palabra

El proceso para la determinación de la línea base de la palabra consiste en los siguientes pasos:

1. Calcular el histograma de densidad vertical, contando el número de pixeles negros en cada línea horizontal de la imagen.
2. Rechazar la parte de la imagen que probablemente sea un trazo descendente, como en las letras g, q, y. Tal trazo es indicado por un pico en el histograma de densidad vertical. El mínimo en el histograma sobre este punto es encontrado y la imagen es cortada de ese punto hacia abajo. Aún para palabras inclinadas esto da una idea suficiente de la localización de los trazos descendentes, así que las líneas inclinadas pueden estimarse por los pasos restantes.
3. Encontrar el pixel restante más bajo en cada línea de exploración vertical.
4. Retener sólo los puntos alrededor del mínimo de cada cadena de pixeles.
5. Encontrar la línea del mejor ajuste a través de estos puntos.

- Rechazar los puntos lejanos de la línea del mejor ajuste y volver a estimar. Ésta se considera ahora la línea base del carácter.

Dada la estimación de la línea base más baja, la escritura puede enderezarse para obtener la línea base horizontal por la aplicación de una transformada paralela al eje y, después, la altura de la línea base más baja puede ser re-estimada, bajo la suposición de que es horizontal ahora. La línea superior se puede estimar usando un procedimiento similar, aunque éste es menos robusto, debido a la presencia de trazos ascendentes, que son más difíciles de separar del cuerpo del texto que los trazos descendentes.

Corrección de inclinación

La inclinación de una palabra es determinada encontrando el ángulo medio de los trazos más cercanos a la vertical. Esto se calcula encontrando los bordes de los trazos, hallando el contorno de la imagen con un umbral o usando un filtro de detección de bordes. Ambas técnicas dan una cadena de píxeles conectados que representan los bordes de los trazos. La orientación de esos bordes cerca de la vertical se utiliza como una estimación de la inclinación total.

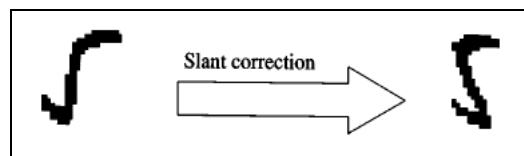


Figura 1.3. Corrección de inclinación aplicada al número 5

Los algoritmos de pre-procesamiento para la corrección de la inclinación se aplican para obtener imágenes de palabras independientes de la inclinación de la escritura de cada autor. La mayoría de los métodos de reconocimiento son tolerantes a pequeñas variaciones en la escritura, sin embargo, su eficiencia se reduce por la presencia de inclinación en la misma. El algoritmo usado se basa en la idea de que si una palabra se rota a través de una serie de posiciones inclinadas, se logra generalmente su anchura mínima cuando se inclina lo menos posible. Considerando a α como el ángulo al cual se desea rotar la palabra en dirección contraria a las manecillas del reloj (y $-\alpha$ el sentido de las manecillas del reloj), la ecuación para transformar un mapa de bits por dicho ángulo es [28]:

$$\forall p(x,y) = 1, b(x',y') = 1,$$

donde

$$x' = x - y \tan(\alpha)$$

$$y' = y$$

Todos los píxeles dentro de la misma fila son movidos horizontalmente por una cantidad dada, pero nunca son movidos a una fila diferente. Cada fila es desplazada a la derecha o a la izquierda (dependiendo del signo de α) por una cantidad que se incrementa linealmente con la altura de la fila sobre el renglón.

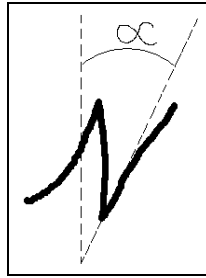


Figura 1.4. Carácter inclinado en un ángulo α , el ángulo entre la dirección vertical y la dirección del trazo

El ángulo que proporciona la posición menos inclinada se obtiene por medio de una estrategia de búsqueda binaria definida para minimizar el ancho total del carácter. El algoritmo inicia con una valor inicial de $\alpha=0^\circ$, y un incremento inicial $\Delta\alpha=20^\circ$. Entonces se analiza $\alpha+\Delta\alpha$ y $\alpha-\Delta\alpha$ para encontrar una mejor solución. Cuando la misma solución se repite, el incremento es dividido por 2. El proceso es repetido hasta que se alcance una precisión de 1° .

Suavizado

Por otro lado, para quitar ruido de la imagen proveniente del documento original, por ejemplo, defectos de escaneo, o la aplicación de transformadas a las imágenes discretas, se puede realizar un suavizado (smoothing) de la imagen con diversos procesos. Después de la normalización y el suavizado, se aplica un umbral a la imagen para dejar cada pixel negro o blanco. El umbral es encontrado simplemente buscando un mínimo entre dos máximos en el histograma de niveles de grises de la imagen. Después se aplica un algoritmo de adelgazamiento iterativo del tipo erosivo sobre los trazos en la escritura, de modo que se reduzcan a un ancho de un pixel.

Parametrización

El método de parametrización más usado por la mayoría de los expertos del área es la codificación del esqueleto de la palabra para pasar la información referente a las líneas del esqueleto hacia el sistema de reconocimiento [1]. En el esquema de codificación del esqueleto, el área cubierta por la palabra se divide primero en una rejilla de rectángulos. Las tramas verticales son de un ancho fijo para la palabra entera, una longitud determinada por la estimación de la altura del carácter. Así, hay un número variable de estas tramas verticales en una palabra, con las palabras largas teniendo más marcos que las palabras cortas, pero un carácter dado ocupará siempre aproximadamente el mismo número, suponiendo además que la altura del carácter es proporcional al ancho del carácter.

La resolución vertical de la rejilla se elige para dividir la palabra en un cierto número de regiones, cada una de las cuales se puede identificar para que juegue un papel definido, pero distinto, en la representación de la escritura cursiva. Las regiones cercanas a las líneas de base superiores e inferiores ya identificadas contienen la mayoría de los trazos horizontales en una palabra, representando los puntos críticos en lo alto y en la base de la mayoría de las letras minúsculas, y las ligaduras entre las letras. Estas dos regiones también contienen los extremos de los trazos cortos verticales. La región media entre estas dos líneas contiene información importante sobre los trazos cortos verticales que conforman la mayoría de las letras cursivas, así como el detalle interno de las letras *e* y *s*. Los trazos ascendentes y descendentes se encuentran en las regiones sobre la línea media y debajo de la línea de base, y dos regiones más se pueden identificar conteniendo los puntos finales

o los ganchos de los trazos ascendentes y descendentes [20], [21].

1.3.3 Segmentación

La segmentación es la separación de unidades de escritura, las cuales pueden ser letras, palabras, etc. Existen dos tipos de segmentación: externa e interna.

La segmentación externa es el aislamiento de varias unidades de escritura previo a su reconocimiento. Dado que varias letras pueden ser escritas con un sólo trazo en la escritura cursiva, se requiere de algún tipo de reconocimiento para su aislamiento. Una vez aislada la unidad a reconocer, en ocasiones es necesario aplicarle otra segmentación a esta unidad para obtener características de ella, a esta segmentación se le llama segmentación interna.

Los primeros métodos de segmentación se basaban en señales explícitas de los usuarios. La segmentación espacial usaba sólo la información de las coordenadas en x para separar las unidades de escritura por sus proyecciones en el eje x. Otros métodos tempranos usaban sólo información temporal para separar unidades de escritura. Cuando la diferencia de tiempo entre el fin de un trazo y el inicio del siguiente excede un cierto tiempo, se asume que el carácter se ha completado [16].

Para caracteres escritos dentro de cajas predefinidas, el escritor hace la mayor parte de la segmentación. Esto se debe a que las tabletas donde se escribe se dividen en regiones donde debe escribirse cada carácter. A menudo, un espacio de tiempo permite a los caracteres ser reemplazados o corregidos re-escribiéndolos. También, la segmentación espacial puede manejar un trazo que atravesase de una caja a otra. Los algoritmos de segmentación para caracteres escritos en cajas predefinidas pueden ser complejos a causa de la oportunidad de usar la información adicional proporcionada por las cajas.

Las unidades de escritura espaciadas por el escritor pueden ser caracteres o palabras. El espaciado de caracteres no es normal en ciertos idiomas; en inglés y español, muchos escritores, aún cuando se les pide que lo hagan, encuentran difícil espaciar los caracteres consistentemente. En contraste, es más natural para las personas espaciar palabras cuando se imprimen o se escriben cursivamente. Algunas técnicas recientes de segmentación espacial prueban con una separación bidimensional de las unidades de escritura. Un método reciente combina información espacial, temporal y de otros tipos para lograr la segmentación de palabras. Finalmente, conociendo el número de caracteres en una cadena se puede lograr la segmentación de una palabra en los caracteres que la forman [17].

La tarea de la segmentación es identificar bloques de texto, líneas, palabras y caracteres, típicamente iniciando con un análisis de conectividad el cual transforma una imagen binaria en un conjunto de componentes conectados. En la escritura cursiva, debe distinguirse entre el cuerpo del carácter y el trazo que lo une con el siguiente carácter, si es el inicio o el fin de la palabra, etc.

Un problema serio en la segmentación lo representa el traslape (vertical u horizontal) de estructuras (líneas, palabras, caracteres, etc.), el cual impide localizar correctamente los puntos de separación de las unidades de escritura [25].

De acuerdo al artículo escrito por Lecolinet y Casey [25], se puede concluir que existen tres estrategias “puras” de segmentación, más otras numerosas aproximaciones que son combinaciones de estas tres. Las estrategias elementales son:

1. La aproximación clásica, en la cual se identifican segmentos en base al parecido de los caracteres en estudio con caracteres de una base de datos. Este proceso de cortar la imagen en componentes significativos recibe el nombre de disección.
2. Segmentación basada en el reconocimiento, en la cual el sistema busca en la imagen componentes que concuerden con ciertas clases en su alfabeto.
3. Métodos holísticos, en los cuales el sistema busca reconocer palabras como un todo, y no a través de cada uno de los elementos que las conforman, sin necesidad de segmentarlas en caracteres.

En la estrategia 1, el criterio para una buena segmentación es el acuerdo de propiedades generales de los segmentos obtenidos con aquellos esperados para caracteres válidos. Ejemplos de tales propiedades son la altura, ancho, separación de componentes vecinos, disposición a lo largo del renglón, etc. En el método 2, el criterio es la confianza en el reconocimiento, quizás incluyendo correcciones sintácticas o semánticas de los resultados totales. Los métodos holísticos en esencia revierten la aproximación clásica con palabras como el alfabeto a leer.

En dicho artículo [25], se presenta una serie de técnicas englobadas dentro de estos tres métodos principales, de las cuales algunas son aplicadas a caracteres impresos, otras a caracteres escaneados provenientes de diversas fuentes, otras a caracteres manuscritos, y otras a caracteres manuscritos cursivos. De las estrategias para estos últimos, para el primer método existe el análisis de proyecciones, el cual usa la proyección vertical (o histograma vertical), para hacer un conteo de los píxeles negros en cada columna de la imagen, lo cual ayuda a detectar espacios blancos entre caracteres, o bien, mediante la aplicación de un umbral, se puede encontrar las ligaduras entre ellos.

En el reconocimiento de escritura cursiva, es común analizar la imagen de una cadena de caracteres para definir las zonas superior, inferior y media. Esto permite la detección de trazos ascendentes o descendentes, características que pueden servir como marcas para la segmentación de la imagen, ya que pueden encontrarse las ligaduras presentes en la escritura.

Otro método consiste en encontrar grafemas, es decir, no caracteres como tales sino combinaciones de letras o porciones de estas. La separación se basa en la detección de áreas características en la imagen. Las clases reconocidas por un clasificador no corresponden a letras, sino a formas específicas que pueden ser segmentadas confiablemente. El concepto de grafema ha sido aplicado principalmente a la escritura cursiva por muchos investigadores. Las técnicas para diseccionar curvas se basan en reglas heurísticas derivadas de la observación visual.

Los métodos existentes para tratar el problema de la segmentación en el reconocimiento de caracteres se han desarrollado notablemente en la última década. Una variedad de técnicas se ha combinado, influenciadas por los desarrollos en campos relacionados tales como el reconocimiento en línea y otros. La dificultad de realizar una segmentación exacta está determinada por la naturaleza del material a leer y por su calidad. Generalmente, los porcentajes de segmentación errónea para material libre de restricciones se incrementan progresivamente desde los caracteres impresos hasta la

escritura cursiva manuscrita. Entonces, las técnicas simples basadas en los espacios blancos entre caracteres son suficientes para caracteres limpios de ancho fijo. Para escritura cursiva realizada por muchos escritores y un vocabulario grande, en el otro extremo, se emplean métodos de mucha mayor sofisticación [25].

Los investigadores actuales emplean modelos no sólo de caracteres, sino también de palabras y frases, y aun de documentos enteros, y poderosas herramientas como los HMM (Hidden Markov Models), redes neuronales, métodos contextuales, entre muchos otros [18]. Creemos que un uso inteligente del contexto y la confianza del clasificador ha conducido a exactitudes mejoradas, pero hay pocos datos experimentales que permitan que una valoración de la cantidad de mejora sea atribuida a técnicas avanzadas.

Muchos investigadores se han dado cuenta de la importancia que la segmentación juega en el proceso de reconocimiento, ya que a ella se debe parte del éxito o fracaso del reconocedor, ya que si no se posee un buen algoritmo de segmentación de palabras, los índices de reconocimiento decrecen notablemente. Es por eso que muchos métodos son necesarios para ser empleados y comparados con el trabajo de otros investigadores.

1.4 Extracción de Características de los Caracteres Manuscritos

1.4.1 Generalidades

Un sistema de reconocimiento de caracteres, requiere del análisis previo de una serie de características presentes en la escritura que pretende reconocerse, tal como ocurre en el ser humano y su proceso mental de reconocimiento. Primero se extrae de manera visual una serie de características de las letras, de tal forma que al aprenderlas y asociarlas, nos permitirán reconocer cada carácter diferenciándolo de otro.

Ese análisis es fundamental para el desarrollo del sistema, pues de él, parte el proceso de extracción. Como resultado se obtiene un conjunto de características que describirán a cada carácter, de tal forma que al ser comparadas, el sistema pueda diferenciar un carácter de otro.

El proceso de reconocimiento de caracteres y en particular el algoritmo de extracción, involucra el conocimiento de una serie de características y diferencias de los caracteres a reconocer, para realizar, a partir de ellas, los procesos posteriores convenientes que le permitan al sistema de reconocimiento, hacer la distinción correcta entre caracteres. Existen algunos casos en los que el parecido entre caracteres es tal, que para diferenciar a uno de otro, requerimos de la comparación de una mayor cantidad de características. En otros casos, el análisis del contexto puede ayudarnos en esta tarea.

El alfabeto nos permite representar de manera gráfica el lenguaje mediante una serie de elementos gráficos llamados letras, en conjunto con otros símbolos y signos de puntuación. Las características asociadas a cada uno de los caracteres determinan una diferencia visual reconocible. Esa diferencia, es precisamente, la que permite el desarrollo de nuestra capacidad de reconocimiento y a su vez ese reconocimiento, permite el proceso de comunicación. Nuestro alfabeto está formado por 26 letras y cada una de ellas puede representarse en dos formas: mayúscula y minúscula. Los estilos básicos para su escritura son el de imprenta o molde y el cursivo, que es el de interés para

nuestro sistema. El estudio de las características particulares de este tipo de estilo, será conveniente para todos los procesos posteriores. La escritura cursiva consiste en una secuencia de caracteres entrelazados, que típicamente presentan una inclinación de izquierda a derecha, la que puede variar según el dibujante. En este tipo de escritura se presentan también cruces y puntos, como en el caso de las letras t, i, x y j. Para ellas, primero se dibuja la parte inferior y después se completa el dibujo de la letra con el cruce o el punto [18].

El dibujo de una letra se realiza mediante una serie de movimientos en donde la mano recorre trayectorias que inician comúnmente en la parte inferior del renglón y terminan en la parte superior y viceversa. Cada dibujo se compone de una serie de trayectorias cortas a las que se denominan trazos o golpes, como traducción de la palabra *stroke*, muy comúnmente usada en los artículos en inglés sobre el tema. Las letras se pueden dibujar con uno o más de dichos trazos. La posición y medida de la letra también son características importantes que pueden presentar variaciones. Las letras mayúsculas se sitúan en un renglón y ascienden hasta completar una cierta medida. En cambio, las letras minúsculas son pequeñas, como de la mitad del tamaño de las mayúsculas. Algunas minúsculas ascienden, se extienden hasta alcanzar el tamaño de las mayúsculas, algunas descienden extendiéndose hacia abajo del renglón y otras presentan ambas características.

Todos los caracteres presentan variaciones en sus propiedades estáticas y dinámicas. Las variaciones estáticas se refieren al tamaño, la forma, la inclinación, características que pueden ser observadas en la zona de escritura o de despliegue de la información capturada. Las variaciones dinámicas son aquellas que no se representan necesariamente en el dibujo de la escritura, sino en el entorno de escritura y es posible registrarlos por medio de dispositivos adecuados como sensores de presión, velocidad o movimiento virtual de la pluma al escribir. Los caracteres pueden presentar además ciertos retrasos, por la repetición de un trazo (sobre-escritura). Los grados y tipos de variación dependen básicamente del estilo y la rapidez de la escritura.

Distinguir las distintas formas entre caracteres es una tarea que se vuelve más difícil para aquellos caracteres que son muy similares, como en el caso de las siguientes parejas de letras: u-v, a-o, a-d y n-h, principalmente. Esta situación ocurre también entre letras y números como en los casos siguientes: 0-O, 1-l, Z-2, S-5, 6-G; de hecho, algunas de las parejas anteriores podrían dibujarse completamente igual y lo único que nos podría ayudar a distinguirlos sería el contexto. Para las letras mayúsculas y minúsculas que son muy similares, la diferenciación se puede hacer considerando el tamaño de la letra y en otros casos, la posición de la letra en el renglón [16].

1.4.2 Métodos para la extracción de características

Hemos visto ya anteriormente, que son muchas las características que pueden ser extraídas de un carácter. Dichas características cuentan con propiedades estáticas, dinámicas o ambas. En esta sección se comentará algunos de los métodos generales utilizados para la extracción de determinadas características.

Algunos métodos consideran ciertas características que pueden estar o no presentes en la escritura, utilizando con ello procesos binarios, evaluando cada carácter sólo con dos opciones, la presencia o ausencia de una característica determinada. Por ejemplo al observar un carácter, podemos saber si en él existe o no una cierta característica como un punto por ejemplo, o si en él se presenta o no un trazo que descienda. Cuando se analiza a un carácter de esta manera se puede recurrir a un

árbol de decisiones, con el que podremos descartar en el reconocimiento ciertas opciones. Para letras minúsculas, si se presenta un descenso bajo el renglón, los caracteres sólo podrían ser las letras g, j, p, q, y, sin embargo, si además se presenta un punto, el carácter definitivamente sería una j. El uso de árboles de decisión es sencillo, y nos permite reducir el conjunto de candidatos, aunque generalmente se combina con otros métodos. Por supuesto que existe otro tipo de características que no son binarias y además son éstas las que más ocurren y dan complejidad al reconocimiento, de ellas hablaremos mucho más [16].

a) Métodos de análisis en el dominio de la imagen

Los métodos de extracción que involucran el análisis de secuencias temporales de zonas, direcciones o extremos, son métodos que nos brindan información sobre las características dinámicas. De esta manera, una secuencia de códigos que contengan una serie de características de este tipo, podrán representar a un carácter. Las zonas de análisis se especifican por divisiones dentro de un rectángulo (o zona de escritura) que contiene al carácter. Después de seccionar el área y extraer las características, el código resultante del carácter desconocido es comparado con un diccionario de códigos de secuencias de zonas que permitirá su reconocimiento.

Un método similar utiliza una secuencia de direcciones del movimiento de la punta de la pluma durante la escritura del carácter. Utilizando cuatro direcciones básicas: arriba, abajo, izquierda, derecha, un sistema codifica las primeras cuatro direcciones de la secuencia, y entonces clasifica al carácter haciendo una revisión de una tabla que tiene 256 entradas (4x4x4x4). A medida que el número de direcciones e intervalos de tiempo se incrementa, revisar la tabla se va volviendo menos práctico, y las secuencias son entonces comparadas por pareado de curvas (curve matching) [16].

El pareado de curvas (curve matching) es un método popular de procesamiento de señales, en el cual se obtienen curvas de un carácter prototipo y el prototipo que más se ajuste se asigna al carácter desconocido. Las curvas son usualmente funciones del tiempo, de los valores en X y Y, la dirección de la tangente del ángulo de la trayectoria de la escritura o ambas. Usando un código de ocho direcciones, un carácter es dividido en seis o diez regiones de tiempo.

Otro método es el de código de trazos (stroke codes), que clasifica segmentos de un carácter y después identifica al carácter desde la secuencia de segmentos ya clasificados [9]. En este caso se recurre al uso de árboles de decisión para el código de secuencias de trazos. Los diferentes ángulos de los segmentos han sido usados para distinguir ciertas curvas de los caracteres. Las direcciones de los ángulos se usan como primitivas en la obtención de un modelo descriptivo de los caracteres.

Otro acercamiento a la solución del problema de reconocimiento es el análisis por síntesis (analysis-by-synthesis), algunas veces también llamado reconocimiento por generación [17]. Se han hecho ya muchos estudios concernientes al modelado de generación de escritura manuscrita. Esos modelos usan generalmente trazos (o segmentos de trazos) y reglas para conectarlos y construir los símbolos. Los símbolos generados a partir de un inventario de trazos constituyen la representación ideal estándar de los caracteres. Una aproximación a la escritura real puede hacerse atendiendo a especificaciones de esos trazos con modelos matemáticos que describen el movimiento de la pluma como una función del tiempo. Entonces, la palabra manuscrita puede ser dividida en trazos, los trazos clasificados usando parámetros del modelo y una secuencia de letras y palabras reconocidas.

Un análisis similar utiliza programación dinámica para parear trazos reales y modelados. Relacionado al análisis por síntesis existe una teoría de percepción de escritura manuscrita en la cual la información dinámica se infiere a partir de la estática [27].

Se han instrumentado también estudios perceptuales en el desarrollo de métodos de distinción de pares (pairwise distinction) [23]. Aquí, un procedimiento especial separa cada parte de los caracteres que podrían ser confundidos. Estudiando la manera humana de distinción se permite a la teoría de pares basarse en atributos funcionales.

La distinción de pares por atributos funcionales ha permitido aumentar notablemente la eficiencia de los métodos de reconocimiento en los sistemas comerciales. Algunas veces los mismos atributos diferencian a más de un par de caracteres. Por ejemplo, en el caso de la extensión de la línea que diferencia tanto al par D-P como al V-Y.

Otro método representa al carácter por el número, orden y posición relativa de los trazos. Algunos trazos son divididos en más partes, particularmente aquellos caracteres con pocos trazos. El método estadístico de un modelo de Markov es particularmente conveniente para información dinámica. Por ejemplo, un modelo de Markov de primer orden usó ocho estados correspondientes a ocho direcciones de la punta de la pluma [16].

b) Análisis en el dominio de la frecuencia

Una alternativa a los métodos ya mencionados consiste en la comparación de los coeficientes de Fourier obtenidos de las curvas $x(t)$ y $y(t)$. Este método es apropiado cuando los caracteres pueden ser representados por un número razonablemente pequeño de coeficientes de Fourier. El uso de este método es conveniente cuando predominan los trazos curvos, como en los números o en las concatenaciones de trazos.

El pareado de curvas en el dominio de la frecuencia (transformada de Fourier) es equivalente al pareado de patrones (pattern matching) en el espacio característico cuando el número de puntos que caracterizan a las curvas es constante y en correspondencia uno a uno. Es en realidad una alineación de la curva. Sin embargo, debido a ciertas no linealidades, el mejor ajuste no es usualmente una alineación o pareado lineal [16].

Para muchos problemas en la secuencia de comparación, el pareado elástico (elastic matching) ha sido exitoso. Este método ha sido aplicado a caracteres alfanuméricos y chinos. Como el pareado elástico es un método computacional complejo, los prototipos son frecuentemente primero seccionados para reducir el número de pareados.

La aplicación de una transformación local puede realzar la discriminación de forma del pareado elástico. Usando la correspondencia de puntos para el pareado elástico entre entradas y patrones de referencia, se genera una Deformación en el Campo Vectorial (DVF). Entonces, la DVF es aproximada mediante la aplicación de iteraciones significativas de una Transformación Local Afín (LAT). Finalmente, un pareado elástico entre el patrón de entrada y el patrón de referencia deformada superpuesto por componentes LAT de bajo orden aumenta la discriminación de forma, reduciendo a la mitad la tasa de error [20].

Existe también un método por síntesis, que se realiza a partir de la transformada de Fourier, al aplicarle esta transformación a la función de tiempo obtenida por el método en el dominio de la imagen. Entonces, la función puede ser dividida en trazos, los trazos clasificados usando parámetros del modelo y una secuencia de letras y palabras reconocidas [16].

1.5 Clasificación de Características

El paso posterior a la extracción de características, consiste en su comparación, con la finalidad de obtener el reconocimiento del carácter. Aquí se vuelve necesario el uso de una base de datos que nos sirva como referencia para la localización de las características guardadas para los caracteres denominados prototipos. Debido a lo anterior, un sistema de reconocimiento no puede dejar a un lado procesos relacionados con este tipo de bloques de comparación y reconocimiento. En esta sección se describirán los procesos mas comúnmente utilizados en ellos.

1.5.1 Selección de características

El proceso de selección permite la obtención de un conjunto de características, que nos servirá para el reconocimiento final, a partir de su comparación con ciertas referencias. Esta selección está involucrada en el proceso de extracción de características, en el análisis previo a la aplicación de los procesos de extracción, en donde se define el conjunto de características necesarias. Sin embargo, en ciertas ocasiones una revisión de este conjunto, antes de la comparación y la creación de la base de datos, permite descartar aquellas características que son poco determinantes para la diferenciación entre caracteres.

1.5.2 Clasificación de características

La clasificación de características pretende la ordenación de las mismas de tal forma que sea posible su identificación, por medio de códigos o claves más convenientes para su localización. Esta clasificación permitirá agilizar y definir el proceso de comparación entre las características del carácter y la localización inmediata del carácter prototipo asociado al desconocido.

Al clasificar las características, podremos hacer una ordenación de las mismas, lo que nos permitirá elaborar una base de datos, con la estructura definida por la clasificación. Para ello se realiza un proceso de reducción de información, en el cual, a partir de una gran cantidad de características extraídas de los dibujos para un mismo carácter, se marca una estructura de ordenación para el almacenamiento de las mismas, con ello procesos posteriores para la extracción de promedios y estadísticos son fácilmente realizados, guardando como resultado la información prototipo para cada carácter, con una estructura convenientemente definida; pensando además en la utilización de esta información representativa en la comparación posterior entre caracteres prototipo y prueba. Esa información general, ordenada según una estructura conveniente que guarda información acerca de los caracteres denominados prototipo, formará la base de datos del sistema.

Para el reconocimiento del carácter prueba, su vector de características deberá cumplir con la estructura definida, para poder ser comparado con los vectores prototipo de la base de datos.

A continuación se enumeran algunas características resultantes del proceso de selección y extracción que pueden servir como ejemplo de características requeridas por diversos sistemas, a las que será necesario aplicar un proceso de clasificación con la finalidad de definir la estructura de la base de datos para la comparación y posterior reconocimiento de carácter prueba.

1. Puntos: la presencia y ausencia de puntos, es una característica que ciertos sistemas de reconocimiento extraen con la finalidad de reducir el conjunto de caracteres posibles. Pueden identificarse puntos sobre las letras i y j con un sencillo conjunto de reglas o barridos. Las ocurrencias de trazos cortos y aislados sobre las líneas medias de la escritura son marcadas como puntos.
2. Uniones: ocurren cuando dos trazos se unen o se cruzan y es fácil encontrarlos en el esqueleto del carácter como puntos con más de dos puntos vecinos.
3. Extremos: son puntos en el dibujo con sólo un punto vecino y marcan el fin de los trazos, aunque algunos son artificios del algoritmo de esqueletización (skeletonizing).
4. Puntos de viraje (turning points): son puntos donde un segmento del dibujo cambia su dirección hacia arriba o hacia abajo, hacia la derecha o hacia la izquierda.
5. Lazos: se obtienen del análisis de componentes conectados en la imagen, para encontrar áreas no conectadas con la región que rodea la palabra.

Al ser clasificadas, cada una de éstas características pueden ser codificadas por un número para su posterior identificación en el algoritmo de reconocimiento [18].

1.5.3 Bases de datos

Una base de datos es una colección lógica y coherente de datos con algún significado inherente. Para un sistema de reconocimiento una base de datos es la colección de información específica de los caracteres denominados prototipos, requerida por el sistema y ordenada según una estructura conveniente para su utilización posterior en la etapa de comparación [31].

Un conjunto de datos aleatorios no puede ser considerado una base de datos, ya que si queremos imaginárnoslas, son como una tabla (tipo Excel) donde cada fila es un registro, que se conforma por campos (en este caso, cada celda). Una base de datos se diseña, construye y maneja con información para un propósito en específico. Tiene un grupo de usuarios y aplicaciones de interés para los usuarios.

Un sistema administrador de bases de datos (Data Base Management System, DBMS) es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Definir la base de datos consiste en especificar los tipos de datos, estructuras y restricciones de la información a ser almacenada en la base de datos. Construir la base de datos consiste en guardar los datos en un medio de almacenamiento controlado por el DBMS.

Un concepto importante es el relativo a las Bases de Datos Relacionales (Relational Data Base, RDB), uno de los puntos más importantes para implementar grandes bases de datos. Las RDB

son, a diferencia de las tradicionales, un conjunto de tablas donde se divide por categorías la información, lo que agiliza el proceso de búsqueda y actualización.

Definir una base de datos relacional implica un proceso, a veces muy complejo, de darle la estructura lógica que realmente agilice los procesos, puesto que si creamos excesivas tablas y fragmentamos demasiado la información, puede resultar contraproducente. Solicitar tres datos que se encuentran en tres tablas diferentes y cada una de ellas, de tamaño considerable, utilizando una rutina inapropiada, es más tardado y confuso que una base no relacional.

En este punto es muy importante saber que tipo de base de datos utilizaremos, puesto que en combinación con el lenguaje que escojamos para explotarla, la flexibilidad y velocidad que nos proporcione definirá nuestra posibilidad de crecimiento, tanto de información como de mecanismos de explotación.

Otro concepto que se debe entender es el relativo al Structured Query Language o Lenguaje de Búsqueda Estructurada (SQL). Este es un lenguaje estandarizado para solicitar información de una base de datos, habiendo sido desarrollada una primera versión por IBM a mediados de la década de los 70 e introduciéndose comercialmente en 1979 por Oracle Corporation.

Hemos mencionado bases de datos, un tipo de lenguaje para su explotación (SQL) y los DBMS, a primera vista parecen entes independientes, siendo que en realidad están muy ligados entre sí, al grado que muchas bases de datos son relacionales, tienen su propio SQL integrado y hasta conforman DBMS cuando se conjuntan con otros programas, por lo que elegir una, es elegir una solución, con sus propias ventajas y desventajas, siendo necesario conocer algunas de las propuestas más importantes, sus características y costos [32].

Otros conceptos importantes respecto a las bases de datos son los siguientes.

- Independencia entre datos y programas: cambiar las características de un dato no afecta a los programas que lo utilizan.
- Los programas de acceso de DBMS son independientes de cualquier archivo específico, la estructura de los datos se almacena en un catalogo separado de los programas de acceso.
- Modelo de datos: es un conjunto de conceptos que se pueden utilizar para describir la estructura (tipos de datos, relaciones y restricciones que se deben mantener en la información) de una base de datos.

Categorías de modelos de datos

- Alto-nivel o conceptual: parecido a como percibimos los datos.
- Bajo-nivel o físico: provee detalles de cómo se almacena la información en disco.
- Representacional o implementación: pueden ser entendidos por usuarios finales pero se parecen a los datos tal como se organizan en la computadora.

Modelos de datos de alto nivel

- Entidades: representan objetos o conceptos del mundo real.
- Atributo: representan una propiedad de interés que describe a una entidad.
- Relación: una relación entre dos o más entidades representa una interacción entre las entidades.

1.5.4 Comparación y reconocimiento

Una vez creada la base de datos y aplicados los algoritmos de extracción de características para los caracteres prueba, se puede realizar ya la comparación. El proceso de comparación pretende la obtención de un carácter prototipo relacionado con la base de datos, que será asignado como carácter reconocido al carácter prueba. Como puede observarse, los procesos de comparación y reconocimiento van de la mano, el resultado de uno es propiamente el otro. El proceso de comparación deberá ser lo más adecuado a la información que se deba comparar, porque de eso depende también en gran medida el buen reconocimiento. Qué tan cercano o semejante sea el código que representa al carácter prueba, con cada uno de los que representan a los caracteres prototipo, dependerá además de las características propias del carácter, del proceso de comparación. Si las características son las adecuadas para un determinado carácter, tenemos que asegurar que nuestro proceso de comparación lo detecte. Comúnmente las características obtenidas son representadas con valores numéricos, los cuales pueden ser fácilmente comparados utilizando una diferencia lineal o una distancia geométrica.

El reconocimiento es el resultado casi inmediato del proceso de comparación, pues el vector, código, conjunto de características, etc. de la base de datos que más cercano sea al de prueba será asignado como carácter reconocido o pre-reconocido, en caso de que el sistema utilice un posterior proceso de validación. Puede ocurrir que como resultado del proceso de comparación, todas las diferencias sean muy grandes, en tal caso el carácter quedará como no reconocido, porque aunque existen diferencias debidas principalmente a ciertas variaciones en la escritura, se acepta como límite de variación un cierto valor para cada carácter con la finalidad de asegurar al sistema que ese carácter sea el buscado. En este caso, todos los caracteres son descartados, y el sistema no arroja ningún reconocimiento.

Hay dos grandes ramas para el reconocimiento de un carácter manuscrito: los reconocimientos globales y los analíticos [29].

Los métodos de reconocimiento de letras manuscritas por comparaciones globales son poco utilizados porque están mal adaptados a la variabilidad de la escritura. Sólo algunos métodos basados en la programación dinámica funcionan para este caso. Los métodos más usados con este enfoque se basan en métodos estadísticos, estructurales, neuro-miméticos, markovianos o mixtos, basados en inteligencia artificial.

Los métodos analíticos, ya sea geométricos o estadísticos, consisten en extraer del carácter dado, un conjunto de m medidas que constituyen la composición de un vector de un espacio de representación R_m de dimensión m . Estas medidas suelen ser elevadas en número ($m > 100$) para el

reconocimiento fuera de línea. Sus componentes fundamentales son medidas topológicas y métricas, como superficies, regiones, perfiles, concavidades, bucles, intersecciones, etc. [29]. El sistema que presentaremos más adelante considera este tipo de enfoque para el reconocimiento.

Existen también los denominados métodos estructurales, que involucran una esqueletización de los caracteres, de manera que pasan por una detección de contornos interiores y exteriores para encontrar una serie de puntos singulares, de los que se extraerá información topológica. Posteriormente, se aplicará un proceso de vectorización (vectorization), que permite representar una descripción en forma de cadena de símbolos o de gráficos.

Los métodos Neuro-Miméticos están basados en la utilización de redes neuronales de todo tipo. La estrategia general parte de una imagen con caracteres normalizados o con símbolos. Los resultados obtenidos son buenos, con la condición de disponer de grandes bases de datos de entrenamiento. Como contrapartida, los tiempos de entrenamiento son elevados, además es muy difícil determinar las causas de errores de estos métodos [29].

Básicamente los HMM (Hidden Markov Models) son máquinas de estados finitos que, en este caso, se utilizan para modelar las secuencias de vectores de características derivados de instancias de los caracteres a reconocer [62].

Los métodos basados en inteligencia artificial se aplican en sistemas de decisión conforme a reglas dentro del proceso de reconocimiento. Son algoritmos de búsqueda de árboles, que se sustentan en la base de algoritmos clásicos. Los métodos de Zadeh se basan en la lógica difusa, y están muy bien adaptados al reconocimiento de caracteres con un tipo de datos impreciso. Usan máquinas de estados o reglas de Zadeh. Finalmente, los métodos Mixtos consisten en una combinación de algunos de los métodos anteriores, como por ejemplo un sistema CTRBF, que combina un árbol de decisión con una red neuronal.

En el caso de reconocimiento de palabras, existen también dos ramas diferenciadas: la del reconocimiento global y la del analítico. Los métodos de reconocimiento globales, son determinísticos o probabilísticos, y se basan en la localización de ciertas cadenas de símbolos que representan a las palabras para buscar una comparación con un patrón determinado.

Los métodos analíticos son una generalización de los métodos usados en el reconocimiento de caracteres aislados. Es interesante señalar que existen métodos, llamados implícitos y derivados de los markovianos de caracteres aislados, que tratan de predecir la siguiente letra en la palabra en función de un diccionario. Esta solución, aunque está limitada al lenguaje a reconocer, ha dado muy buenos resultados.

Con la información de este capítulo, se pretende brindar al lector los conceptos básicos en el área de los sistemas de reconocimiento, tanto de caracteres como de palabras, que le permitan seguir el enfoque dado al sistema para el reconocimiento de escritura cursiva manuscrita que hemos desarrollado, y que se presentará en los capítulos siguientes, esperando que las generalidades anteriores sean útiles para tal fin.

Capítulo 2

Captura

En este capítulo veremos como se realiza la captura de la escritura manuscrita cursiva, que analizaremos durante el desarrollo de nuestro estudio. Principalmente estudiaremos dos aspectos: el referido al hardware de captura utilizado (tarjeta digitalizadora), en el cual mencionaremos sus características y su forma de comunicación con la computadora huésped, mientras que en el segundo aspecto nos referiremos a la programación del hardware, lo que comprende la programación de la comunicación del dispositivo mencionado, así como la captura y despliegue de la información obtenida de él.

En la Figura 2.1, en la que se muestra el diagrama a bloques del sistema de RCM propuesto, se resalta este primer paso a efectuar dentro de la secuencia de procesamiento para el reconocimiento de caracteres manuscritos (letras y palabras) objetivo de este trabajo. Del mismo modo, se muestran esquemáticamente los procesos a realizar posteriormente.

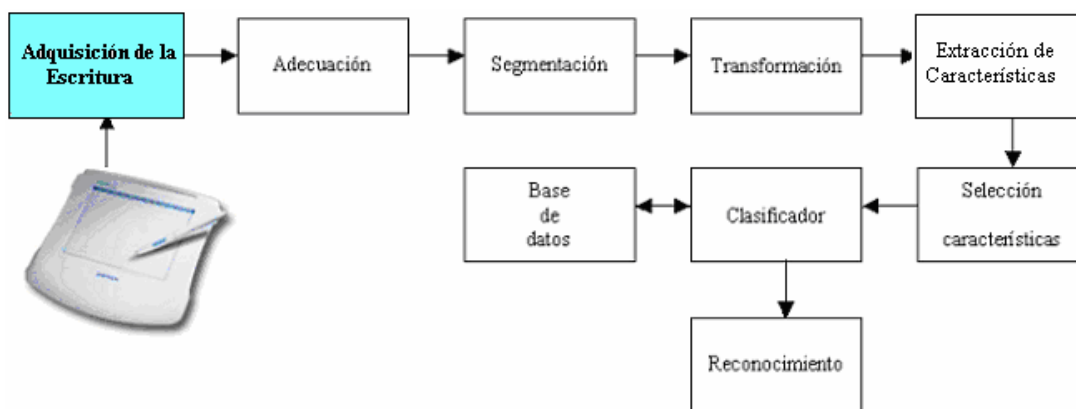


Figura 2.1. Ubicación del bloque de captura en el diagrama general del sistema de RCM

2.1 Hardware de captura

Para nuestro sistema, utilizamos como herramienta hardware de captura una tableta digitalizadora Acecad III, de 5 x 5 pulgadas (12.7 cm x 12.7 cm), fabricada y distribuida por la empresa ACECAT. Este dispositivo puede trabajar en línea o fuera de línea, esto significa que la

información puede ser mostrada a medida que se va generando, o se puede guardar hasta que se acaba de escribir para después mostrarla, respectivamente; en nuestro sistema el dispositivo se encuentra en línea, por lo cual podemos observar los trazos conforme se van generando, sin esperar a que se termine de escribir para ver el o los caracteres escritos.

La tecnología utilizada por esta tableta se conoce como tecnología electromagnética / electrostática, esto significa que dicha tableta contiene una malla de conductores en los ejes “x” y “y”, espaciados 0.1 cm dentro de ella, y una espiral de alambre en la punta de la pluma incorporada a la tableta. La posición de la punta de la pluma se determina como sigue. Ya sea la malla o el lazo es excitado con un pulso electromagnético, y el otro detecta el voltaje inducido o la corriente en una señal senoidal. Los conductores de la tableta son escaneados para localizar el par más cercano a la espiral, y se realiza una interpolación para determinar la posición precisa entre esos dos conductores y realizar la traslación de voltaje a un punto en el espacio cartesiano de dos dimensiones.

La interfaz utilizada por esta tableta es el puerto serial con conector RS-232, a consecuencia de esto, el dispositivo de interfaz serial necesita una conexión de energía separada; en nuestro caso, dicha conexión de energía se realiza mediante una conexión en paralelo junto con el ratón de nuestra computadora huésped.

La pluma de la tableta se encuentra conectada por medio de un cable, del cual obtiene su alimentación. La pluma tiene dos interruptores o botones en su cuerpo, y un tercero se encuentra en la punta, el cual es el más importante para nosotros, ya que como se explicara más adelante este será configurado como nuestro botón principal.

En la Figura 2.2 se muestran las partes de la tableta utilizada en nuestro estudio, cuyas partes describimos a continuación.

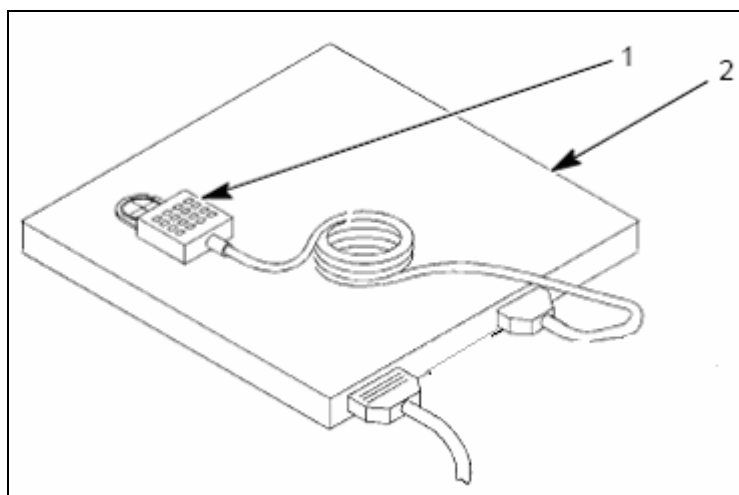


Figura 2.2. Partes principales de la tableta del digitalizador

1. Rejilla de detección
2. Tableta

La tableta funciona con una computadora huésped que usa una interfaz RS-232, como la que se muestra en la Figura 2.3.

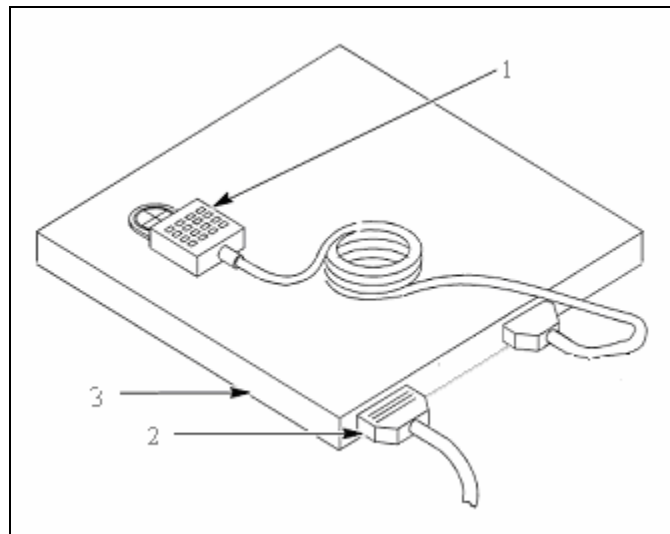


Figura 2.3. Partes del digitalizador

1. Rejilla de detección
2. Conector de puerto RS-232
3. Tableta

Un cursor electromagnético (pluma) se conduce sobre la rejilla de detección interna de la tableta. De las señales recibidas, el procesador calcula la posición del cursor. Después de ser escaladas y compensadas, las coordenadas se transmiten a la computadora huésped.

La salida de la posición del cursor está en el formato de coordenadas cartesianas (XXX, YYY). Dicha posición es un valor numérico representado en código ASCII de 3 dígitos de longitud; este valor numérico indica también el estado de la pluma y los estados de sus botones.

2.2 Software de captura

El software del controlador utilizado es el proporcionado por el fabricante, el cual nos permite controlar el funcionamiento de la tableta, así como su configuración para obtener un rendimiento óptimo para nuestro sistema. En la Figura 2.4 mostramos la ventana de información de la tableta, en la que se muestran, tanto las medidas de la tableta en su zona de escritura, así como su resolución en puntos por centímetro cuadrado, la cual es un parámetro muy importante para nuestro sistema, ya que afecta directamente la nitidez de la imagen de la escritura capturada. Dicha tableta tiene 197x197 puntos por centímetro cuadrado.

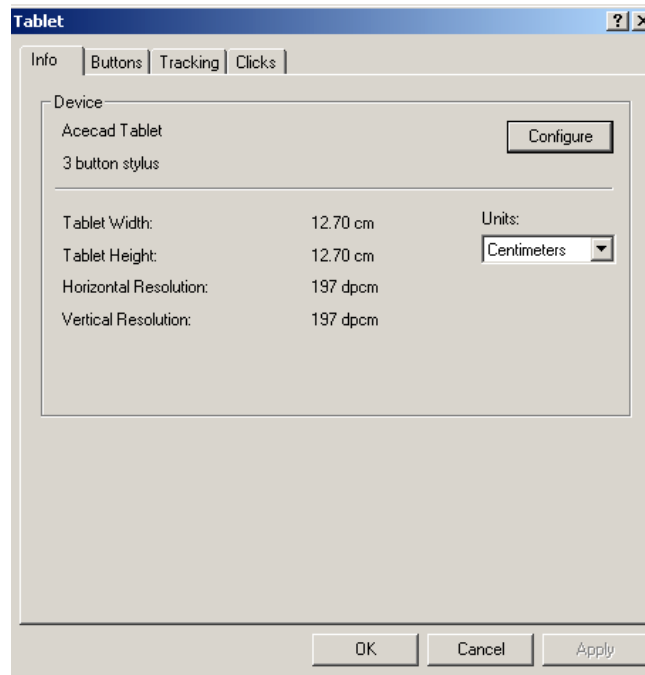


Figura 2.4. Información de la tableta

Para nuestras necesidades configuramos los botones de la pluma mediante sus controladores, teniendo como botón principal el que se ubica en la punta de la pluma, activando su evento mediante un click como se muestra en la Figura 2.5

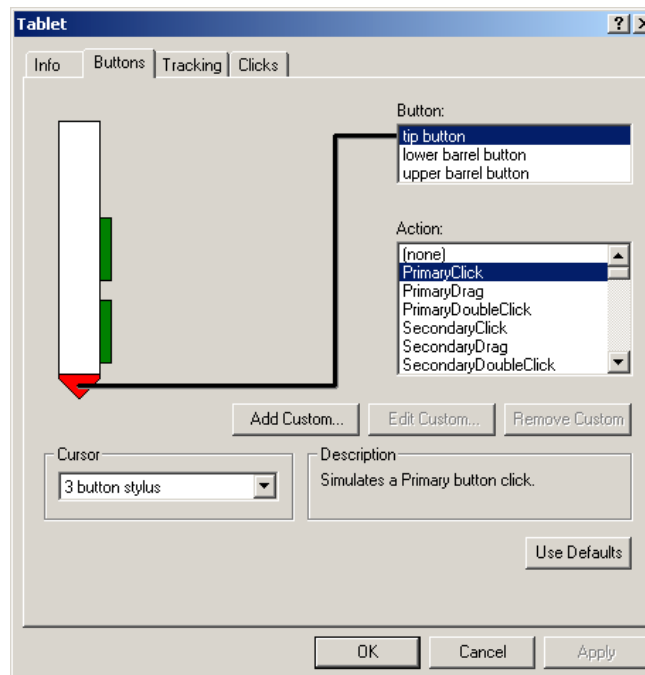


Figura 2.5. Configuración de los botones de la pluma

Además de configurar la pluma, también realizamos una primera configuración de la tableta, tanto de su área activa, de su sensibilidad, la cual utilizaremos en su configuración más sensible (rápida), así como del tipo de movimiento realizado, este movimiento se refiere al modo como se trasladarán los movimientos de la pluma en la tableta hacia el monitor, el cual será absoluto, como podemos observar en la Figura 2.6.

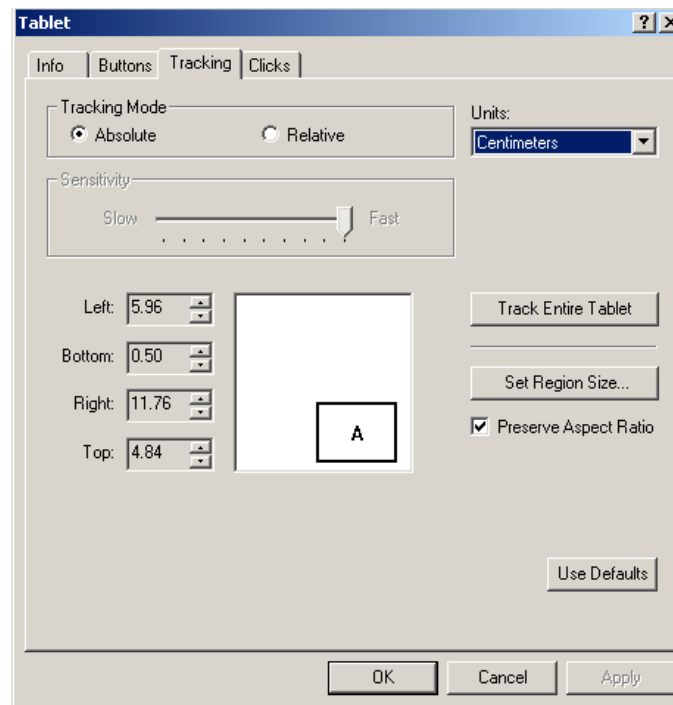


Figura 2.6. Configuración del área activa de la tableta

El digitalizador debe funcionar en el modo de punto, para hacer salir solamente un punto de referencia cuando el operador pone la pluma en la superficie activa del digitalizador. El puerto de comunicación se fija para funcionar con una configuración realizada mediante sus controladores, tal configuración debe corresponder con la programación de la adquisición de datos.

La configuración del puerto de comunicación consiste principalmente en la velocidad de recepción del puerto de datos, la cual se fija en 9600 bits por segundo; la cantidad de bits de datos a manejar por paquete, que en nuestro caso serán paquetes de 8 bits; y la paridad que maneja dicho paquete de datos es cero, contando con un bit de parada. En nuestra configuración no utilizamos ningún control flotante. Dichas propiedades se configuraron dentro de la interfaz del puerto 1 de comunicaciones (COM1), como se muestra en la Figuras 2.7.

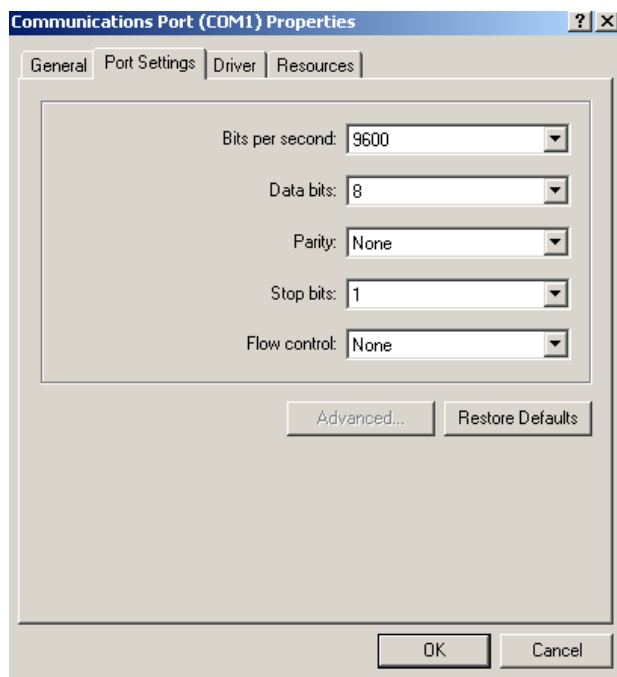


Figura 2.7. Ventana de configuración del puerto de comunicación con la tableta

En la Figura 2.8 podemos ver el resultado de la configuración realizada para comunicar el puerto de comunicación con la tableta digitalizadora.

Driver \ DIP-Switch	1								2							
Graphtec WINTAB	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
Graphtec ADI	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
Summagraphics MM	1	3	3	4	5	6	7	8	1	2	3	4	5	6	7	8
	OFF	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	ON	ON

Figura 2.8. Configuración del puerto de comunicación con la tableta

Un aspecto muy importante es la manera como recibimos la información del dispositivo de captura. En nuestro caso, como se ha mencionado, nuestro dispositivo se conecta mediante un puerto de comunicación en serie, en la llamada conexión dura, donde los dos dispositivos de comunicación se interconectan para intercambiar señales eléctricas, y mientras que uno recibe la información (receptor), el otro la transmite (transmisor) en forma directa de computadora a computadora. En dicha conexión tanto el receptor como el transmisor solo funcionarían como tales, esto quiere decir que no intercambiarían funciones.

2.3 Interfaz de captura

Con la finalidad de mostrar las diferentes etapas del proceso de reconocimiento de caracteres y palabras manuscritas cursivas, y principalmente con la finalidad de contar con nuestro propio ambiente de trabajo y de simulación, diseñamos una interfaz gráfica desarrollada mediante la herramienta GUIDE (Graphical User Interface Development Environment) de Matlab versión 6.5. La interfaz aprovecha los recursos que Matlab ofrece para un diseño rápido y sencillo de este tipo de aplicaciones, además de facilitar la liga de los algoritmos de cada parte del proceso de reconocimiento con la interfaz [24].

El software de la interfaz de captura gráfica, es la manera como se realiza y se presenta la captura de la escritura al usuario; es preciso recordar que esta captura es en línea, utilizando un ambiente tipo ventana. Mediante esta interfaz manejamos la información proveniente de la tableta electrónica, para lo cual colocamos un área activa dentro de una interfaz gráfica, dado que la pluma de la tableta electrónica no cuenta con tinta y que el usuario no podría ver sus trazos (escritura), dicha interfaz nos ayuda a resolver esta situación, ya que además de obtener la captura de la escritura, también sirve para proporcionarle al usuario una vista de sus trazos conforme los está realizando, con lo cual se obtiene una escritura más natural.

En nuestra interfaz gráfica de captura tenemos un área activa (área de escritura) correspondiente a una matriz llamada PC de 100x100 ó de 100x600 píxeles, ya sea para un carácter aislado o para una palabra, respectivamente. En cada caso lo que obtenemos es una matriz que maneja 256 tonos de gris (en valores de 0 a 255), inicializada con ceros, lo cual representa en la interfaz al fondo negro (área activa).

Al acercar la pluma al área activa (cerrando el campo electromagnético) la pluma sobrescribe inmediatamente en forma interruptible en la matriz PC valores de cero, esto quiere decir que al mover la punta de la pluma sobre la pantalla sólo movemos el cursor, pero no escribimos nada en la pantalla. Esta acción corresponde a la función “Negro”, con lo que no estamos modificando dicha matriz, y no apreciamos ningún cambio en la imagen, como se muestra en la Figura 2.9.

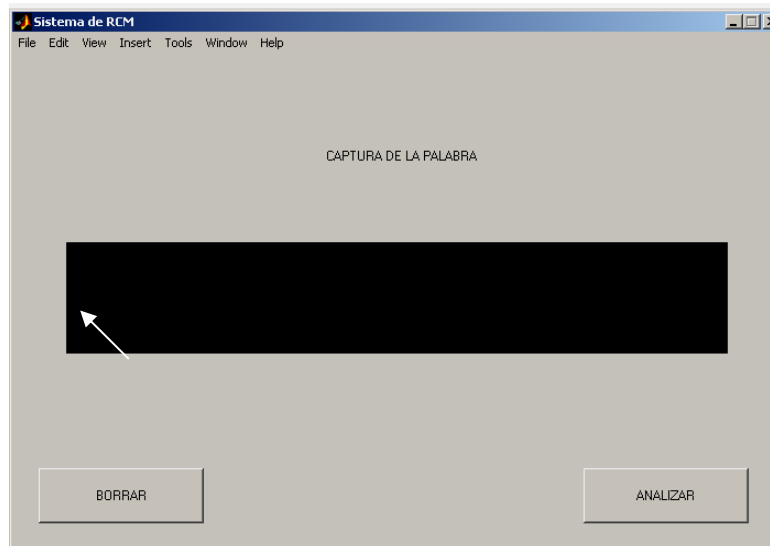


Figura 2.9. Interfaz de captura con la pluma sobre el área activa, sin presionar el botón principal

El proceso anterior se explica en el siguiente fragmento del código del algoritmo de la interfaz de captura.

```
function Negro
global aux_color flat1 PC;
set(id.hLinea,'xData',aux(2,1),'yData',aux(1,2),'Marker','.', 'Color',aux_color);
if aux_color=='r'
    aux = get(id.hEjesImgOrig,'CurrentPoint');
    cadena=round(aux(2,1));
    set(id.hTextK,'String',cadena);
    x=round(aux(2,1));
    y=round(aux(1,2));
    PC(y,x)=0;
    PC(y-1,x)=0;
    PC(y,x+1)=0;
    PC(y-1,x+1)=0;
    PC(y,x-1)=0; %DIBUJA UN PUNTO NEGRO DE 3X3 PÍXELES
    PC(y-1,x-1)=0;
    PC(y+1,x+1)=0;
    PC(y+1,x-1)=0;
    PC(y+1,x)=0;
end
```

Cuando realizamos la interrupción con el botón principal de la pluma (en nuestro caso la punta de la pluma), la acción de sobre-escribir con ceros cambia a escribir en la matriz con 255, para lo cual se llama a la función “Rojo”, que modifica a la matriz PC. En la interfaz, al realizar esta acción, nosotros vemos el trazo en color rojo, para que fuera más notorio el cambio en dicha matriz, como se muestra en la Figura 2.10.



Figura 2.10. Interfaz de captura activa y con el botón principal activado

Para poder escribir, necesitamos realizar una serie de pasos para los eventos de escritura, tales pasos son:

1. Colocar la pluma en el área activa
2. Tener presionado el botón principal de la pluma, lo que llama a la función “Rojo”.

Lo anterior se expresa en el siguiente fragmento del código de la interfaz, que ilustra dicha función “Rojo” llamada para escribir en rojo la escritura capturada.

```
function Prueba2
global aux_color flat1 PC;
if flat1==0
    aux_color='r';
    aux = get(id.hEjesImgOrig, 'CurrentPoint');
    cadena=round(aux(1,2));
    set(id.hTextK, 'String', cadena);
    x=round(aux(2,1));
    y=round(aux(2,2));
    PC(y,x)=255;
    PC(y-1,x)=255;
    PC(y,x+1)=255;
    PC(y-1,x+1)=255;
    PC(y,x-1)=255; %DIBUJA UN PUNTO ROJO DE 3X3 PIXELES
    PC(y-1,x-1)=255;
    PC(y+1,x+1)=255;
    PC(y+1,x-1)=255;
    PC(y+1,x)=255;
    flat1=1;
else
    aux_color='black';
    flat1=0;
end
```

Como podemos observar en el código anterior, cada punto por donde pasa la punta presionada de la pluma de la tableta y los ocho pixeles que lo rodean cambia a rojo (255), con lo que se obtiene un ancho de trazo de tres pixeles, ya que un ancho menor provocaba discontinuidades en el trazo, debido a la lentitud con que Matlab registra el movimiento del cursor.

En caso de error en la escritura, el botón “BORRAR” llama a la función “Borrar”, la cual vuelve a rellenar de ceros la matriz PC y pone nuevamente la pantalla en negro, para poder volver a escribir, lo cual se expresa en el siguiente fragmento de código.

```
function Borrar
global PC
for ii=1:100
    for jj=1:100
        PC(jj,ii)=0;
    end
end
```

De estar satisfecho con la escritura lograda el botón “ANALIZAR” guardará la imagen para su posterior procesamiento mediante la siguiente instrucción.

save PC; %SE GUARDA EL RESULTADO DE LA ESCRITURA EN LA MATRIZ PC

Dicho procesamiento dependerá de si estamos reconociendo un carácter aislado o una palabra, en el primer caso el siguiente proceso será la extracción de características, en el caso de una palabra, el siguiente proceso es la segmentación de la misma.

En caso de que volvamos a pasar la punta de la pluma sobre la escritura realizada se llamará a la función “Negro”, por lo que aunque veamos puntos negros superpuestos sobre nuestra escritura, los valores guardados en la matriz PC no se modificarán.

2.4 Interfaz gráfica del sistema

La interfaz es la parte de un sistema informático que permite que el usuario interactúe con la computadora; mediante esta interfaz el usuario será capaz de introducir los datos necesarios a la computadora, y que ésta a su vez pueda devolverle el resultado deseado, en este caso el reconocimiento del o de los caracteres manuscritos.

El objetivo principal de la interfaz es hacer más amigable el sistema al usuario y en nuestro caso, como diseñadores y desarrolladores del sistema de reconocimiento, para realizar de una manera eficiente nuestro trabajo. Esto se puede ejemplificar con los juegos de video, en los cuales anteriormente un simple punto o una barra podía significar un avión o un proyectil, ahora, con los avances en las interfaces gráficas enfocadas a videojuegos dicho punto o barra se transforma en un avión con detalles sumamente realistas, que vuelve al mismo sistema más atractivo al usuario. En nuestro caso, como hemos mencionado, la interfaz de nuestro sistema se realizó como una forma de presentación que aumenta las perspectivas de uso del sistema, haciendo que sea sencillo para el operador hacer uso de él.

Al realizar nuestra captura en línea y poder visualizar cómo realizamos nuestra escritura en cada instante, necesitamos que nuestros trazos sean más lentos de lo habitual para no perder información, ya que el flujo de información es constante, por lo que si realizamos la escritura a una velocidad mayor no nos permitiría observar el modo en que se forman los trazos.

Esta interfaz está adaptada con el principal propósito de reconocer escritura manuscrita, tanto de caracteres aislados, como de palabras cortas; en el segundo caso también tiene como propósito mostrarnos un paso anterior al reconocimiento, el cual es la segmentación de la palabra en la unidad mínima a reconocer, que es un carácter.

El diseño de la interfaz se realizó de una manera sencilla y accesible a prácticamente cualquier usuario, mediante el uso de botones gráficos, ya que es mucho más práctico y sencillo que aprenderse los programas que habría que correr para obtener el reconocimiento, asimismo, presentamos de una forma más agradable nuestros resultados mediante imágenes.

En el diseño también se pensó en no saturar al usuario con información de resultados de procesos intermedios al reconocimiento, como serían algunos procesos de la captura, adecuamiento del carácter, extracción de características y comparación, por lo cual solo mostramos la segmentación

y el despliegue del reconocimiento, el cual será mediante el despliegue del carácter reconocido en un estilo prediseñado, así como en el caso de que no se logre el reconocimiento esto se representará mediante un signo de interrogación “?”.

Como se ejemplifica en la Figura 2.11, el representar nuestros resultados en esta forma es una gran ventaja, ya que no es necesario tener que interpretar un resultado numérico o alguna gráfica, lo cual vuelve al sistema más práctico, utilizable y fácil de entender.

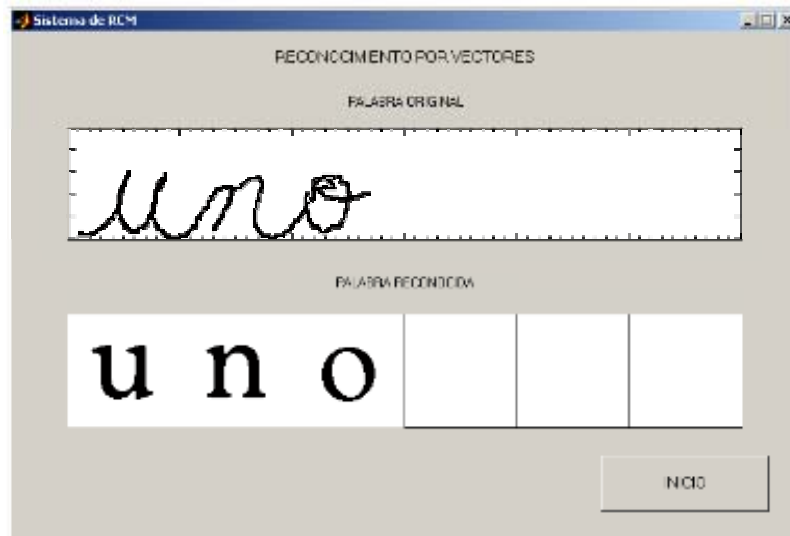


Figura 2.11. Pantalla de reconocimiento que forma parte de la interfaz del sistema

Habiendo capturado la escritura mediante sus propiedades estáticas (imagen) y habiendo adentrado al lector en las interfaces gráficas, podemos continuar con el siguiente bloque del sistema, el cual se refiere a la adecuación de la escritura capturada.

Capítulo 3

Pre-procesamiento y Segmentación

Una vez revisada la teoría necesaria para entender todos los procesos requeridos en un sistema de reconocimiento de caracteres, y presentado el funcionamiento de la etapa de captura, explicaremos nuestra implementación del sistema de reconocimiento que cubra los objetivos que nos hemos propuesto alcanzar. Deseamos hacer el reconocimiento de caracteres aislados, los cuales son las letras a, e, i, o, u, c, d, h, n, r, s, t, v. Por otro lado, haremos el reconocimiento de palabras, las cuales son los nombres de los números cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve. Tanto los caracteres como las palabras de este diccionario formado por las palabras y letras mencionadas, son escritos de forma cursiva, esto es, que cada letra tiene un trazo que la une con la siguiente (enlace).

En el diagrama de bloques que presentamos en la Figura 3.1, se han resaltado los bloques de adecuación, segmentación y transformación, los cuales se engloban en nuestro sistema dentro del bloque de pre-procesamiento de la escritura, aunque la segmentación sólo se aplique a las palabras, como lo explicaremos en la sección 3.2.

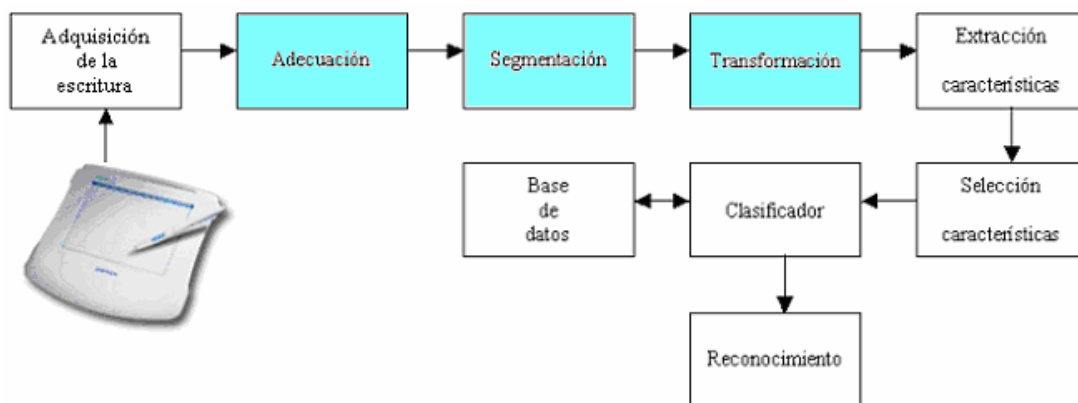


Figura 3.1. Ubicación del bloque de pre-procesamiento en el diagrama general del sistema de RCM

En el Capítulo 2 hemos revisado la metodología de captura de nuestro sistema, la cual es básicamente la misma para palabras y letras. En este capítulo revisaremos la estructura del bloque de pre-procesamiento que desarrollamos, la cual considera un conjunto de algoritmos de normalización

para caracteres, mientras que para el pre-procesamiento de palabras se involucra también el desarrollo y selección de estrategias de segmentación.

De acuerdo a lo anterior, podemos distinguir entre los procesos de reconocimiento de caracteres aislados y de palabras, los cuales describiremos en las secciones siguientes.

3.1 Caracteres

El primer paso en el reconocimiento de caracteres aislados, como se ilustra en la Figura 3.1, es su captura mediante la interfaz que diseñamos para tal efecto, lo cual se muestra en la Figura 3.2.

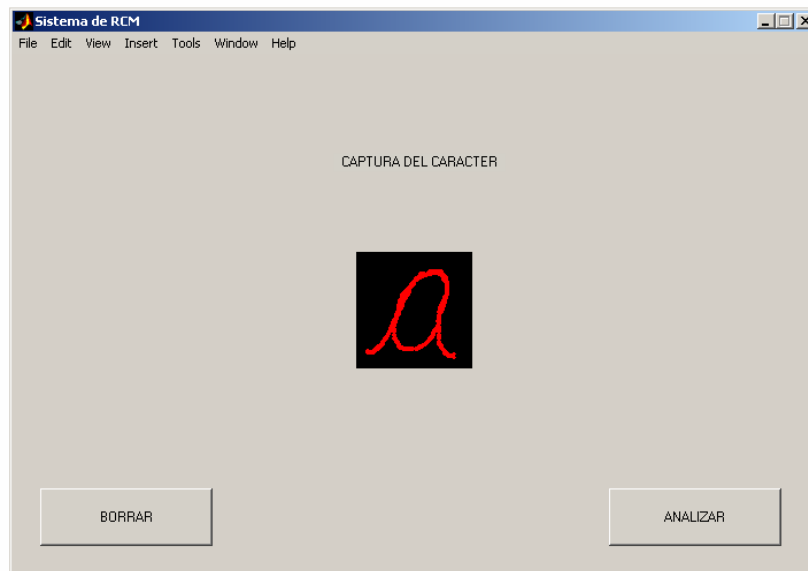


Figura 3.2. Interfaz de captura del carácter

El resultado del proceso de captura es la obtención de una matriz PC de 100x100 píxeles, en la cual guardamos los puntos que forman al carácter, representando con 255 el color rojo del trazo y con 0 el área del fondo donde se encuentra escrito el carácter.

De esta manera se guardan los puntos por donde pasa el cursor y los ocho píxeles que rodean a cada uno de ellos, como se puede ver en la Figura 3.3, donde se muestra un fragmento de la imagen binaria capturada, representada por 0 y 255, con los valores de la matriz PC invertidos respecto a como se mencionó en la sección 2.3 del Capítulo 2; lo anterior con el fin de facilitar algunos procesos posteriores de extracción de características y segmentación en el caso de las palabras.

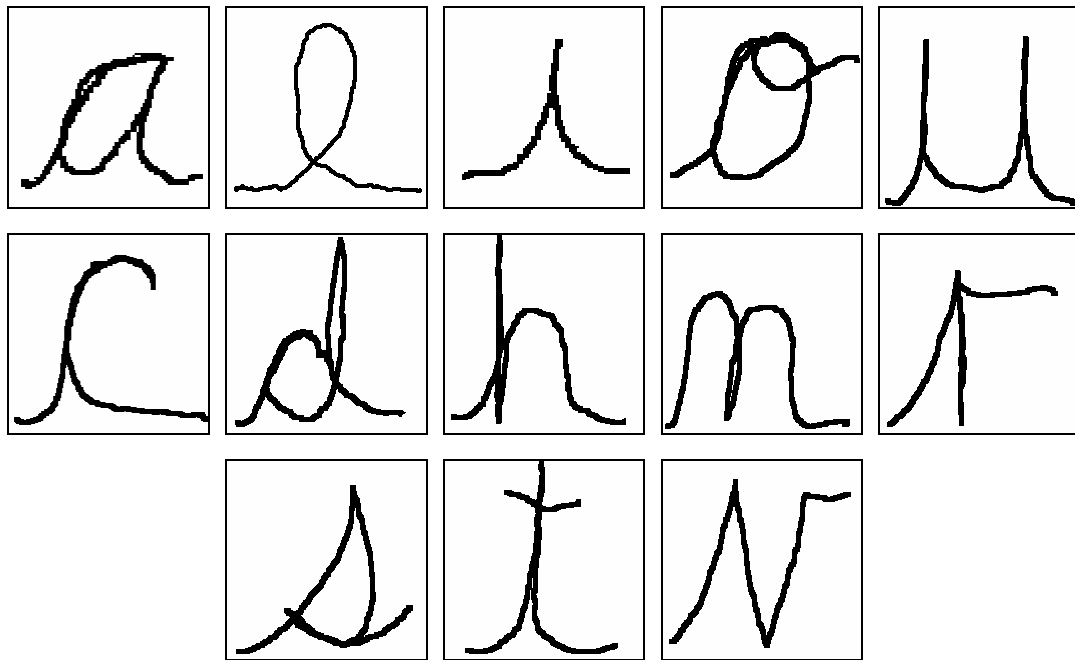


Figura 3.4. Caracteres prototipo

La forma de escritura de estos caracteres se definió en base a la forma en la cual éstos se escriben dentro de las palabras del diccionario elegido, es decir, tal como se muestra en la Figura 3.5. De esta manera es como se obtiene la forma de los caracteres mostrados en la Figura 3.4.

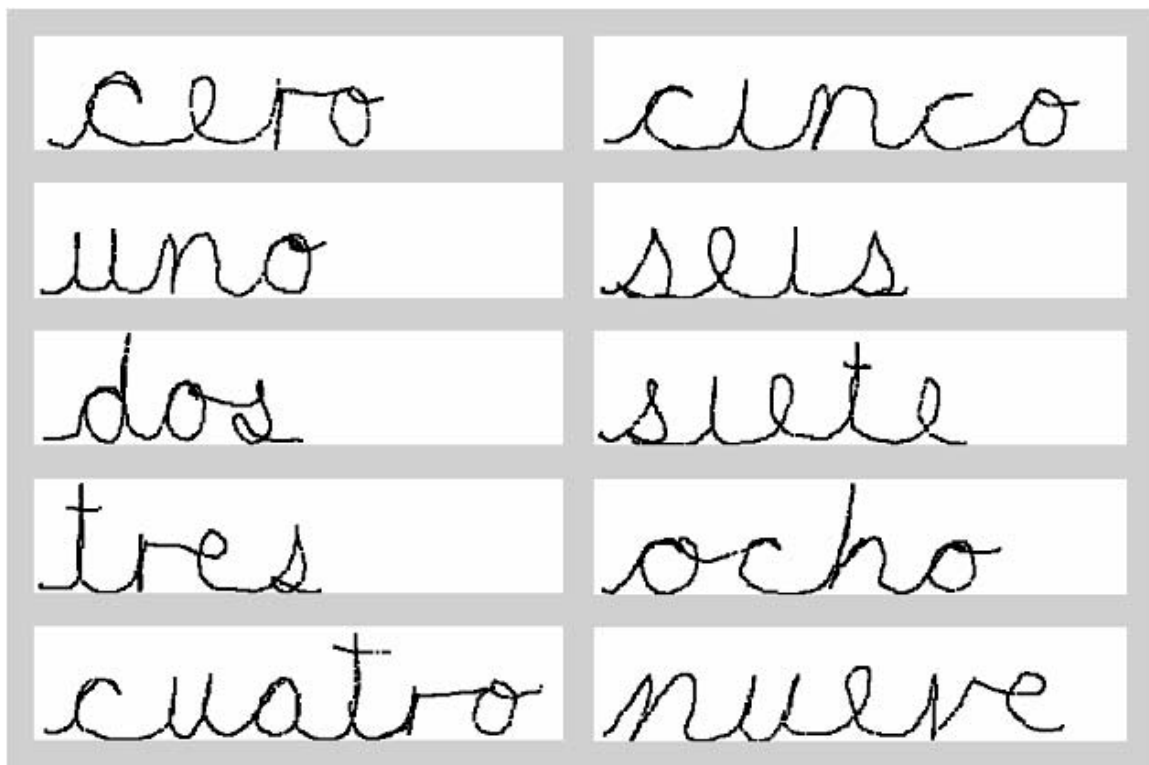


Figura 3.5. Palabras prototipo

Estos caracteres prototipo los usaremos para la extracción de características y el reconocimiento en los métodos de diferencias, correlación y trayectorias que describiremos en los Capítulos 4 y 5, mientras que para el método de vectores, usaremos sus vectores característicos compuestos por cuatro elementos, lo cual será descrito en el Capítulo 5, y para la obtención de los cuales también se desarrolló una base de datos.

3.2 Palabras

Para realizar el pre-procesamiento de las palabras requerimos de más procesos que los utilizados para los caracteres aislados. Del mismo modo que en los caracteres, las palabras son capturadas en la interfaz, esta vez en un área activa de 600x100 pixeles, como se muestra a continuación.

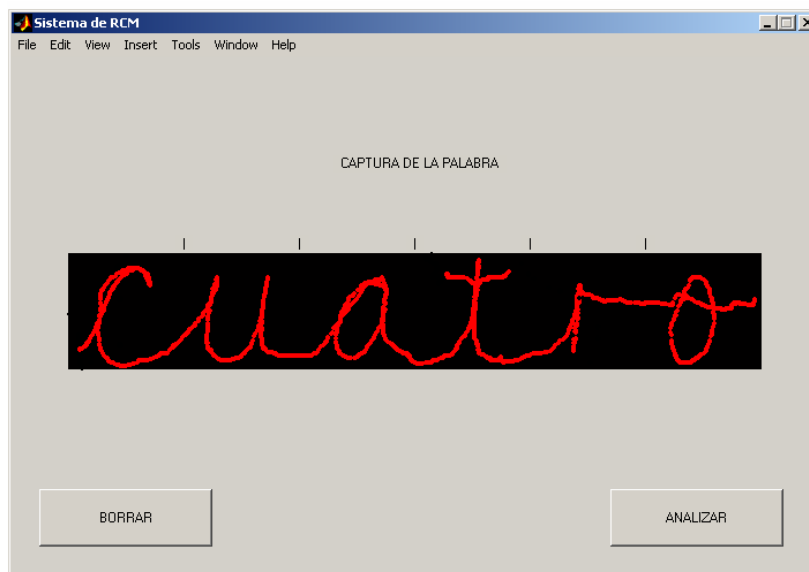


Figura 3.6. Interfaz de captura de palabras

El pre-procesamiento de las palabras capturadas consiste en la segmentación, cuyo objetivo es obtener los puntos de corte en la palabra que permitan dividir a la matriz que la contiene en tantas matrices como letras hay en la palabra, donde cada matriz contiene un carácter perteneciente a la palabra. Para realizar la segmentación de las palabras desarrollamos tres métodos: segmentación gráfica, segmentación por cortes y segmentación por localización de extremos superiores. En [16] se hace mención a un método de segmentación, cuya idea básica es escribir los caracteres en regiones predefinidas (boxes), con lo cual el escritor realiza la mayor parte de la segmentación. Nuestro algoritmo de segmentación gráfica es una adaptación de los métodos mencionados en dicha referencia bibliográfica. Los dos algoritmos restantes se basan en análisis geométricos de las imágenes de las palabras; estos algoritmos son nuestra propuesta original, y están enfocados al tipo de escritura que analizamos en este estudio.

Estos tres métodos funcionan bajo condiciones de escritura diferentes, y mediante el análisis de su efectividad y desempeño al aplicarlos al tipo de escritura que estamos estudiando, se eligió,

como lo describiremos en la sección 3.2.4, el método más eficiente para nuestros propósitos. En las siguientes secciones explicaremos los algoritmos de estos tres métodos.

3.2.1 Segmentación gráfica

En la segmentación gráfica tenemos que al escribir cada carácter que forma a la palabra en un espacio de 100x100 pixeles, podemos considerar la colocación de un punto de segmentación cada cien pixeles respecto al eje horizontal, de manera que si se escribe cada una de las letras respetando el espaciamiento de 100 pixeles entre ellas, la segmentación no producirá ningún error, como se muestra a continuación con la palabra “cuatro”:

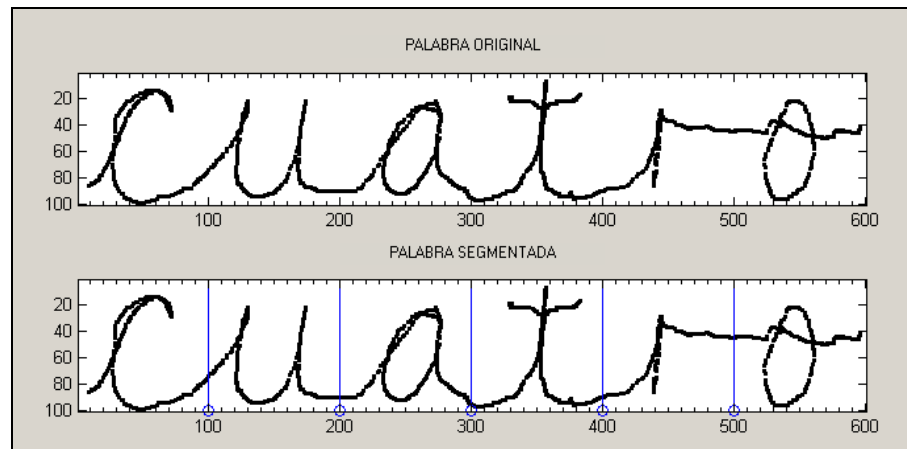


Figura 3.7. Segmentación gráfica de la palabra cuatro

A fin de hacer más claro este proceso, en la Figura 3.8 se muestra el diagrama de bloques de este algoritmo de segmentación, mencionando los dos pasos fundamentales que intervienen en dicho proceso.

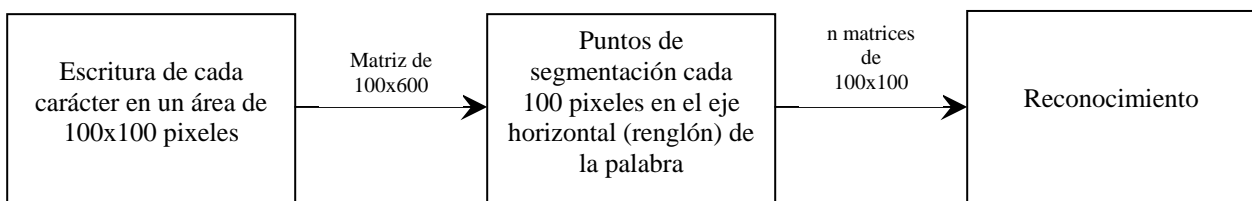


Figura 3.8. Diagrama de bloques del algoritmo de segmentación gráfica

Este es un método sencillo y eficiente de segmentación, que realiza una colocación eficiente de los puntos de corte de la palabra, a condición de escribir exactamente cada letra en cada espacio de 100x100 pixeles. Como podemos apreciar, sin embargo, este método requiere que todas las letras ocupen prácticamente todo el alto disponible y que su separación sea constante, haciéndolo un tanto antinatural al escribir.

3.2.2 Segmentación por cortes

La palabra guardada en una matriz de 100 x 600 elementos está representada por 0 para el color negro de la escritura, y 255 para el fondo blanco de la misma. En primer lugar, de la matriz A que contiene a la palabra, se obtiene mediante la operación sum de Matlab [21] un vector de 1x600, donde cada elemento de dicho vector representa la suma de los elementos de cada una de las columnas de la matriz A. En las columnas vacías esta suma será de 25500, y donde la suma sea menor que 25500 habrá píxeles en blanco, y por lo tanto, escritura; esto nos sirve para determinar el inicio de la palabra, ya que donde encontremos la primera columna cuya suma sea menor a 25500 se ubica el inicio de la palabra.

Para encontrar el final de la palabra, buscamos desde la mitad de la primera letra (70 píxeles, puesto que nuestra palabra más corta es de tres letras) la columna para la cual sus tres columnas siguientes sumen 25500 y ella sea menor a 25500, lo cual nos garantiza que aunque haya algún hueco en la escritura, la ubicación del fin será correcta.

Después obtenemos desde el principio hasta el fin la altura máxima en cada columna, con lo cual obtenemos la envolvente superior de la palabra, y cada altura se guarda en un vector d. De esta envolvente vamos leyendo cada altura, de esta manera, si el píxel i tiene una altura determinada y el píxel i+1 varía su altura en 2 o 3 píxeles, se le asigna a ambos un 1 en el vector d. Esto ocurre donde hay enlaces en la palabra, ya que donde hay una letra las alturas van ascendiendo y descendiendo más de 2 o 3 píxeles, por lo que en todo lo largo de la letra se asigna una secuencia de ceros. De este modo tenemos en el vector d secuencias de unos y ceros, donde las secuencias de unos son enlaces y las de ceros letras. Para realizar los cortes lo que nos interesa son las secuencias de unos, por lo que se obtiene la longitud de cada una de ellas y se obtiene el punto medio de cada una.

Tenemos ahora que a partir de la primera se toman de dos en dos las secuencias y se obtiene la distancia entre sus puntos medios. El primer punto es el inicio de la letra y el segundo es el final. Si la distancia entre ellos es mayor a 100, por ejemplo 103, el principio y el final de la letra se recorren 3 píxeles, para el siguiente carácter su inicio es el final del anterior y se repite el proceso anterior hasta terminar con todas las secuencias. De esta manera obtenemos los puntos preliminares de segmentación, mostrados en la Figura 3.9.

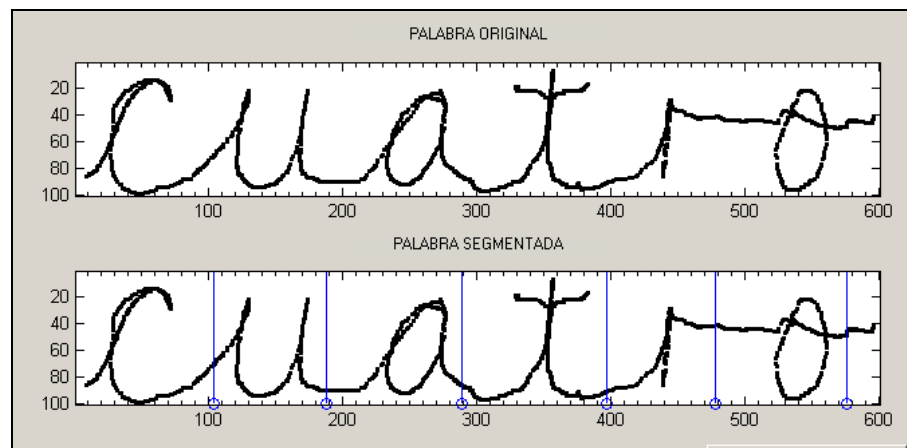


Figura 3.9. Segmentación por cortes de la palabra cuatro

Al colocar los puntos de segmentación podemos separar la matriz original en seis matrices de diferentes tamaños, cada una de las cuales contiene a cada letra de la palabra cuatro ya mostrada, las cuales se muestran como imágenes en la Figura 3.10.

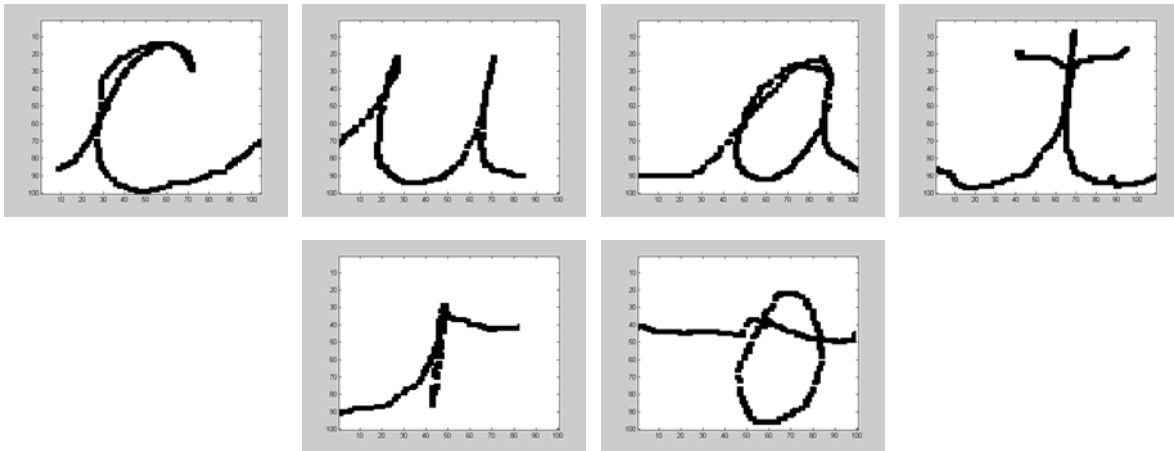


Figura 3.10. Caracteres segmentados. Primer paso

Como podemos observar, en algunas ocasiones, al escribir las palabras dibujamos enlaces muy largos entre una letra y otra, lo que hace que el cuerpo de la letra, es decir, el dibujo de la letra sin enlaces, no esté centrado y dificulte su posterior reconocimiento, por lo que encontramos ahora la longitud de las secuencias de ceros, que representan el cuerpo de la letra, y de nuevo encontramos el punto medio de la secuencia; este punto medio nos sirve para centrar el carácter y que no quede muy orillado. Antes de centrar debemos eliminar los sobrantes de los enlaces, quitando diez píxeles de un lado y diez del otro. Ahora, a partir del punto medio del cuerpo, distribuimos la mitad de los píxeles de un lado y la mitad del otro, de esta manera el carácter queda centrado en una matriz de 100x100 píxeles, la cual puede utilizarse ahora en los siguientes pasos de los procesos de reconocimiento. El resultado de la aplicación de este proceso da como resultado las imágenes mostradas en la Figura 3.11, correspondientes a las letras de la palabra cuatro de la Figura 3.10.

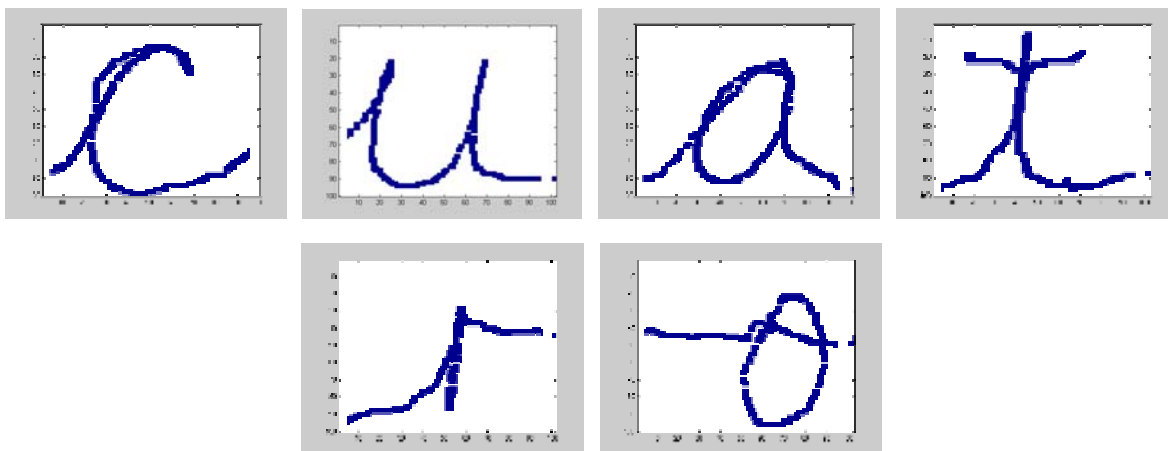


Figura 3.11. Caracteres segmentados. Segundo paso

Con el fin de mostrar claramente los pasos que conforman este algoritmo de segmentación por cortes, en la Figura 3.12 tenemos el diagrama a bloques de este proceso, en el cual se resaltan sus elementos principales.

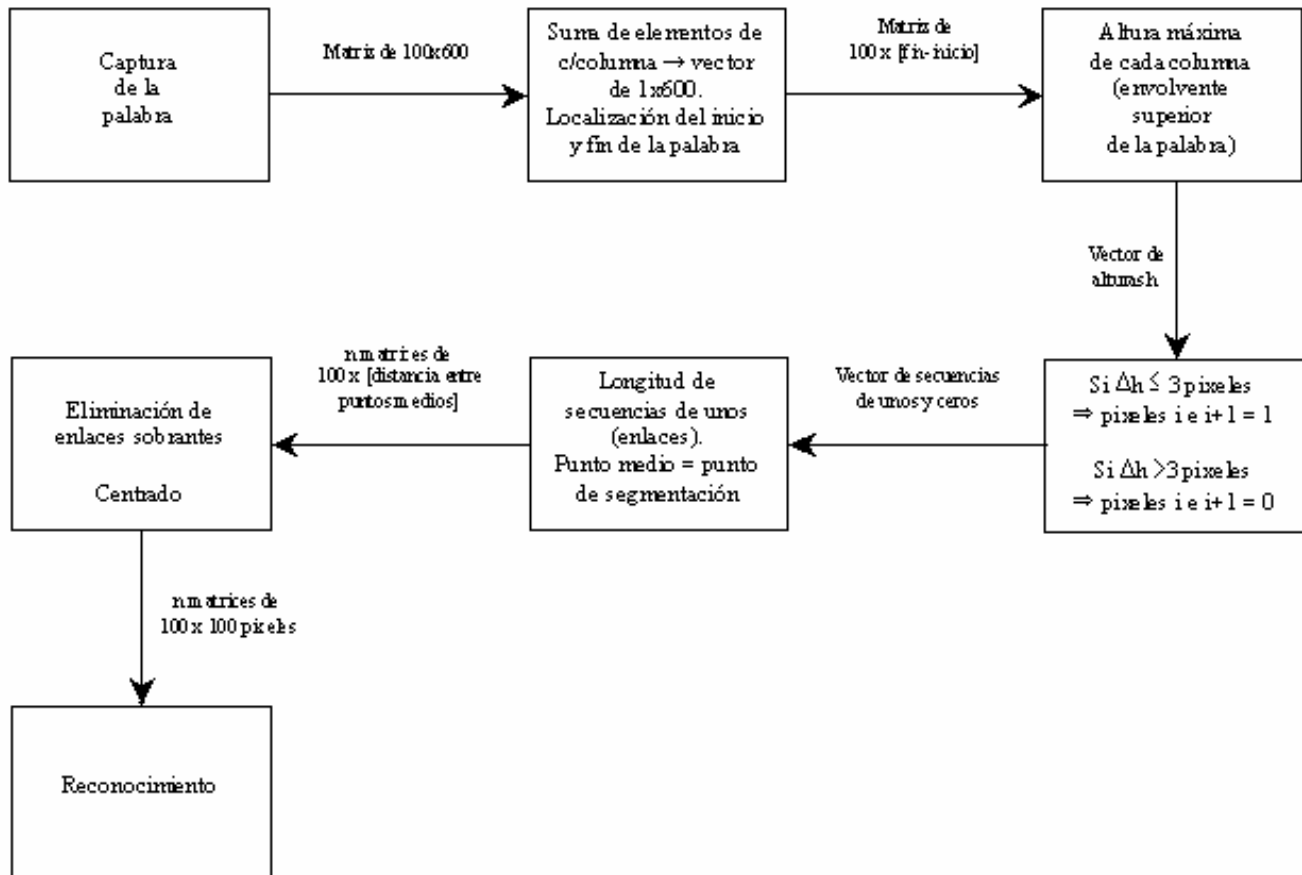


Figura 3.12. Diagrama de bloques del algoritmo de segmentación por cortes

Con las letras *i* y *u*, un problema al segmentar es que el cuerpo de la *i* es muy pequeño, y que dentro de la *u* tenemos una secuencia de unos; por esto se establece una distancia mínima entre el principio y el fin del carácter de setenta píxeles, para evitar cortar demasiado la *i* y cortar en dos la *u*.

A pesar de que este método permite cierta libertad en el espaciado de los caracteres, si éstos se juntan demasiado, se rebasan las distancias establecidas para obtener la segmentación y el centrado, por lo que este método puede generar errores en la segmentación, sobre-segmentando o colocando menos puntos de los necesarios, como se muestra en la Figura 3.12.

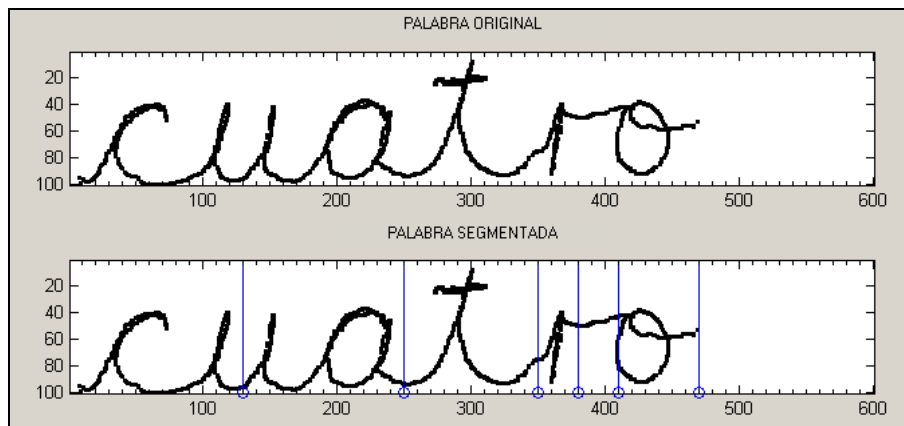


Figura 3.12. Palabra mal segmentada

3.2.3 Segmentación por localización de extremos superiores

Este método se basa, como su nombre lo indica, en la localización de los extremos superiores de la palabra completa, obteniendo sus distancias, y mediante promedios, obtener los puntos de corte en la palabra. En la Figura 3.13 se muestra la captura de la palabra “cuatro”, en la cual podemos apreciar una escritura más natural, dando a cada letra un tamaño y un espaciado más adecuado, sobre la cual se aplicará el algoritmo de segmentación que explicaremos a continuación.

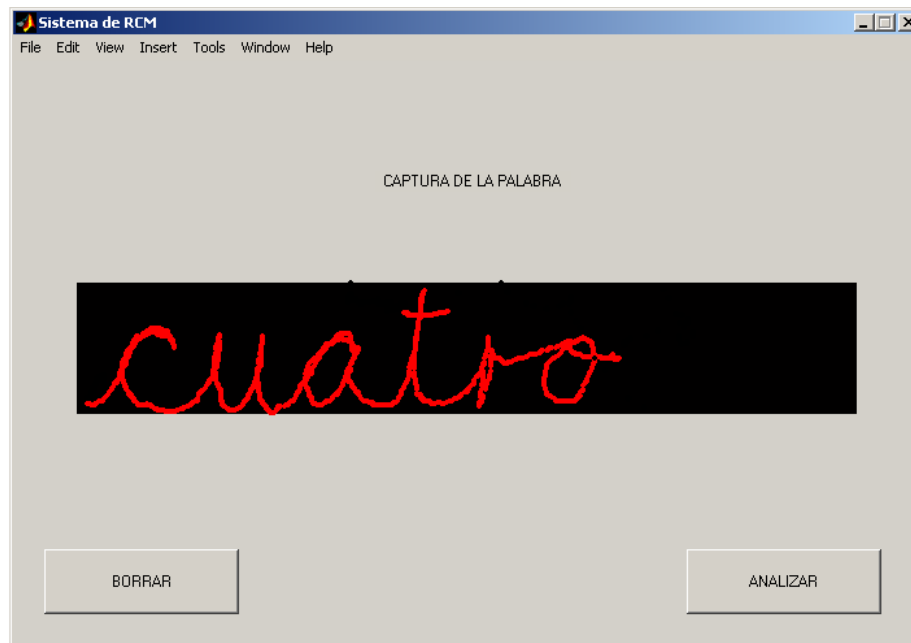


Figura 3.13. Captura de la palabra cuatro

En primer lugar, tenemos la matriz de 600x100 en la cual está contenida la palabra, de la cual se obtiene para cada punto activo su altura respecto al eje x, y se guarda en otra matriz; a los puntos que no están activos se les asigna el valor de cero. Tenemos ahora una matriz en donde en algunas de

sus columnas hay más de un valor, entonces obtenemos el valor máximo de cada columna, con lo cual obtenemos la envolvente de la palabra. Después, aplicamos un criterio para cada punto de dicha envolvente: si su altura es mayor de 45, se guarda en un arreglo, y si su altura es menor a 45 se descarta. Ahora, vamos leyendo el arreglo, y se obtienen los extremos más altos en toda la palabra. En cada carácter vamos a encontrar un solo punto más alto, excepto en letras como la u, donde encontramos dos extremos, entonces, establecemos una distancia mínima de separación de dichos extremos, y puesto que los de la u están muy juntos, se promedian las posiciones de estos dos extremos y se obtiene un solo punto a la mitad de la distancia entre ambos.

Tenemos ahora un vector donde se encuentran los puntos más altos de la palabra: tomamos esos puntos de dos en dos, se promedian y el punto a la mitad corresponde a la posición del punto de segmentación entre dos letras. Puesto que los caracteres están un poco inclinados, dicho punto se recorre unos cuantos píxeles a la derecha para que esté correctamente ubicado. En la Figura 3.14 se muestran dichos puntos de segmentación en la palabra “cuatro” de la Figura 3.13.

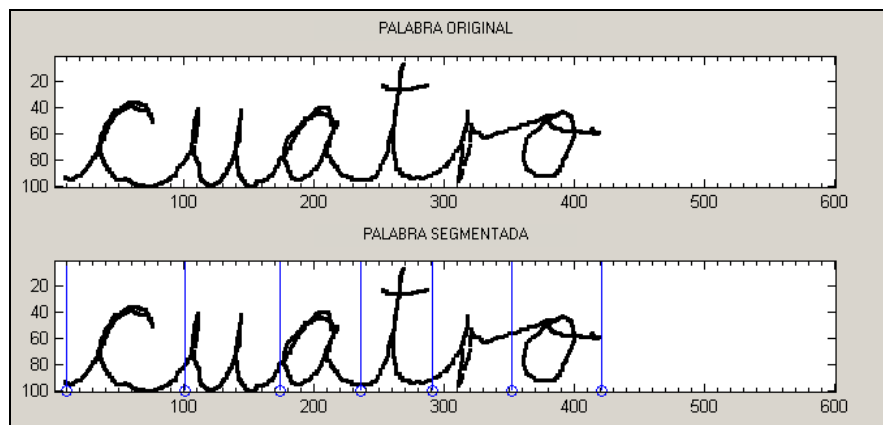


Figura 3.14. Segmentación por localización de extremos superiores

Como resultado de esta segmentación, tenemos los siguientes caracteres, contenidos en seis matrices del mismo alto y distinto ancho.

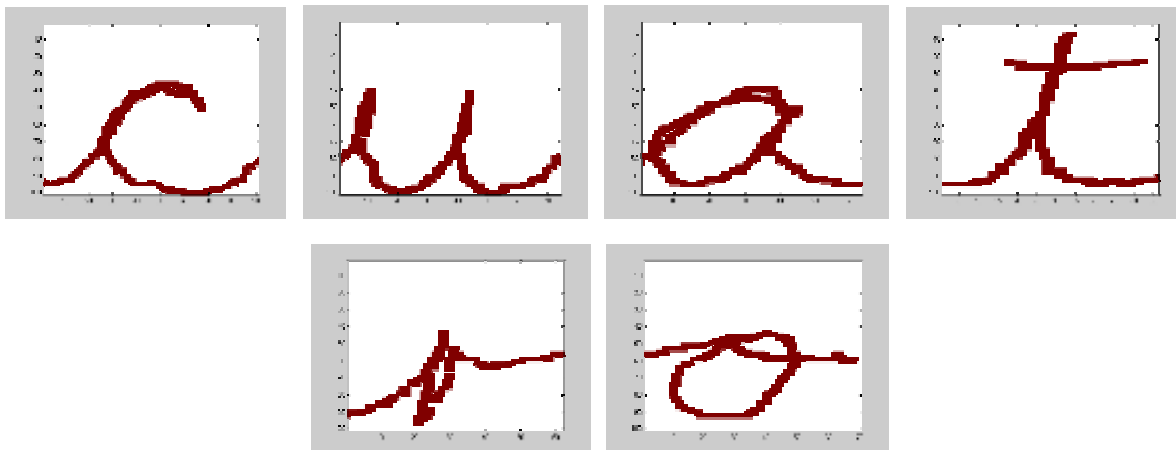


Figura 3.15. Caracteres segmentados

Como podemos observar, estos caracteres segmentados no se encuentran centrados, y tienen sobrantes de los enlaces que los interconectaban, por lo que se requiere un proceso de centrado y eliminación de enlaces sobrantes. Para centrar a cada carácter, se mide el ancho de cada uno de ellos, y su diferencia con el ancho de la matriz donde debe ser contenido para entrar al siguiente bloque de procesamiento, que es de 100x100, se divide entre dos, y la cantidad resultante corresponde a la columna a partir de la cual se coloca el carácter segmentado. Puesto que para esta segmentación se permite escribir libremente, generalmente los caracteres no están demasiado separados, por lo que en el único caso en el que se eliminan sobrantes de enlaces es cuando el ancho del carácter segmentado rebasa los 100 píxeles, en ese caso se elimina a ambos lados una parte de los enlaces y se centra en la matriz de 100x100 correspondiente. En las imágenes de la Figura 3.15 se muestran los caracteres segmentados anteriores que han sido centrados en matrices de 100x100 píxeles.

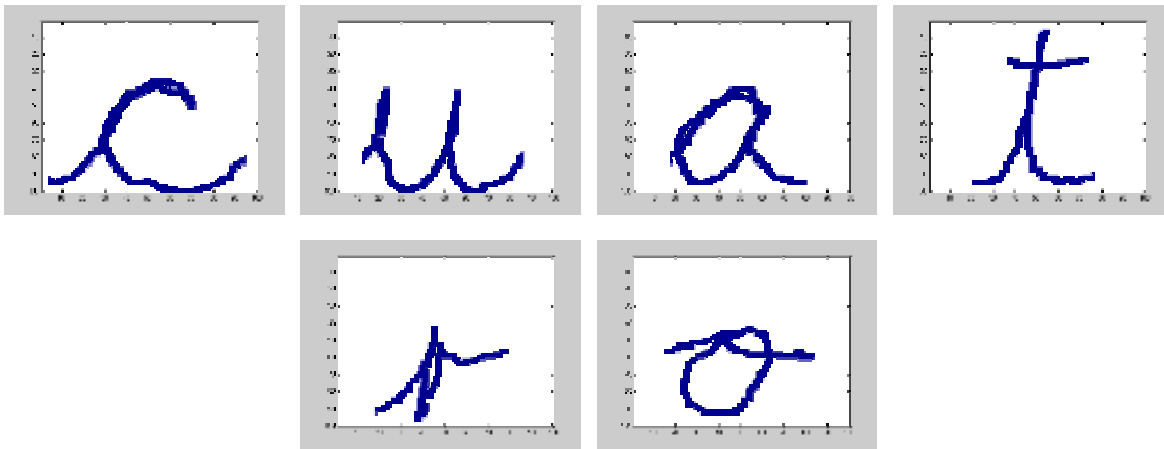


Figura 3.15. Caracteres centrados

Con el fin de hacer más claro el procedimiento que se lleva a cabo en la segmentación por localización de extremos superiores, en la Figura 3.16 tenemos el diagrama de bloques del algoritmo, donde se destacan los pasos más importantes de éste.

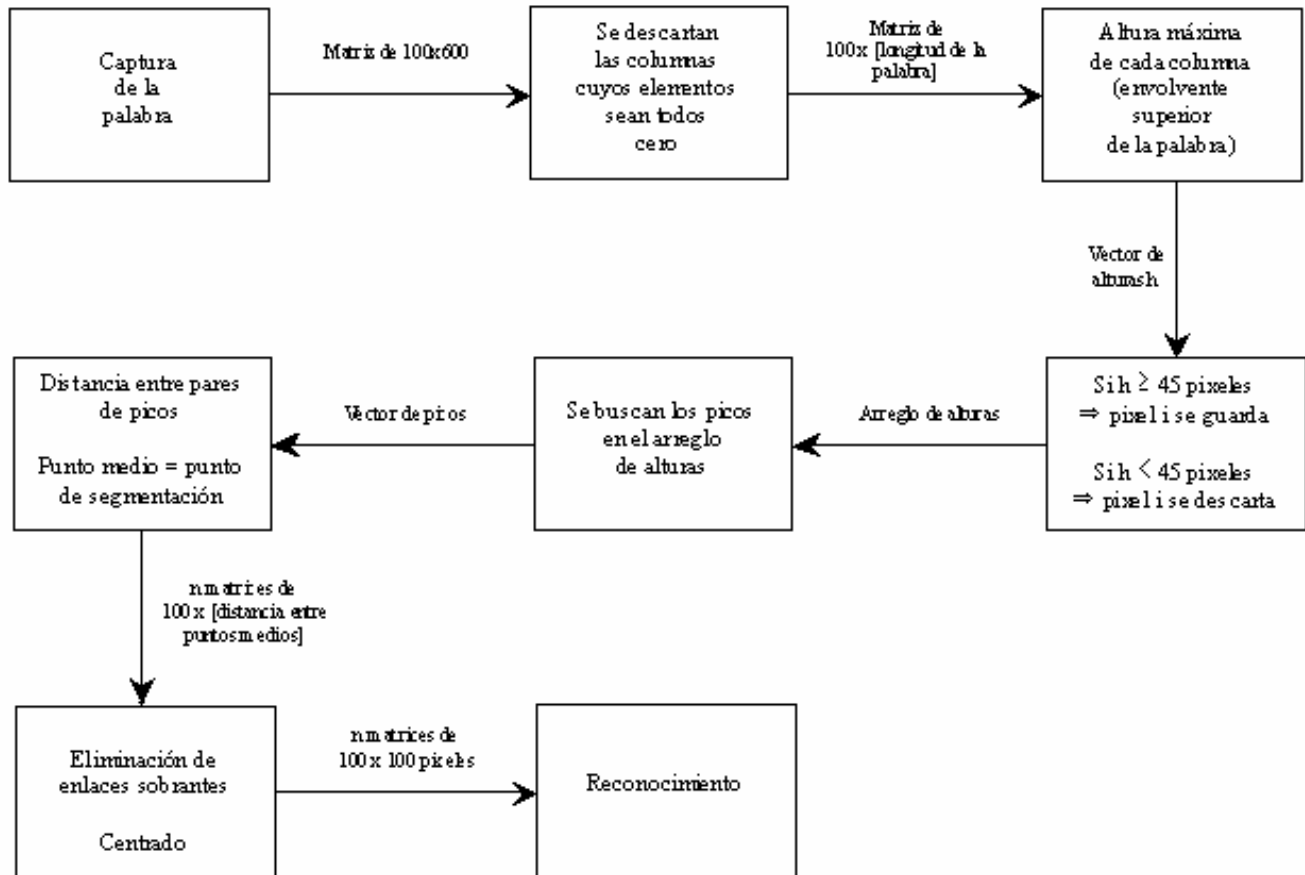


Figura 3.16. Diagrama de bloques del algoritmo de segmentación por localización de extremos superiores

Como podemos observar en la Figura 3.14, este método de segmentación permite trazar una escritura más libre y natural, ya que no solamente se respeta el estilo de escritura del autor, sino que se considera una variación importante en la escritura cursiva manuscrita, tal como su inclinación natural.

3.2.4 Selección del algoritmo de segmentación

Para seleccionar el algoritmo de segmentación que mejor se adapte al tipo de escritura que estamos analizando, ya que hemos explicado en la sección anterior el funcionamiento de cada uno de los tres algoritmos que desarrollamos, haremos una comparación de las ventajas y desventajas de cada uno de dichos métodos.

A fin de hacer la evaluación lo más clara posible, tenemos en la Tabla 3.1 un contraste directo de las tres estrategias de segmentación que implementamos.

MÉTODO DE SEGMENTACIÓN	SENCILLEZ DEL ALGORITMO	LIBERTAD EN LA ESCRITURA	EFICIENCIA	DESVENTAJAS
Gráfica	Alta	Baja	Alta	Debe escribirse cada letra en cada espacio marcado, ocupando toda el área de escritura disponible, con separación y altura constantes.
Por cortes	Media	Media	Media	Caracteres mal espaciados producen exceso o defecto de puntos de segmentación, se debe ocupar todo el alto disponible, altura constante.
Por localización de extremos superiores	Media	Alta	Alta	Error al encimar caracteres.

Tabla 3.1. Comparación del desempeño de los tres algoritmos de segmentación

Para determinar las ventajas y desventajas mostradas en la Tabla 3.1, por cada algoritmo se realizaron pruebas de segmentación con un gran número de pruebas por cada una de las diez palabras del diccionario, con lo que se obtuvieron las características resumidas en dicha tabla.

Para evaluar la sencillez de los algoritmos, partimos del hecho de que el algoritmo de la segmentación gráfica tiene muy pocas líneas de código, sin ninguna estructura compleja de programación, por lo que los dos métodos restantes, aunque tampoco poseen un alto grado de complejidad algorítmica, sí tienen una cantidad de líneas de código mucho mayor, además de usar ciertos recursos de programación, tales como ciclos y condicionales.

Los niveles de libertad y naturalidad en la escritura que admiten los métodos de segmentación, dependen de las características de la escritura en las cuales se basan dichos métodos para realizar la segmentación, puesto que en todos ellos la geometría, posición y tamaño de las letras es fundamental para la correcta separación de una palabra en los caracteres que la forman. Este aspecto del desempeño de los algoritmos es tan importante, que como podemos ver en la Tabla 3.1, aunque la segmentación gráfica tiene, al igual que la segmentación por localización de extremos superiores, dos aspectos con una alta tasa de desempeño, la desventaja de que la forma de la escritura esté totalmente relacionada con la efectividad de dicho algoritmo, hace que en ese aspecto la segmentación por localización de extremos superiores sea superior en eficiencia.

En el caso de estos tres métodos no se evaluó su eficiencia en porcentajes, sino en apreciaciones como alta, media o baja, ya que no se puede comparar cuantitativamente las tasas de eficiencia, puesto que las características de la escritura que se consideran varían notablemente de un

método a otro. En este sentido, los algoritmos que mostraron una alta eficiencia fueron la segmentación gráfica y por localización de extremos superiores, sin embargo, como vemos en la columna de desventajas y como lo mencionamos en la sección 3.2.1, la eficiencia de la segmentación gráfica se basa principalmente en la correcta ubicación, espaciamiento y tamaño que el autor haga de la escritura de las palabras. Esto hace difícil asegurar que la tasa de eficiencia de dicha segmentación sea siempre alta, puesto que depende de un factor externo que nuestro sistema de reconocimiento por sí mismo no puede controlar o predecir.

Por las razones explicadas en los párrafos anteriores, concluimos que el método de segmentación por localización de extremos superiores es un algoritmo que ofrece las ventajas que necesitamos para su aplicación en el tipo de escritura cursiva manuscrita que deseamos reconocer, por lo que este es el algoritmo que finalmente se incorporó al sistema que proponemos, ya que al tener un funcionamiento adecuado contribuye a mejorar la eficiencia de los procesos de reconocimiento posteriores.

Debemos resaltar el hecho de que este método de segmentación, funciona bajo la consideración de que las letras que forman a las palabras del diccionario, son escritas dentro del área activa de un tamaño tal, que las letras altas como la t y la d deben ocupar en lo posible todo el alto disponible en dicha área. De esta manera se determina también el tamaño de las letras bajas, como la a y la e, aproximadamente de la mitad de la altura de las letras altas. Otra consideración importante es que las letras deben ser escritas con una separación tal que se permita una correcta segmentación, a fin de evitar que dos letras demasiado juntas o demasiado separadas impidan encontrar el punto real de segmentación.

Finalmente, la palabra, así como las letras, no deben estar demasiado inclinadas ni encimadas, ya que de igual manera se producirán errores al segmentar, y por lo tanto al reconocer. La inclinación adecuada es aquella que se da naturalmente al escribir caracteres cursivos, respetando el renglón indicado en el área activa. Esto es porque en el sistema no hemos considerado un algoritmo de corrección de inclinación, ya que al realizar un algoritmo para tal fin no obtuvimos resultados aceptables que nos ayudaran a corregir la inclinación de palabras y letras.

Lo que obtenemos entonces de este proceso de segmentación por localización de extremos superiores es un número de matrices igual al número de las letras de la palabra, cada una de ellas de 100x100 pixeles, conteniendo a cada carácter centrado y sin enlaces sobrantes, y un contador donde se almacena el número de cortes hecho a la palabra, lo que sirve para realizar únicamente el número de iteraciones necesarias en los siguientes procesos.

Una discusión importante que realizamos acerca de estos métodos de segmentación que se basan en la densidad de pixeles y otras características geométricas, es la razón por la cual elegimos utilizar estrategias geométricas y no otras. En la bibliografía, existen muchos artículos sobre el reconocimiento de caracteres manuscritos cursivos [15], [16], [20], [23], [27], [33] a [57], sin embargo, la cantidad de ellos dedicados al tema de la segmentación constituyen un pequeño porcentaje del total. En aquellos artículos donde se analiza la eficiencia del método de reconocimiento en general, no se hace un análisis de la eficiencia del algoritmo de segmentación en particular, por lo que en general es difícil establecer puntos de comparación en la eficiencia de estos métodos. En otros artículos [18], [19], [25], [26], [58], [59], [60], [61], se analiza una serie de estrategias orientadas a la segmentación de palabras, sin embargo la estrategia a utilizar y su

eficiencia depende enormemente del objetivo del reconocimiento, el tipo de escritura, y los recursos, tanto computacionales, de tiempo y esfuerzo.

En la búsqueda por una estrategia eficaz de segmentación, encontramos un trabajo de tesis con propósitos similares a los nuestros en ciertos aspectos. Se trata de una tesis profesional [26] en la que se realiza el reconocimiento de telegramas escritos por el General Porfirio Díaz, de manera que su contenido pueda digitalizarse y subirse a la red. Este trabajo llama la atención para esta parte del proceso porque la evaluación de los resultados del reconocimiento está fuertemente enfocada hacia la eficiencia de los algoritmos de segmentación. El tipo de letra con la que el General escribe es manuscrita, además de que no sigue renglones ni tiene limitaciones en la escritura determinadas por el proceso de reconocimiento, de manera que en este sentido podemos hallar ciertas similitudes de utilidad respecto a nuestro trabajo.

Las estrategias de segmentación empleadas en dicho trabajo utilizan densidades verticales de píxeles. Éste es un método heurístico que trabaja a nivel pixel con las matrices que contienen a las imágenes de las letras, lo cual sigue la tendencia que manejamos en nuestro trabajo en cuanto a sencillez en los algoritmos y el análisis sobre las matrices de las imágenes.

La evaluación de estos algoritmos de segmentación se lleva a cabo verificando visualmente que los puntos de segmentación obtenidos sean correctos tanto en número como en ubicación. Por ejemplo, esto abre la posibilidad de que el número de puntos detectados sea correcto pero la ubicación no sea adecuada. Por lo tanto, la evaluación de los métodos toma como puntos correctos solo a aquellos que cumplen con todos estos requisitos.

Dadas las condiciones anteriores, decidimos que un algoritmo sencillo y eficiente lo constituye la estrategia por densidad de píxeles, en particular el método propuesto, segmentación por localización de extremos superiores. Este método es muy utilizado en diversos artículos sobre el tema, haciendo variaciones en la metodología que sin embargo se basa en el mismo concepto. Nosotros hacemos uso no de histogramas sino de secuencias que representan la densidad de píxeles en cada columna de la imagen, y en el análisis de dichas secuencias podemos resolver ciertos problemas como la sobre-segmentación en letras tales como la u y la n.

Una vez que hemos explicado los elementos que conforman al pre-procesamiento de la escritura, ya sea de palabras o de caracteres aislados, en el Capítulo 4 se mostrarán los procesos de extracción de características, que en ambos casos se aplicarán a caracteres aislados, como paso anterior al reconocimiento de los mismos.

Capítulo 4

Extracción de Características

Como ya se ha mencionado, el estudio de ciertas características presentes en los dibujos de las letras es un análisis previo y necesario para el reconocimiento. Es en este análisis precisamente, en donde se obtienen todas aquellas características que serán determinantes para la diferenciación posterior entre caracteres. El conjunto de características obtenido es específico para cada uno de los cuatro métodos de reconocimiento utilizados por el sistema. Las curvas envolventes del carácter, las zonas de escritura presentes en el área activa, la presencia o ausencia de ciertos trazos, la inclinación, el tamaño, la altura, son solo algunos ejemplos de estas características.

En la Figura 4.1 se muestra el diagrama general de nuestro sistema, en donde se ubica el bloque correspondiente a la extracción de características. Pueden observarse los procesos que se realizaron en los Capítulos 2 y 3, y aquellos que se realizarán tomando como base los resultados obtenidos de este proceso.

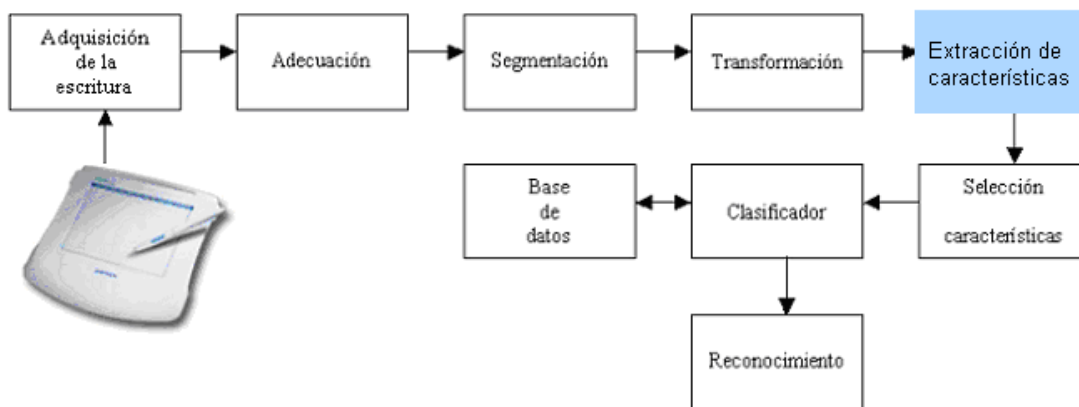


Figura 4.1. Ubicación del bloque de extracción en el diagrama general del sistema de RCM

Si la entrada inicial al sistema es el dibujo de la escritura de una palabra (cursiva), el proceso de segmentación previo permite su separación en tantas letras como caracteres hay en la palabra, de tal forma que cada una de dichas letras pase por el proceso siguiente de extracción. Sin embargo, si la entrada inicial al sistema es de un solo carácter, la segmentación ya no es necesaria, por lo que se pasa directamente de la captura a la extracción.

Para este bloque contamos básicamente con tres tipos de análisis geométrico de la imagen de la escritura: la extracción de envolventes, el análisis de sus trayectorias y el análisis de las zonas de dicha imagen. Estas estrategias proporcionan al sistema las características necesarias para realizar el reconocimiento posterior de letras o de palabras, a partir del reconocimiento de cada uno de los caracteres que las forman.

Aunque se ha revisado bibliografía relacionada con este tema, el análisis para la obtención de los conjuntos de características y los procedimientos de extracción se brindan en esta tesis, como una propuesta original para el tratamiento de la imagen del carácter a reconocer. Estos procesos se explicarán a continuación con mayor detalle, para cada uno de los métodos de reconocimiento utilizados.

4.1 Método por envolventes

Existe una diversa cantidad de características que pueden ser extraídas de una imagen y más específicamente, de una imagen de la escritura. Por ejemplo, podríamos obtener el ancho, el alto, la inclinación, la presencia o ausencia de ciertos enlaces, el grosor, etc. Por eso es necesario un análisis previo, que nos permita definir qué características serán utilizadas por nuestro sistema y por consiguiente, extraídas. El conjunto de características a extraer depende de la forma en la que se realizará la tarea de reconocimiento; el poder visualizar previamente esta tarea, nos permitirá elegir las características precisas que serán utilizadas. Hay que recordar además que en este bloque las imágenes de las cuales se extraen características son los dibujos de la escritura ya segmentada, es decir, las imágenes de cada uno de los caracteres que forman a la palabra.

En una de las estrategias de extracción de características propuestas para este sistema, utilizada en el Método de Reconocimiento por Envolventes (MRE) que mencionaremos en el Capítulo 5, se extraen las envolventes superior e inferior del carácter. Con dichas envolventes se realiza posteriormente un proceso de normalización y comparación entre los resultados obtenidos para los caracteres prueba y los caracteres prototipo, con la finalidad de obtener el prototipo con el cual se presenta la mínima diferencia. El método por envolventes se complementa además, con la extracción de otra característica, el promedio de puntos de corte vertical, lo que ayuda al sistema a diferenciar a aquellos caracteres que cuentan con envolventes muy similares.

Las características que el sistema extrae siguiendo este método son las siguientes:

1. Envoltente superior
2. Envoltente inferior
3. Promedio de puntos de corte

En la Figura 4.2 se muestra en forma esquemática el proceso de extracción de características por el método de envolventes.

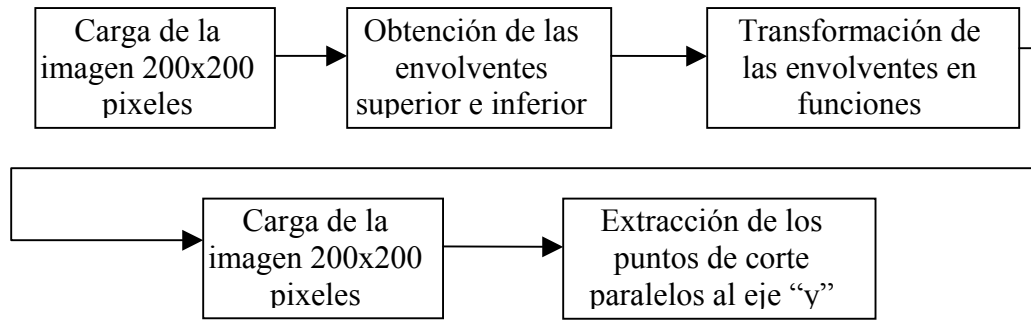


Figura 4.2. Diagrama del proceso de extracción de envolventes

4.1.1 Extracción de envolventes

Como se observa en el diagrama de la Figura 4.2, el método de extracción de envolventes involucra una serie de pasos, los que serán descritos a continuación.

En primer lugar, el sistema lee una matriz de 200x200 pixeles correspondiente a una imagen de prueba (carácter a reconocer). El carácter de prueba se captura mediante la tableta y la interfaz gráfica diseñada para tal fin, de tal manera que obtenemos una matriz B1 a partir de la imagen guardada, ver Figura 4.2.

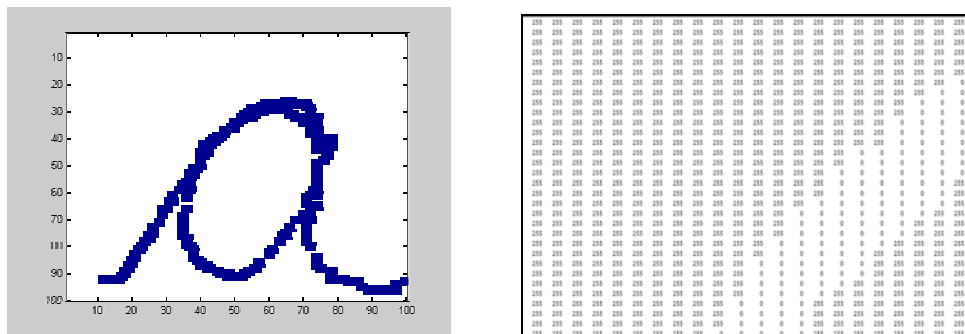


Figura 4.3. Carácter prueba capturado y un fragmento de su matriz correspondiente B1

El tamaño de la escritura de los usuarios fue obtenida mediante el promedio de los diferentes tamaños obtenidos de un proceso repetido de captura para diferentes usuarios, y arrojó como resultado un valor de 0.7 cm de altura y hasta 0.7 cm de ancho. Sin embargo la zona de escritura máxima para los caracteres prototipo se determinó de 1x1 cm (200x200 pixeles aproximadamente), para englobar así diferentes tamaños de escritura.

Considerando que el usuario tiende a escribir cubriendo toda la zona correspondiente a un recuadro determinado a su tamaño de escritura (menor o igual a 1 cm por lado), y debido a que se compararán las deformaciones generadas por el carácter prueba con las generadas por el carácter prototipo de la base de datos, considerando siempre que se cubrió toda la zona de escritura máxima, se realiza además un proceso de normalización. En este proceso, las funciones obtenidas tanto para la

prueba como para los caracteres prototipo se escalan y centran de tal forma que correspondan al tamaño del recuadro de escritura máxima (200x200 píxeles) y a la posición centrada de las funciones dentro del área de escritura. Para dicho escalamiento obtenemos una matriz PRU de 200x200 píxeles, en la cual la esquina inferior derecha corresponde al carácter de prueba. Ver Figura 4.4.

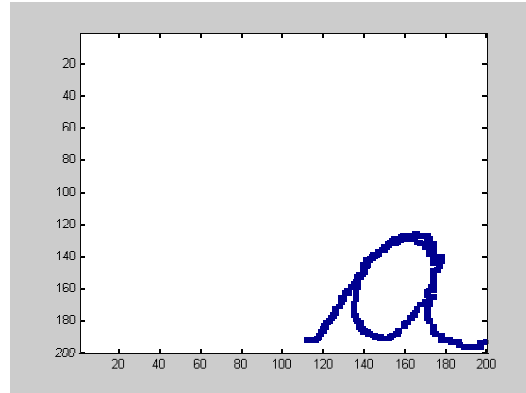


Figura 4.4. Carácter de prueba esquinado

Con el carácter de prueba colocado en dicha esquina se extrae de él la información necesaria para el escalamiento, como su altura y su longitud, entendiendo por ésta, la distancia máxima horizontal que existe entre el inicio y el final del dibujo del carácter.

El siguiente paso consiste en la extracción de la función que representará a la deformación superior de la imagen, para esto se hace un barrido de la matriz encontrando las coordenadas de todos los puntos superiores activos y guardando esa información como una función. La función del carácter de prueba todavía sin escalar, se muestra en la Figura 4.5.

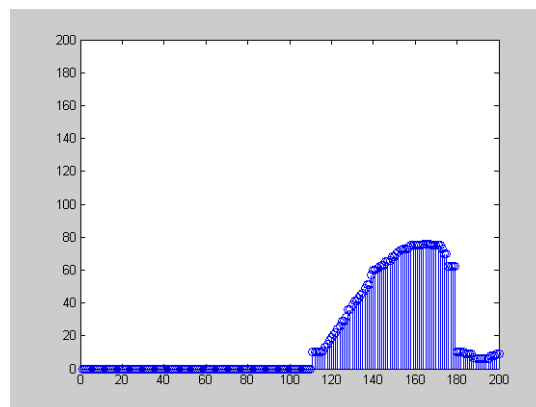


Figura 4.5. Alturas sin escalar

Las Figuras 4.5 y 4.6 se diferencian por el tamaño, además de la posición de la función dentro del área. Si la normalización no se realizara, al obtener las envolventes superior e inferior de la prueba y compararlas directamente con las de los prototipos que son funciones ya normalizadas, se ocasionaría un error en el reconocimiento.

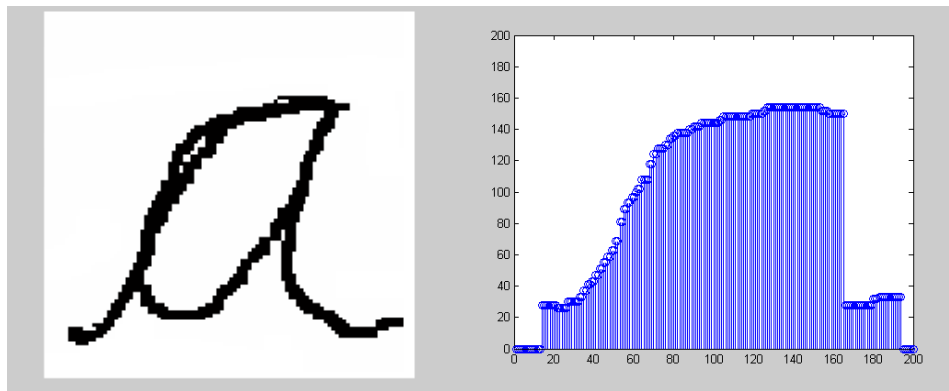


Figura 4.6. Carácter prototipo a y su envolvente superior escalada

El escalamiento y centrado, es un proceso de normalización que será, como ya explicamos, necesario para reducir errores en el reconocimiento. Se utilizan para ello factores de escalamiento tanto para las alturas como para la longitud del carácter de prueba. Para escalar la altura se obtiene un factor de escalamiento llamado “factor y”, que será el cociente de la altura del área de escritura (1 cm) y la altura de la prueba. Dicho factor es aplicado, multiplicando a la función del carácter prueba, con lo cual se normalizará la altura.

Por otro lado, para normalizar completamente a la función se obtiene además un factor de longitud en x multiplicando el factor de escala “factor x” por la longitud de la prueba. De esta manera la nueva función normalizada cubrirá toda la zona de escritura. Cuando se escala la altura y la longitud, la función se extiende, por lo que quedan espacios sin información de altura, lo que se corrige rellenando dichos espacios con el valor de altura anterior.

En la Figura 4.7 se muestra una comparación entre las funciones antes y después del proceso de normalización del carácter prueba. En la imagen se observan a las funciones sin escalar y ya escalada en altura y longitud. La función final ya normalizada, es la deformación superior de dicho carácter llamada “envolvente” o “función superior”.

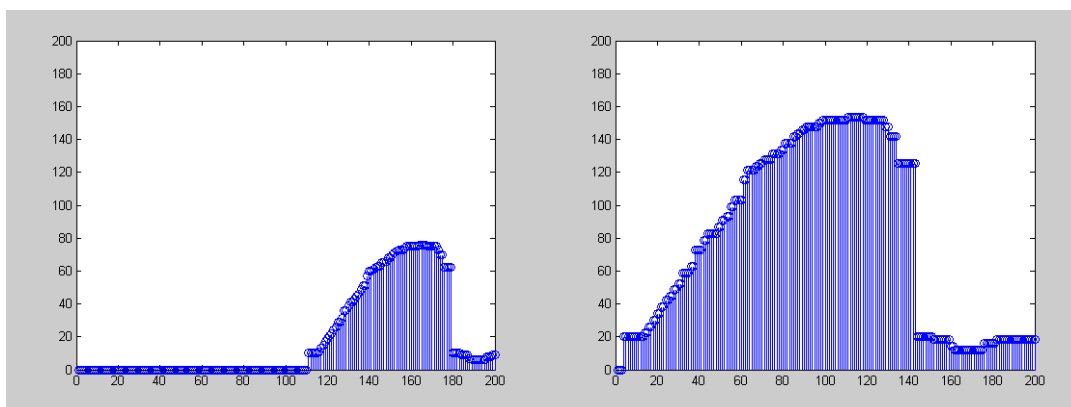


Figura 4.7. Envoltentes superiores del carácter prueba no normalizada y normalizada

Para obtener la envolvente inferior del carácter, el algoritmo realiza un proceso muy similar al anterior. Aunque en este caso la imagen original del carácter prueba es rotada 180°. Después de

aplicar este procedimiento, se llega al resultado mostrado en la Figura 4.8, la función envolvente inferior extraída ya normalizada.

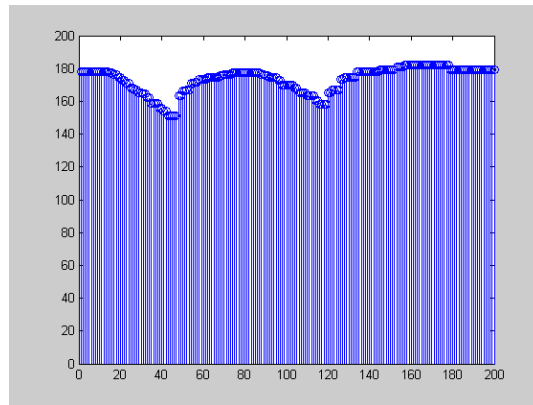


Figura 4.8. Envolvente inferior del carácter de prueba

4.1.2 Extracción de puntos de corte

Una vez que se obtienen las envolventes de los caracteres, el procedimiento continúa con la extracción del promedio de los puntos de corte vertical. Esta característica se obtiene como complemento para la extracción de envolventes, debido a que permite la diferenciación entre algunos caracteres, como la “c” y la “e”, para los cuales las envolventes extraídas suelen ser muy parecidas.

En el proceso de extracción se consideran líneas paralelas al eje “y”, que cortan la imagen del carácter tal como se observa en la Figura 4.9.

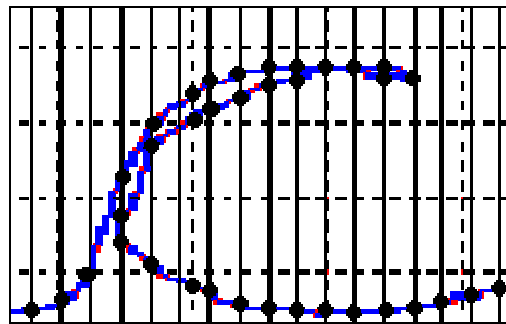


Figura 4.9. Cortes verticales del carácter c

El promedio de cortes “cor” es obtenido para el carácter prueba anterior, utilizando la siguiente expresión:

$$\text{cor} = \left(\sum_{i=1}^n \text{pci} \right) / n$$

donde p_{ci} , es la cantidad de puntos de corte para la línea de corte i , y n , es la cantidad de líneas de corte ($n = 200$).

El proceso de obtención de este valor, se basa en el barrido de la matriz generada por la imagen, en dicha matriz se localizan los puntos activos, haciendo un conteo de cada uno de ellos por columnas y calculando posteriormente su promedio. De este método de extracción se obtiene para cada carácter de prueba, un vector formado por 201 elementos, en donde los primeros 200 son los valores de la función envolvente extraída y el último valor, es el promedio de puntos de corte.

El método de extracción, descrito anteriormente es aplicado a todos los caracteres prototipo de la base de datos de imágenes, obteniendo de esta manera vectores representativos para cada carácter, formando con ellos una nueva base de datos de vectores. Lo anterior le permitirá al sistema, comparar las características extraídas para cualquier carácter prueba con las extraídas para los caracteres prototipo. Sobre este proceso posterior de comparación, se hablará más adelante, en el capítulo 5.

4.2 Método por trayectorias

Otra alternativa incluida también para este sistema, es la utilizada en el Método de Reconocimiento por Trayectorias (MRT), que obtiene una serie de características geométricas de las trayectorias que forman a la envolvente superior del carácter.

A diferencia del procedimiento anterior, en este método, más que obtener las envolventes, nos interesa obtener el conjunto de características que representarán a esas envolventes, tales como el número de trayectorias suaves que la componen, sus pendientes, proporciones y direcciones, etc. Posteriormente esas características serán comparadas con la finalidad de obtener la identidad del carácter.

En la Figura 4.10 mostramos el diagrama a bloques del proceso de extracción de características por este método.

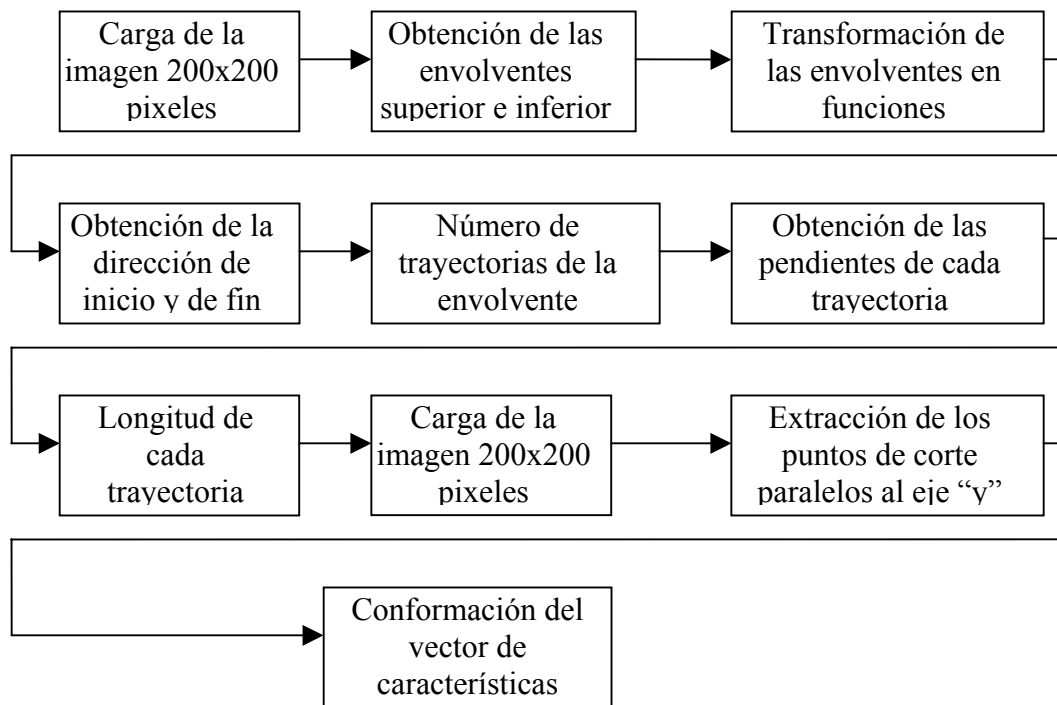


Figura 4.10. Diagrama del método de extracción por trayectorias

En la estrategia por trayectorias se realiza un análisis de la imagen de la escritura, considerando ya no las deformaciones, sino las características de las trayectorias que las forman. Es así como en el proceso de extracción obtenemos como resultado un conjunto de vectores de características. En un procedimiento posterior, estos vectores serán comparados para obtener un reconocimiento. En la Figura 4.11 observamos el dibujo del carácter prototipo “a” y su envolvente superior.

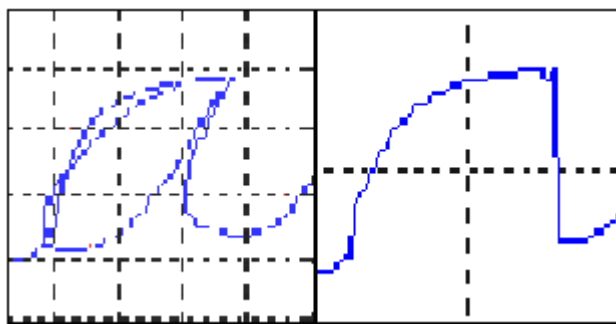


Figura 4.11. Imagen del prototipo y su envolvente superior

Del análisis de las características geométricas de las funciones envolventes obtenidas, que se describirán más adelante, se construyen vectores característicos como el siguiente:

$$F = [1, 1, 3, 0.18, -0.5, 0.1, 0, 0, 0, 0.8, 0.02, 0.18, 0, 0, 0, 3.25]$$

El vector F fue generado, aplicando el proceso de extracción de características para el prototipo “a”.

La estructura de los vectores y los parámetros propuestos en esta tesis se muestran a continuación:

$$F = [\text{dirini}, \text{dirfin}, \text{trayec}, m(1), m(2), m(3), m(4), m(5), m(6), \text{lt}(1), \text{lt}(2), \text{lt}(3), \text{lt}(4), \text{lt}(5), \text{lt}(6), \text{cor}]$$

Observamos que el vector F está formado por dieciséis valores que representan a las características extraídas de las trayectorias suaves que forman a la envolvente del carácter, cada una de ellas se define a continuación y se da una explicación del proceso de extracción.

Las variables *dirini* y *dirfin* hacen referencia al signo de la pendiente de la trayectoria inicial y final, respectivamente, y pueden tomar el valor de 1 ó 2. 1 significa que la trayectoria tiene una pendiente positiva (su dirección es de abajo hacia arriba), mientras que el 2, corresponde a una pendiente negativa (de arriba hacia abajo).

La variable *trayec* representa el número de trayectorias presentes en la envolvente, las cuales se obtienen mediante la detección del cambio de dirección.

Las pendientes $m(n)$ se consideran respecto al eje x.

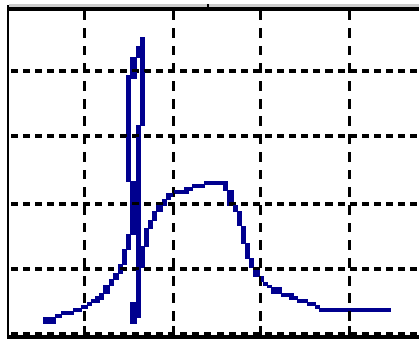
Las variables $lt(n)$ representan el porcentaje equivalente a las proyecciones de cada una de las trayectorias, con respecto a la longitud total de la envolvente en el eje x.

Por último *cor*, es el promedio de los puntos de corte con respecto al eje x del carácter. Se debe aclarar, que esta última característica, es la misma que se extrae para el Reconocimiento por Envolturas y que la finalidad de su extracción es similar al caso anterior, aunque su valor ya no es tan decisivo, sino que forma parte de un conjunto de características obtenidas todas con la misma aportación para el reconocimiento.

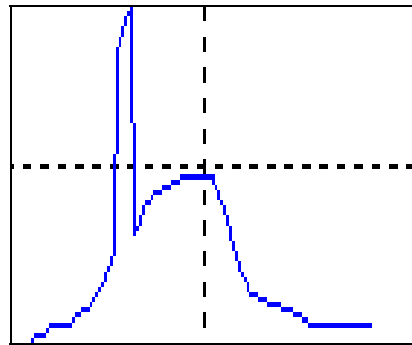
En resumen, las características extraídas por este método son las siguientes:

1. *dirini*: es la dirección inicial de la trayectoria. Puede tener solo dos valores, 1 si la dirección es positiva o -1 si es negativa o cero. La Figura 4.12 muestra dos ejemplos para esta característica.

$dirini = 1$

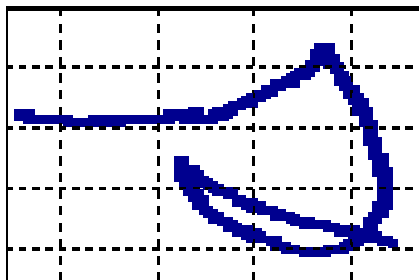


Prototipo del carácter h

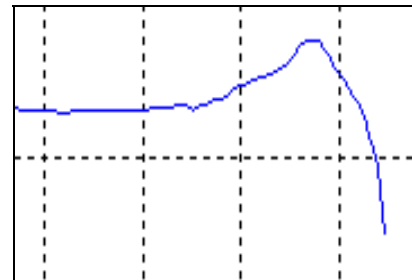


Envolvente superior del carácter h

$dirini = -1$



Prototipo del carácter s



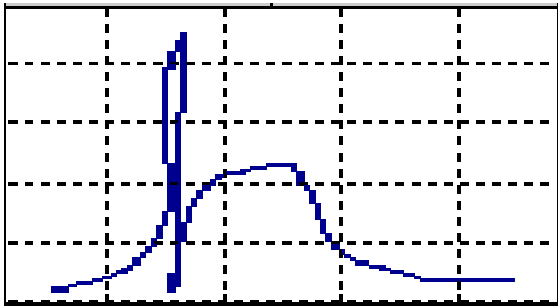
Envolvente superior del carácter s

Figura 4.12. Ejemplos de caracteres con diferentes valores para $dirini$

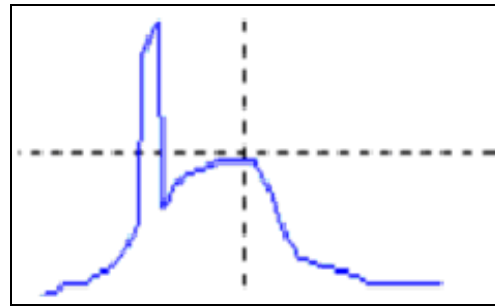
El proceso de obtención de esta característica parte de un barrido de la matriz generada por la imagen de la envolvente, en donde se localiza una secuencia inicial de píxeles con valor de 255, para los cuales el valor de i mantiene una variación mínima. De esta secuencia se extrae la pendiente, si esta es positiva, a $dirini$ se le asigna un valor de 1, por el contrario si es negativa o cero se le asigna un valor de -1 .

2. **dirfin**: es la dirección final del trazo. Esta característica, al igual que la anterior, puede tomar solo dos valores, 1 si la dirección es positiva o -1 si es negativa o cero. La Figura 4.13 muestra ejemplos de esta característica.

dirfin= -1

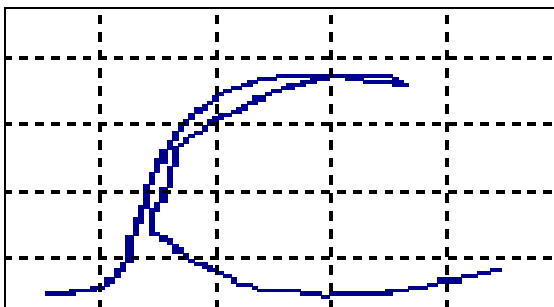


Prototipo del carácter h

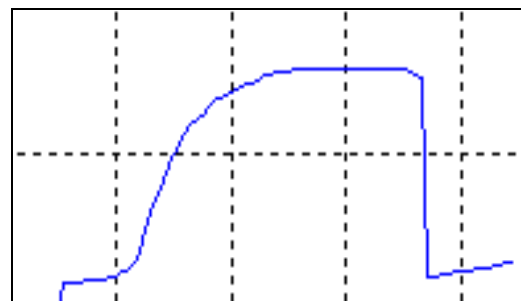


Envolvente superior del carácter h

dirfin= -1



Prototipo del carácter c



Envolvente superior del carácter c

Figura 4.13. Ejemplos de caracteres con diferentes valores para dirfin

El proceso de obtención de esta característica parte de un barrido de la matriz generada por la imagen, en donde se localiza una secuencia final de pixeles con valor de 255, para los cuales el valor de i mantiene una variación mínima. De esta secuencia se extrae la pendiente, si esta es positiva, a dirini se le asigna un valor de 1, por el contrario si es negativa o cero se le asigna un valor de -1 .

3. **trayec**: es el número de trayectorias que forman a la función. Para obtener esta característica, se realiza un barrido de la envolvente, localizando los cambios de signo de la pendiente de la línea recta que une el primer punto con el siguiente, y así sucesivamente. Las Figuras 4.14 y 4.15 muestran ejemplos de la extracción de esta característica.

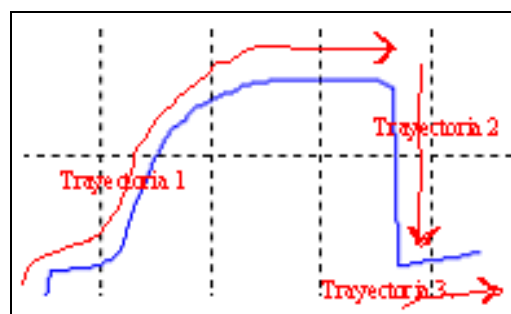
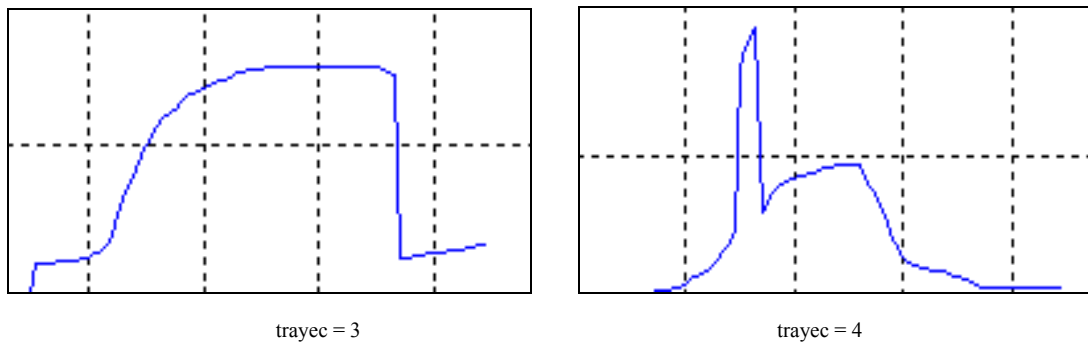


Figura 4.14. Trayectorias que forman a la función



trayec = 3

trayec = 4

Figura 4.15. Caracteres con diferentes valores para trayec

4. $m(i)$: es la pendiente de cada una de las trayectorias que forman a la envolvente. Cuando no existe la trayectoria $m(i)$, el valor asignado es 0. Para extraer esta característica, se obtienen dos puntos por cada trayectoria, el punto de inicio y el de fin. Se considera una recta que pasa por esos dos puntos y su pendiente es asignada a la característica $m(i)$. Considerando la función del prototipo para el carácter “c”, podemos obtener las pendientes de la forma mostrada en la Figura 4.16.

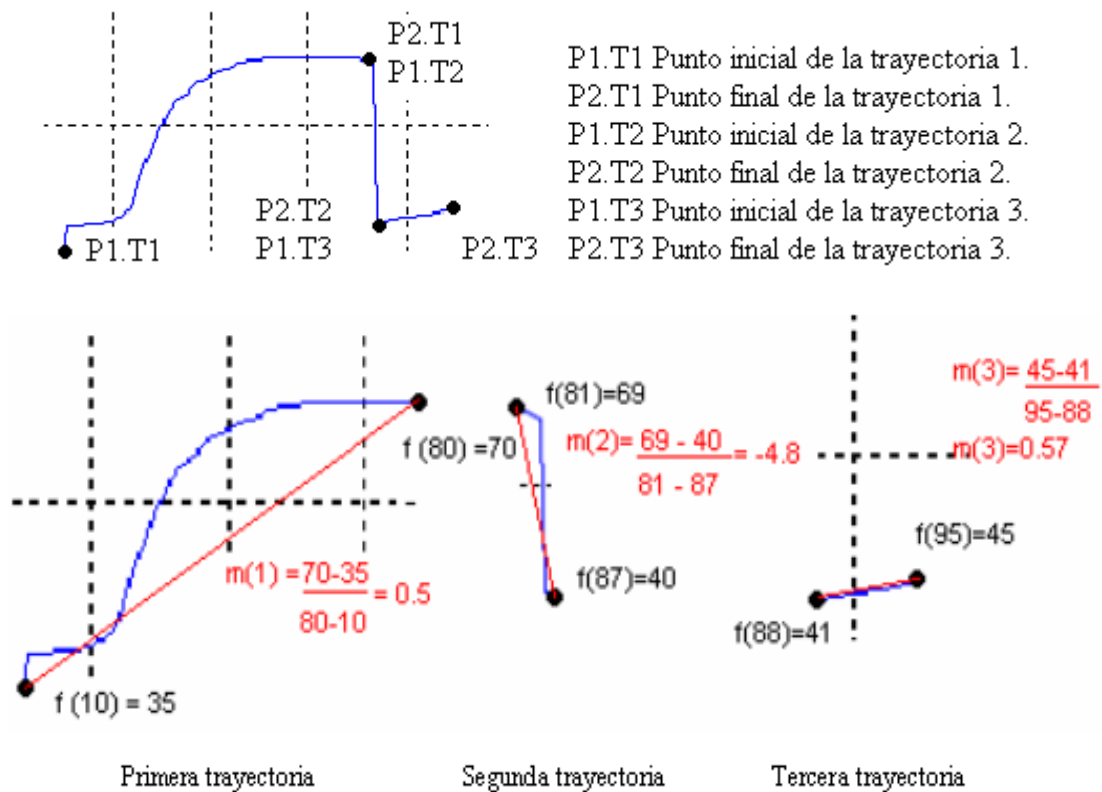


Figura 4.16. Descomposición de la función en trayectorias y obtención de cada una de las pendientes

La forma de obtención de esta característica, parte de la ya conocida expresión geométrica, para obtener la pendiente de una recta en el plano:

$$m = (y_2 - y_1)/(x_2 - x_1)$$

Como los puntos iniciales y finales de las trayectorias obtenidos con anterioridad, están definidos por sus coordenadas x e y , es posible utilizar la expresión anterior y conocer la pendiente.

De esta manera, obtenemos del carácter de ejemplo, los siguientes valores para las pendientes:

$$m(1) = 0.5$$

$$m(2) = -4.8$$

$$m(3) = 0.57$$

$$m(4) = 0$$

$$m(5) = 0$$

$$m(6) = 0$$

Como habíamos indicado, las pendientes correspondientes a las trayectorias 4, 5 y 6 tienen asignado un valor cero, debido a que para el ejemplo, dichas trayectorias no existen.

5. **$lt(n)$** : es la proporción correspondiente a la proyección de una trayectoria con respecto a la longitud total que ocupa el carácter en el eje x . Para calcular estos parámetros, primero se retoma el valor de la longitud total del carácter en el eje x , **LT** . Además se utilizan las coordenadas de los puntos inicial y final, obteniendo la diferencia en x para cada trayectoria, utilizando la expresión:

$$Lt = x_2 - x_1$$

Con esta longitud **Lt** por trayectoria y la total, **LT** , se obtiene la proporción correspondiente a cada una de las trayectorias, con respecto a la total. La expresión utilizada para el cálculo final de estos parámetros es la siguiente:

$$lt = \frac{x_2 - x_1}{x_f - x_i} = \frac{Lt}{LT}$$

Una explicación gráfica de la obtención de estos parámetros se muestra en la Figura 4.17.

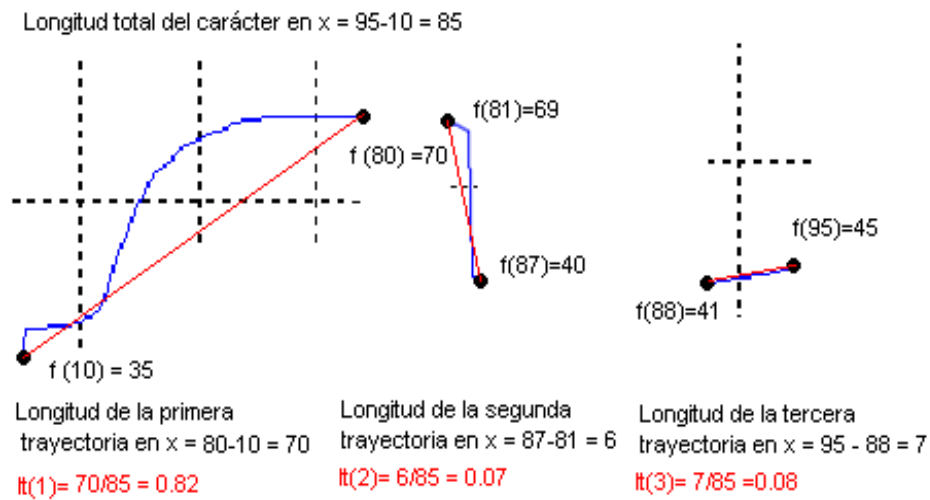


Figura 4.17. Obtención de los valores de $lt(n)$, para el prototipo del carácter c.

Los valores obtenidos, para el ejemplo anterior son los siguientes:

$$lt(1) = 0.82 \quad lt(2) = -0.07 \quad lt(3) = 0.08 \quad lt(4) = 0 \quad lt(5) = 0 \quad lt(6) = 0$$

Como habíamos indicado para el caso de las pendientes, aquí nuevamente se asigna a las trayectorias 4, 5 y 6 un valor de cero, debido a que para este caso dichas trayectorias no existen. Además hay que considerar, que dichos valores fueron truncados a centésimas.

6. **Cor:** es el promedio de puntos de corte del carácter, si consideramos líneas paralelas al eje “y” cortando la imagen de la letra. Este parámetro ya había sido utilizado para el método anterior, aunque en este caso se retoma como una característica más, con la misma aportación que los demás parámetros extraídos.

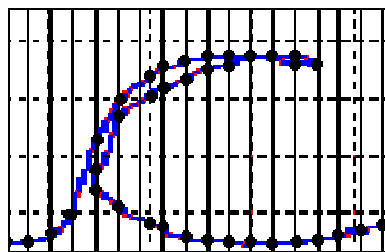


Figura 4.18. Cortes en el eje x del carácter c

La obtención del promedio de cortes “cor” se realiza de la misma forma que en el método anterior, haciendo un barrido de la matriz generada por la imagen, localizando los puntos activos y haciendo un conteo de todos ellos por columnas, calculando posteriormente su promedio.

Una vez obtenidas todas las características anteriores, se ordenan y guardan en forma de elementos de un vector específico para cada carácter. Un conjunto de este tipo de vectores es obtenido para todas las letras de la base de datos de imágenes, que consta de trece caracteres prototipo. Posteriormente esos vectores serán comparados con el vector obtenido para un determinado carácter prueba, con la finalidad de encontrar dentro de los caracteres prototipo uno igual o semejante y poder asignar ese carácter como el reconocido.

4.3 Método por vectores

Este método utiliza la comparación de vectores, obtenidos de la extracción por zonas, de ciertas características de las imágenes capturadas. El proceso de extracción, resulta de un análisis por regiones de la imagen, considerando que cada una de las letras, presenta en una zona específica cierta información característica. Este método se encuentra formando parte del bloque de extracción, dentro de una alternativa más brindada al usuario para el reconocimiento: el Método de Reconocimiento por Vectores (MRV).

Las características extraídas para el análisis utilizando este método son las siguientes:

1. **El ancho horizontal del cuerpo del carácter:** es la distancia horizontal máxima de separación en el cuerpo del carácter, definido este último como la zona de la imagen del carácter que no es un enlace.
2. **El ancho vertical del cuerpo del carácter:** es la distancia máxima vertical de separación en el cuerpo del carácter, definido este último como la zona de la imagen del carácter que no es un enlace o una saliente angosta.
3. **La altura del carácter:** es la diferencia entre el punto máximo y el mínimo en la imagen del carácter.
4. **La posición del término del dibujo del carácter:** es alta o baja, dependiendo de si termina en la parte superior o inferior de un eje ubicado a la mitad horizontal del carácter.
5. **Ancho de la zona media del carácter:** es el ancho del carácter medido en su zona media.
6. **Carácter cerrado o abierto:** en ocasiones, cuando trazamos un carácter, nuestro trazo parte de un punto y regresa a ese mismo punto en algún momento en el trazo del carácter, encerrando una zona, a este tipo de caracteres les llamamos caracteres cerrados. Aquellos en los cuales esto no se presenta, serán caracteres abiertos.
7. **Ancho horizontal a partir del punto máximo:** es el ancho que tiene el carácter contándolo a partir del punto máximo hacia ambos lados.

La manera de medir o extraer las características mencionadas anteriormente, se explicará en las siguientes líneas, basándonos en una imagen capturada como carácter de prueba. Sin embargo comenzaremos explicando algunos conceptos que nos ayudarán a entender mejor el proceso.

4.3.1 Conceptos básicos

A continuación describiremos algunos conceptos que nos ayudarán a comprender mejor el método de vectores para la extracción de características.

1. **Enlace:** trazo o línea que nos sirve para unir caracteres manuscritos aislados, puede ser inicial o final.
2. **Cuerpo:** zona de la imagen del carácter que no contiene enlaces.
3. **Punto máximo general:** es el punto activo que tiene la mayor altura del carácter.
4. **Saliente:** es un trazo que forma parte del cuerpo, casi vertical cuya diferencia en altura respecto a los demás puntos activos del cuerpo sea notoria.
5. **Punto mínimo:** es el punto activo con menos valor de altura del carácter.

En las Figuras 4.19 y 4.20, se indican gráficamente cada uno de los conceptos anteriores, para dos caracteres de prueba distintos.

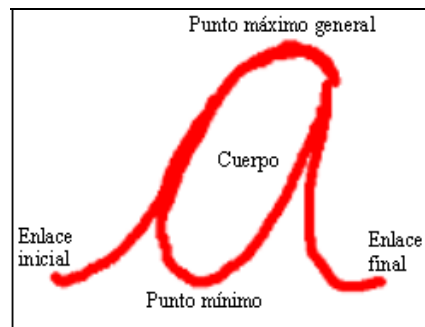


Figura 4.19. Cuerpo, enlaces, punto mínimo y punto máximo general en un carácter bajo



Figura 4.20. Cuerpo, enlaces, puntos mínimo y máximo general, y saliente en un carácter alto

También cabe aclarar, que de la imagen del carácter de prueba capturado, se obtiene la matriz binaria correspondiente y que a partir de ésta, son extraídas todas las características. La Figura 4.21 muestra un carácter de prueba con un fragmento de su matriz.

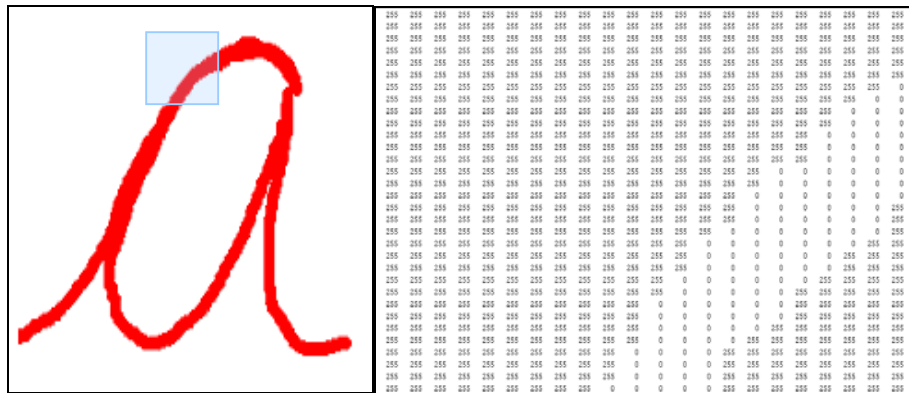


Figura 4.21. Carácter de prueba y fragmento de su matriz binaria

En la Figura 4.22 se muestra en diagrama a bloques la secuencia a seguir para la extracción de las características para el método de vectores.

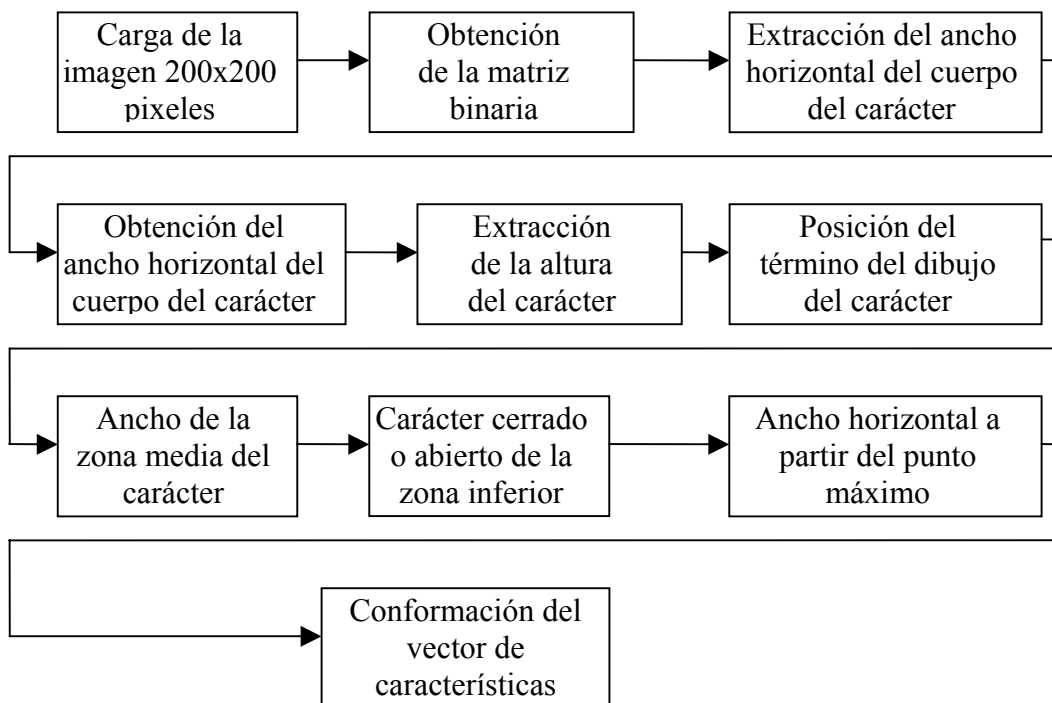


Figura 4.22. Diagrama a bloques de la extracción para vectores

En las siguientes secciones se describirán los procesos de extracción de características necesarias para el Reconocimiento por Vectores.

4.3.2 Métodos de extracción

1. Extracción del ancho horizontal

La primera característica a extraer es el ancho horizontal del carácter. Para ello se realiza un barrido de la matriz que genera la captura de la imagen del carácter prueba. Este barrido es en dirección horizontal, de tal forma que se obtienen un primer punto o inicio del cuerpo del carácter y un punto final o fin del cuerpo del carácter. Con estos dos valores se realiza una diferencia y dependiendo del resultado obtenido se le asigna un valor a este parámetro de entre tres posibles: 0, 1, 2, para las distancias pequeñas, medianas y grandes, respectivamente.

Para obtener los rangos de cada una de las distancias, se agruparon las letras del sistema, en tres conjuntos, el primero, de letras con anchos pequeños, como la “i”, la “t”, la “c”; el segundo, de anchos medianos, como la “e”, la “r”, la “s”, la “v” y finalmente el tercero, de anchos grandes, como la “a”, la “o”, la “u”, la “d”, la “h”, la “n”. Se midieron las distancias horizontales máximas de los cuerpos de las letras por cada uno de los conjuntos. Con esas mediciones se obtienen los rangos correspondientes a cada uno de los conjuntos.

En la Figura 4.23 se muestra el proceso de obtención del ancho horizontal a partir de la matriz generada por la imagen capturada.

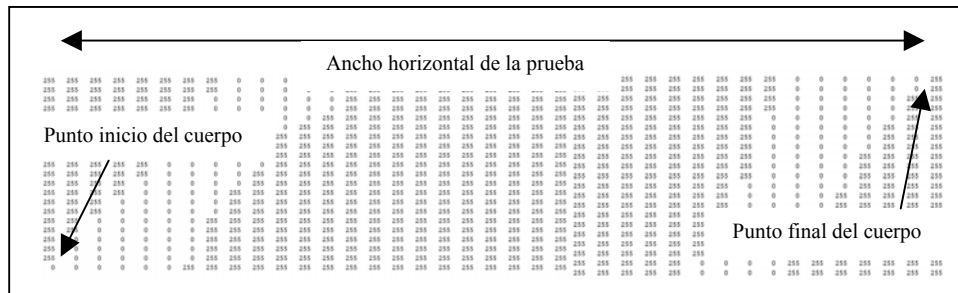


Figura 4.23. Puntos obtenidos del barrido para la extracción del modo del ancho horizontal

La Figura 4.24 muestra diferentes valores obtenidos para este parámetro.

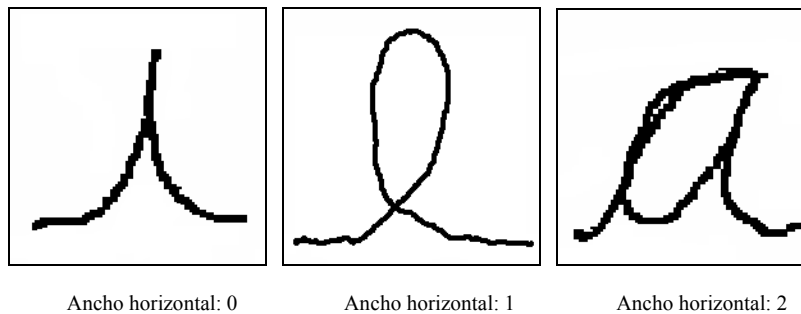


Figura 4.24. Anchos horizontales

2. Extracción del ancho vertical

La segunda característica a extraer es la del ancho vertical. Para ello, el sistema realiza un barrido de la matriz en forma similar al anterior. Aunque en este caso, la dirección será vertical, de tal forma que se obtienen un punto máximo o superior de la zona media del carácter, así como un punto mínimo o inferior. Con ambos puntos, se calcula la diferencia y dependiendo de ese valor se le asigna a este parámetro, un número de entre tres posibles: 0, 1, 2, para anchos pequeños, medianos o grandes, respectivamente. La manera de obtener los rangos para cada uno de los valores del ancho vertical, es la misma que para los anchos horizontales. En la Figura 4.25 se muestra el proceso de obtención del ancho vertical a partir de la matriz generada por la imagen capturada.

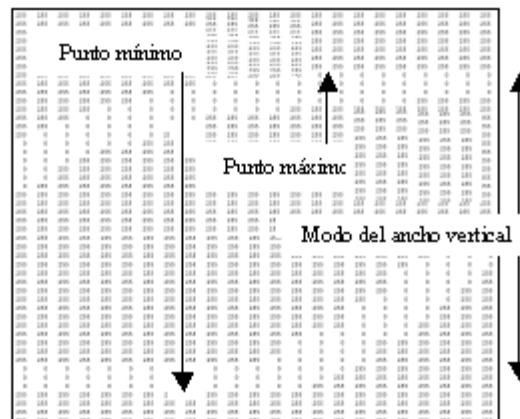


Figura 4.25. Obtención del ancho vertical a partir de la matriz del carácter de prueba

De la misma manera que para el ancho horizontal, mostramos en la Figura 4.26 diferentes valores para este parámetro.

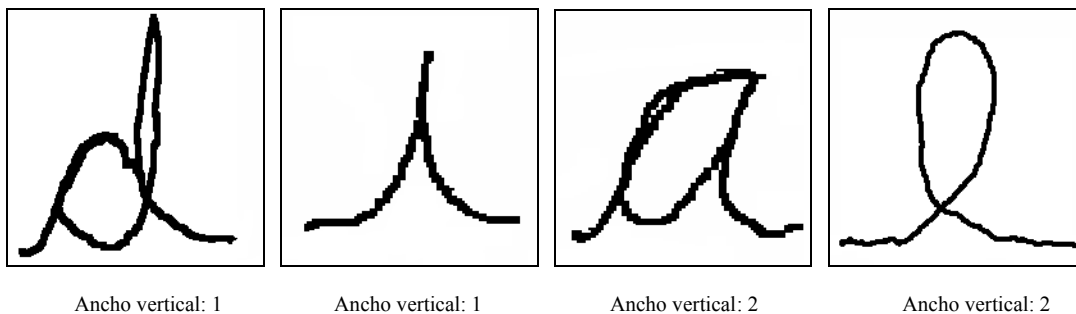


Figura 4.26. Anchos verticales

3. Extracción de la altura del carácter

Para obtener la altura del carácter se calcula la diferencia entre el punto máximo general y el punto mínimo del carácter. Si esta distancia es del 70% o más de la altura del renglón de escritura (1 cm), se considera un carácter alto, asignando al modo de esta característica un valor “1” y en caso

contrario, un carácter bajo con un valor “0” para este parámetro. La Figura 4.27, muestra dos caracteres con distinto valor de altura.

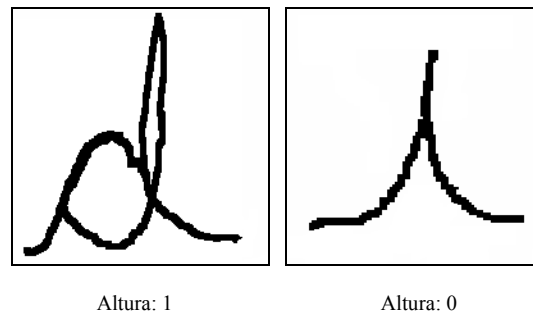


Figura 4.27. Alturas

4. Extracción de la posición final

Para obtener la posición final, se realiza un barrido de la matriz localizando el pixel correspondiente al último punto del enlace final del carácter y se obtiene la altura de este pixel. Si dicha altura es mayor al 30% se considera un final alto y si es menor o igual, se considera un valor bajo. La Figura 4.28, muestra dos caracteres con diferente posición final.

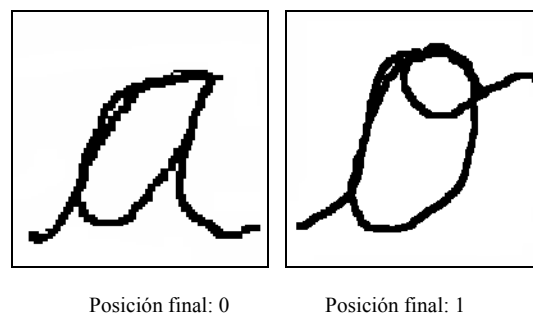


Figura 4.28. Posiciones finales

5. Extracción del ancho medio del carácter

El ancho medio del carácter se obtiene también con un barrido de la matriz en la zona media del carácter, a diferencia del ancho horizontal, que obtiene el ancho del cuerpo. Para extraer el ancho medio, se localizan los pixeles activos extremos en dirección horizontal y con ellos se calcula la diferencia, obteniendo así la distancia media. Los valores que puede tomar esta característica son 0, 1, 2, dependiendo del valor del ancho. Los rangos se obtienen como se ha explicado para los casos anteriores. En la Figura 4.29 se muestran letras con diferentes valores para esta característica.

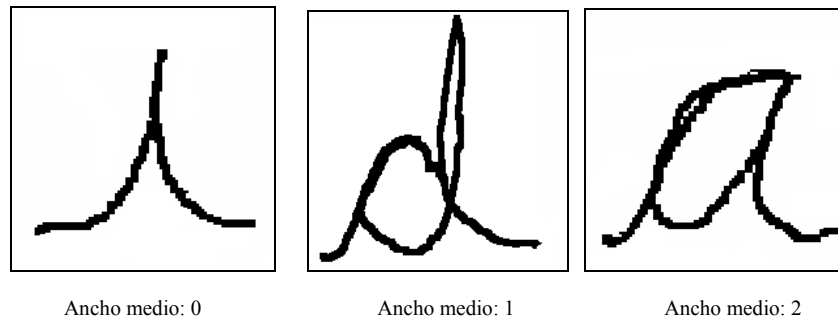


Figura 4.29. Diferentes valores para Anchos medios

6. Extracción del parámetro “carácter abierto o cerrado”

La siguiente característica extraída es la que se refiere a un carácter abierto o cerrado. Para ello se hace un barrido de la matriz localizando zonas cerradas en el carácter. También si la zona inferior del carácter (de la altura 1 a la 30), presenta más de quince píxeles inactivos, consideraremos a dicha letra como abierta. Los valores que puede tomar el modo de esta característica son 0 y 1, dependiendo de si son cerrados o abiertos respectivamente.

La Figura 4.30, muestra dos caracteres con diferentes valores para este parámetro.

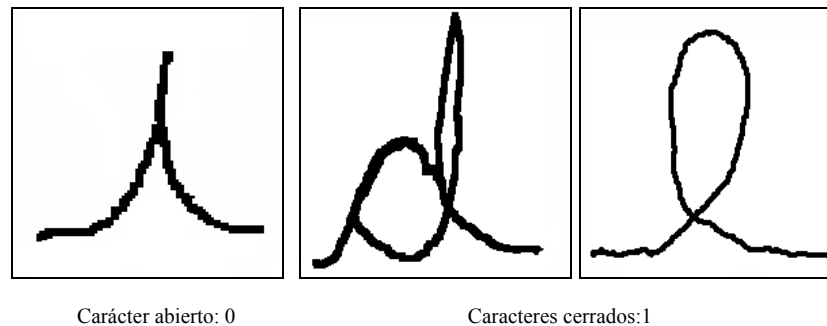


Figura 4.30 Caracteres cerrados y abiertos

7. Extracción del ancho horizontal a partir del punto máximo

Finalmente, se obtiene el ancho horizontal a partir del punto máximo, para ello se localiza el punto máximo del carácter y se cuenta por medio de un recorrido en la matriz, el número de píxeles activos tanto en una dirección como en otra (derecha e izquierda), hasta una distancia de diez píxeles de cada lado. Dependiendo de la cantidad contabilizada de puntos activos se asignan los valores 0, 1, 2, a esta característica.

La Figura 4.31 muestra tres letras con diferente valor para este parámetro.

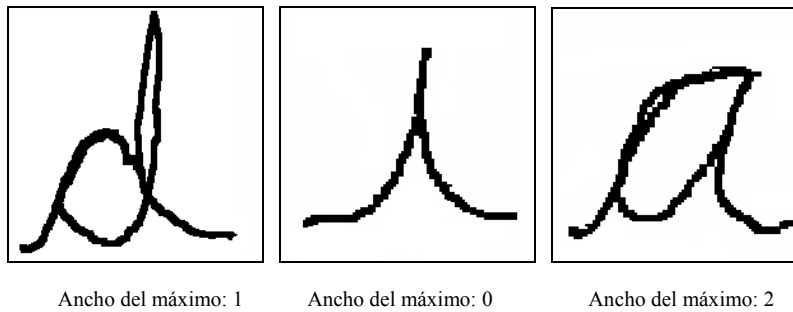


Figura 4.31 Caracteres con diferentes anchos horizontales a partir del punto máximo.

Una vez calculadas todas las características anteriores para un determinado carácter, se guardan en un vector, llamado de “características”, de la siguiente manera:

$$\text{Vectorp} = [2 \ 2 \ 0 \ 0 \ 2 \ 0 \ 2]$$

La estructura de dicho vector es la siguiente:

$$\text{Vectorp} = [\text{AH} \ \text{AV} \ \text{H} \ \text{PF} \ \text{AM} \ \text{AC} \ \text{APM}]$$

[Ancho Horizontal, Ancho Vertical, Altura, Posición Final, Ancho Medio, Abierto o Cerrado, Ancho del Máximo]

Estos vectores serán obtenidos tanto para los caracteres prototipo, como para los caracteres prueba, con la finalidad de compararlos y encontrar entre los primeros, aquel carácter que tenga un vector que coincida con el obtenido para el carácter de prueba.

Después de definir los procesos de extracción de características, será necesaria la creación de las bases de datos, para la posterior comparación entre las características obtenidas para los caracteres prototipo con las obtenidas para los caracteres prueba.

4.4 Bases de datos

En un sistema de reconocimiento la base de datos tiene como finalidad la de brindar al sistema la información necesaria para los procesos de extracción y comparación. En la base de datos es en donde se guardan las imágenes de los caracteres considerados como prototipo o sus características extraídas. El proceso de creación de las bases de datos es previo a la comparación y se guardan en ellas las características específicas requeridas para cada método de reconocimiento utilizado.

El diccionario del sistema de reconocimiento cuenta con todas las letras que forman a las palabras: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve. Aunque el sistema, además de reconocer las palabras anteriores también puede reconocer todas aquellas palabras de no mas de seis caracteres que se puedan escribir con la combinación de las letras del diccionario. Para los Métodos de Reconocimiento por Envoltentes y Trayectorias, se elaboró una base de datos compuesta por

imágenes, mientras que para los de optimización de ese mismo método, y el Método de Reconocimiento por Vectores se elaboraron dos bases compuestas por elementos vectoriales.

La información contenida en una base de datos, es tanta según el método para el cual se utilice. Sin embargo, hay que considerar que mientras más muestras se obtengan de cada carácter, la información guardada será más representativa en cuanto a las posibles variaciones de la escritura (tamaño, posición, inclinación, etc.).

4.4.1 Base de datos de imágenes

La primera base de datos creada fue la de imágenes. Esta base es no relacional además tiene como etiquetas a las letras: a, c, d, e, h, i, n, o, r, s, t, u, v, que asocian imágenes en formato BMP (200x200 pixeles), que son también elementos de entidad. Ser un elemento de entidad significa que dicho elemento representa a un objeto o a un concepto del mundo real. Ya que la cantidad de imágenes a guardar no es muy grande (13 imágenes), no se requiere la aplicación de un proceso de compresión para su almacenamiento. Tampoco requiere un sistema de búsqueda estructurado, ya que a cada etiqueta solo le corresponde un único dato asociado que difiere de los demás. Cada dato asociado tiene como entrada la misma etiqueta.

En la Figura 4.32 mostramos de forma esquemática el proceso de obtención del prototipo por autor para la base de datos de imágenes.

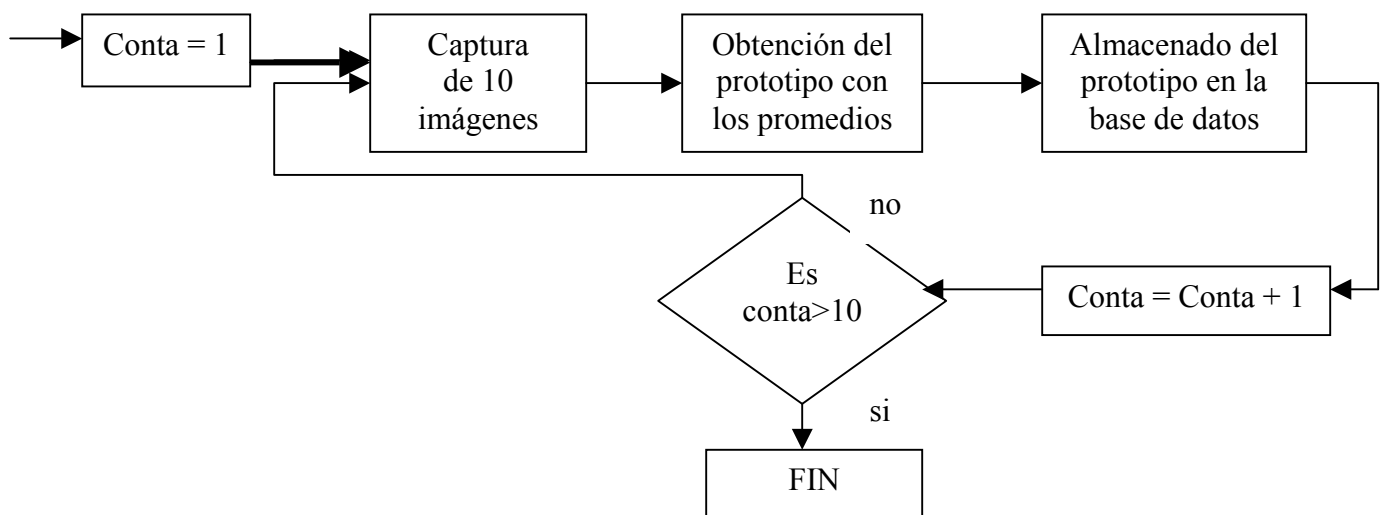


Figura 4.32 Esquema de obtención de prototipos por autor

En la elaboración de esta base, participaron 10 usuarios, cada uno de ellos realizó la captura de diez dibujos de la misma letra, es decir, escribió diez caracteres “a”, diez “e”, etc., con ayuda de una interfaz elaborada para tal fin. La imagen de la interfaz de captura se muestra a continuación en la Figura 4.33.

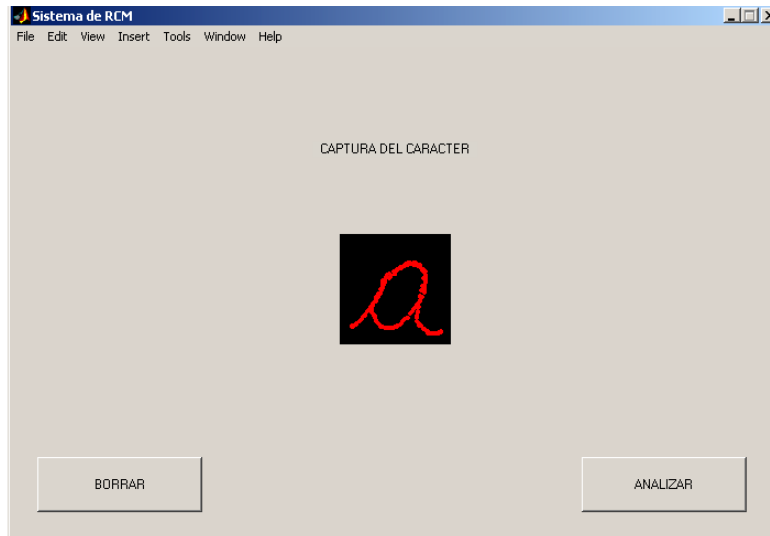


Figura 4.33. Interfaz de captura de caracteres para la base de datos de imágenes

La interfaz cuenta con dos botones. Con el botón ANALIZAR se guarda el carácter que se acaba de dibujar en una matriz que corresponde a una variable particular. Para obtener diez matrices distintas por cada carácter se repite la captura diez veces. El ancho de la línea de dibujo es de tres píxeles, igual que en la interfaz de captura del sistema, por lo que no se requiere de un proceso de adelgazamiento y/o engrosamiento de los caracteres capturados. Con el botón BORRAR, se tiene la opción de rescribir el carácter.

El área activa de la interfaz es de 200x200 píxeles, así que el tamaño de cada imagen de la base estuvo limitado por un recuadro del mismo tamaño. En nuestra tableta ese recuadro, corresponde a una zona de 1x1 cm, siendo el área de escritura máxima para un carácter. Cada carácter se almacena en una matriz diferente y una vez capturados los diez caracteres, se registra la altura de cada píxel en cada una de las imágenes, obteniendo con el promedio de estas alturas los píxeles activos para la imagen prototipo promedio por autor. De la misma manera se obtiene una imagen prototipo promedio general (de todos los autores). A partir de esta imagen se extraen las características requeridas para la posterior comparación.

La imagen prototipo tendrá entonces las características más comunes de la escritura en general. De esta manera, obtenemos todas las imágenes de los caracteres prototipo necesarias para el Método de Reconocimiento por Envoltentes utilizado por el sistema. Contamos hasta aquí, con las imágenes de trece caracteres en una zona de escritura máxima de 200x200 píxeles, en formato bmp.

La Figura 4.34 muestra la imagen prototipo obtenida para el carácter “a”.



Figura 4.34. Imagen prototipo de la base de datos de imágenes

Para el Método de Trayectorias también se utiliza la base de datos anterior, sin embargo en su versión optimizada se elaboró una base adicional a partir de la ya mencionada.

4.4.2 Base de datos de vectores (MRTO)

Esta base de datos se usa en las comparaciones del método de reconocimiento por trayectorias optimizado explicado en el Capítulo 5. Esta base está formada por un conjunto de vectores con 16 elementos cada uno. Dichos elementos se obtienen del proceso de extracción de características para las imágenes prototipo de la base de datos de imágenes anterior. La nueva base formada por vectores también es no relacional y tiene como etiquetas a las letras: a, c, d, e, h, i, n, o, r, s, t, u, v, aunque en este caso sus datos asociados son vectores. Cuenta con elementos de entidad y modelos de alto nivel como atributos que describen a la entidad carácter manuscrito.

En el siguiente diagrama se muestra de forma esquemática la conformación de esta base de datos.

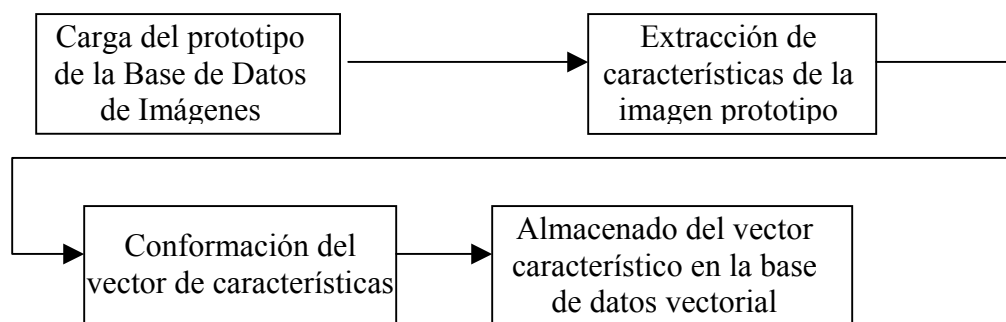


Figura 4.35. Diagrama del proceso de obtención de la base de vectores

La Figura 4.36 muestra la imagen prototipo del carácter “a” de la que se parte para la obtención del siguiente vector prototipo:

$$F = [1, 1, 3, 0.18, -0.5, 0.1, 0, 0, 0, 0.8, 0.02, 0.18, 0, 0, 0, 3.25]$$



Figura 4.36. Imagen prototipo de la base de datos de imágenes

4.4.3 Base de datos de vectores (MRV)

Finalmente, para el Método de Reconocimiento por Vectores se elaboró una base de datos compuesta por vectores de siete elementos, cuya estructura está dada por la extracción de características utilizada por el mismo método. Esta base de datos al igual que las anteriores es no relacional y tiene por etiquetas las letras a, e, i, o, u, c, d, h, n, r, s, t, v, y como datos asociados, vectores de siete elementos con modelos de datos en atributos.

En el siguiente diagrama a bloques se muestra el proceso de formación de la base de datos.

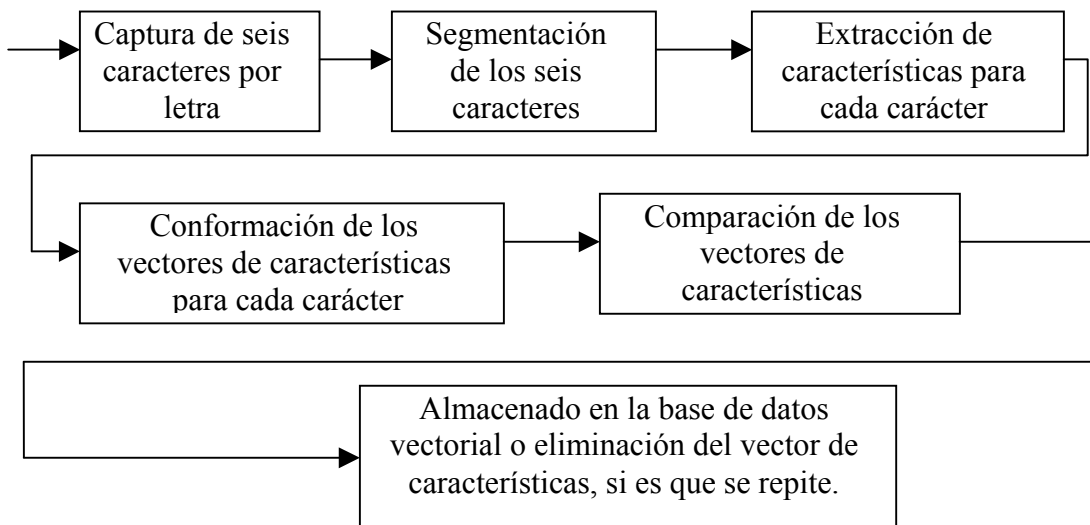


Figura 4.37. Diagrama del proceso de obtención de la base de datos de vectores

Para crear esta base primero se elaboró una interfaz de captura para seis caracteres por letra, como se muestra en la Figura 4.38.

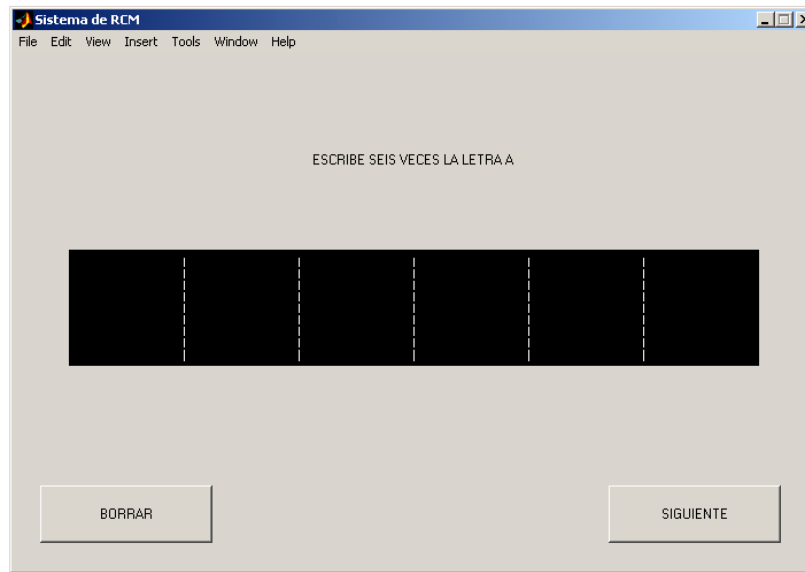


Figura 4.38. Interfaz de captura de caracteres para la base de datos de vectores

Utilizando la interfaz, se pide al autor que escriba seis veces la letra “a”, respetando los espacios marcados para cada carácter, y siguiendo el ejemplo de escritura mostrado en la Figura 4.39.

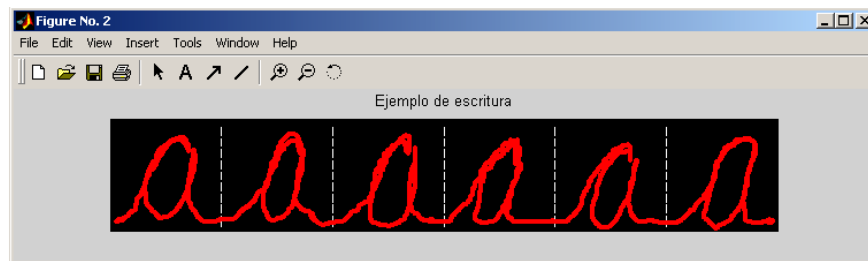


Figura 4.39. Ejemplo de escritura de la letra a

Una vez capturadas las seis letras “a”, al presionar el botón SIGUIENTE se llama al programa que segmenta la palabra para obtener seis caracteres aislados. Una vez obtenida dicha segmentación, el sistema calcula para cada carácter, un vector conteniendo los siete parámetros establecidos por la extracción para el Método de Vectores. Dichos vectores se guardan, siempre y cuando sean distintos entre sí y entre los previamente guardados. Este proceso se repite para cada una de las trece letras seleccionadas, guardando para cada letra un vector o conjunto de vectores diferentes. Cada uno de los siete parámetros fue definido anteriormente en la sección de extracción de características. Estos parámetros se señalan a continuación:

1. Ancho horizontal del carácter
2. Ancho vertical del carácter
3. Altura del carácter
4. Altura del punto final del carácter
5. Ancho a la mitad horizontal del carácter
6. Carácter abierto o cerrado en la parte inferior

7. Ancho alrededor del punto máximo

En la elaboración de esta base de datos participaron veinte personas, por lo que se obtuvo para cada letra un conjunto de veinte vectores, para representar la escritura de cada persona. Finalmente, solo se consideran aquellos que difieren entre sí, de esta manera para cada carácter la base de datos cuenta con más de un vector representativo de las variaciones de la escritura entre esas 20 personas. Dichos vectores fueron los que se utilizaron para las comparaciones correspondientes en el método de reconocimiento por vectores.

Como se mencionó al inicio de esta sección, después de la extracción de características y la obtención de la base de datos, puede ya realizarse el siguiente proceso: la comparación y reconocimiento, cuyo procedimiento se explicará en el Capítulo 5.

Capítulo 5

Comparación y Reconocimiento

Una vez que se ha extraído de los caracteres la información necesaria y se han creado las bases de datos, es posible aplicar los algoritmos desarrollados para la comparación y el reconocimiento de la escritura.

Los procesos de comparación y reconocimiento están estrechamente relacionados, de tal forma que el resultado de la comparación es casi prácticamente el reconocimiento. En esta sección se explicarán estos procesos, indicando la relación que guardan entre sí. En la figura 5, puede ubicarse este bloque dentro del sistema.

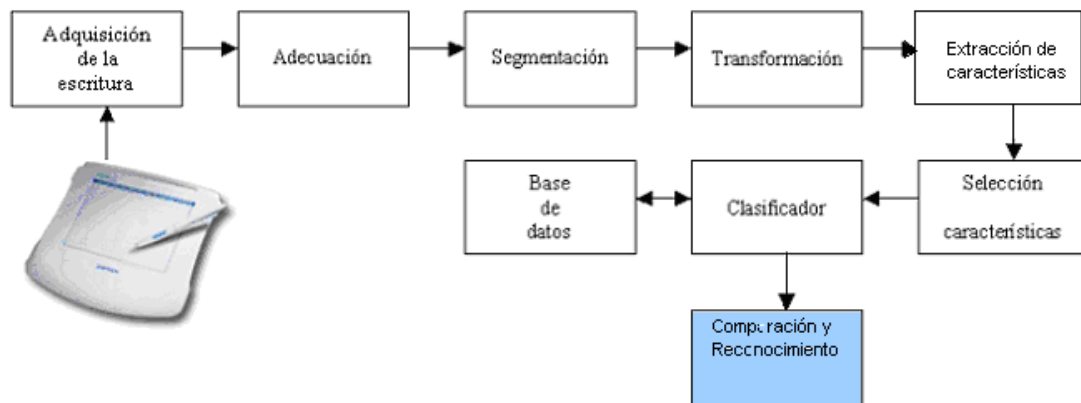


Figura 5.1 Ubicación en el sistema del bloque de comparación y reconocimiento

Los tipos de reconocimiento que el sistema contempla son dos, los cuales el usuario puede elegir a través de la interfaz: el reconocimiento de caracteres aislados y el reconocimiento de palabras. Aunque este último en realidad se basa en el reconocimiento de los caracteres aislados de la palabra, obtenidos mediante el proceso de segmentación.

5.1 Reconocimiento de caracteres aislados

Para caracteres, el sistema brinda al usuario cuatro métodos de reconocimiento. Los primeros tres, el Reconocimiento por Envolventes, el Reconocimiento por Trayectorias y el Reconocimiento

por Trayectorias Optimizado son para vocales aisladas, mientras que el cuarto, el Reconocimiento por Vectores, es útil tanto para letras aisladas como para palabras. Cada uno de estos métodos será explicado a continuación, comenzando con el método de Comparación y Reconocimiento por Envoltentes.

5.1.1 Comparación y reconocimiento por envoltentes (RE)

Comparación

Una vez que se han obtenido los vectores característicos del carácter prueba por el método de Extracción de Características de Envoltentes, serán comparados con los vectores prototipo. El vector obtenido para el carácter de prueba, se compara con cada uno de los vectores que se obtienen de la base de datos de imágenes de los caracteres prototipo. Esta comparación se realiza tanto para los vectores de la envoltente superior, como para los de la envoltente inferior de cada carácter. Para comparar se utilizan los primeros 200 elementos de los vectores, que corresponden a las envoltentes. El último elemento, el promedio de cortes, será utilizado más adelante.

En primer lugar se obtiene la diferencia entre cada envoltente superior prototipo “*Vespt*” y la de prueba superior “*Veprus*”, formando el vector diferencia superior para cada prototipo, “*Vedifs*”.

$$Vedifs = Vespt - Veprus$$

El vector diferencia “*Vedifs*” contendrá entonces cinco elementos, donde cada uno de ellos corresponde a la diferencia entre el vector de cada carácter, de cinco que el sistema puede reconocer de esta manera, y el carácter de prueba. Un ejemplo de un vector “*Vedifs*” obtenido por este proceso es el siguiente:

$$Vedifs = [1500 \ 3234 \ 2345 \ 1845 \ 3456]$$

El vector de diferencias mostrado anteriormente es el resultado de la comparación entre las envoltentes superiores de cada uno de los caracteres prototipo y la del carácter prueba. En él se puede observar que el valor mínimo del vector corresponde al elemento número uno. Mediante un barrido de este vector se encuentra ese valor mínimo y se designa su número de elemento dentro del vector, al carácter de prueba, como un pre-reconocimiento. Ahora debemos comparar a las envoltentes inferiores y a los valores de corte promedio.

Para obtener el vector de diferencias correspondiente a las envoltentes inferiores se procede de la misma manera que para las superiores, el vector que en este caso se generará es el de diferencias inferiores “*Vedifi*”, con la misma estructura que el superior de cinco elementos. Un ejemplo de este vector obtenido es el siguiente:

$$Vedifi = [1342 \ 1945 \ 2435 \ 1355 \ 1301]$$

En el vector anterior, podemos observar que para varios elementos las diferencias son muy semejantes y que aunque el carácter de prueba es una letra “a”, este pre-reconocimiento arrojaría como carácter reconocido a la letra “u”. Esto se genera debido a que las envoltentes inferiores para determinadas letras son muy similares. Sin embargo, al realizar una suma entre los vectores de diferencias superior e inferior, dando una ponderación mayor al de diferencias superiores (debido a

que ese resultado es mejor en cuanto a reconocimiento correcto), se minimiza la posibilidad de error. Ahora se obtiene un nuevo vector, el de diferencias de envolventes, "**Vdif**".

$$\text{Vdif} = 2 * \text{Vdifs} + \text{Vdifi}$$

Del ejemplo que se está considerando, de un carácter de prueba "a", el sistema obtuvo el siguiente vector "**Vdif**":

$$\text{Vdif} = [4342 \quad 8413 \quad 7125 \quad 5045 \quad 8213]$$

Posteriormente "**Vdif**" se suma con el vector de diferencias de cortes "**Vdifcor**", obtenido de la diferencia entre el promedio de cortes de los prototipos y el del carácter de prueba. Luego de esa suma se obtiene el promedio, guardando el resultado en un vector de diferencias totales llamado "**Vdift**".

$$\text{Vdifcor} = [1500 \quad 3500 \quad 6700 \quad 1800 \quad 8900]$$

$$\text{Vdift} = (\text{Vdif} + \text{Vdifcor}) / 2$$

Del mismo ejemplo que se ha estado considerando, el del carácter "a", se obtiene el siguiente vector "**Vdift**"

$$\text{Vdift} = [1461 \quad 2978 \quad 3456 \quad 1711 \quad 4278]$$

Como es de esperarse, el valor más pequeño de este vector representa al carácter prototipo con el cual el carácter prueba tiene menos diferencias. Al multiplicar por un factor de 2 a las diferencias de la envolvente superior le estamos dando un mayor peso en el proceso, debido a que estas envolventes tienen diferencias más significativas entre todos los caracteres, respecto a las de la envolvente inferior.

Observamos que para la letra de prueba "a" que se está analizando, se obtiene que la menor de las cinco diferencias obtenidas para la envolvente superior es igual a **1500**, lo cual identifica a la prueba como una letra "a", mientras que la menor de las cinco diferencias obtenidas para la envolvente inferior es igual a **1301**, lo que identifica a la prueba como una letra "u"; observando esto, entendemos por qué este factor ayuda a reconocer mejor al carácter de prueba.

Reconocimiento

Finalmente, para encontrar la identidad del carácter de prueba, se realiza un recorrido por el vector de diferencias totales, encontrando el número del elemento con la mínima diferencia, y ese número es asignado como el número del carácter reconocido. En este caso ese número corresponde a la posición uno, y el carácter reconocido es la letra a. Sin embargo, al realizar pruebas con esta consideración, se hizo evidente que en algunos casos la confusión de la identidad del carácter de prueba requería considerar además otras características para elevar la tasa de reconocimiento correcto.

Las principales confusiones entre caracteres se presentaron entre las letras "e" e "i" y las letras "a" y "o". Las características que ayudan al sistema a diferenciar estos caracteres son la altura final

del trazo y el ancho del carácter. La identidad de los caracteres en cuestión se decide entonces por medio de un árbol de decisiones, como se explica a continuación.

Las alturas que nos interesan en este caso son las alturas finales del trazo de la prueba y de las letras “a” y “o”. Una vez que se ha identificado al carácter de prueba como una “o” o como una “a”, utilizamos el siguiente criterio para diferenciarlos. Se obtiene el valor de la altura final del carácter prueba, y si es menor que un valor de altura medio (obtenido por el análisis de las alturas de los prototipos), que en este caso es de 10, entonces se reconoce al carácter como una letra “a”, de lo contrario el carácter será reconocido como una letra “o”.

Por otro lado, para distinguir entre una letra “e” de una “i” su ancho es la característica que nos ayudará a diferenciarlas. El ancho de los caracteres se obtiene del proceso de extracción de características explicado en el Capítulo 4. Al obtener los anchos del carácter de prueba tenemos la siguiente regla para su diferenciación. Una vez que se ha identificado al carácter de prueba como una letra “e” o como una “i”, si el ancho es menor a 10, el carácter reconocido será una “i”, de lo contrario será una “e”.

En el ejemplo que se ha analizado, se identificó al carácter como una letra “a”. Al entrar al condicional mencionado se obtiene el siguiente resultado:

$$hp = 57$$

por lo que el carácter reconocido es una letra “o”, ya que su valor es mayor que el valor medio de altura, y esa característica la presenta el carácter “o”.

5.1.2 Comparación y reconocimiento por trayectorias (RT)

Comparación

Todo el conjunto de características contenidas en los vectores característicos nos permiten la diferenciación entre caracteres, pues como resultado de cada proceso de extracción obtenemos un vector completamente distinto para cada prototipo, y su comparación con el vector resultante de la extracción de características para la prueba, arroja el número del vector que representa al carácter reconocido.

El proceso de comparación consiste en hacer un recorrido por los vectores obtenidos a partir de las imágenes de los prototipos de la base de datos, encontrando el vector que sea igual o tenga la menor variación respecto al obtenido para el carácter prueba. Para obtener este valor se realiza una diferencia absoluta entre el vector obtenido para el carácter prueba y cada uno de los vectores obtenidos para los caracteres prototipos.

Reconocimiento

El vector prototipo que arroje el resultado menor para la diferencia anteriormente mencionada es interpretado como el carácter más parecido a la prueba, y ese es el asignado como carácter reconocido.

En el proceso anterior, se obtiene un reconocimiento previo que hay que verificar, debido a varios factores, básicamente los siguientes: que dos o más letras tengan un vector representativo muy parecido, la imposibilidad de considerar todos los estilos de escritura en la base de datos, a los errores en el proceso de segmentación y a los errores esperados a causa de que el autor del carácter haya cometido a propósito o por descuido un error en el trazo que no corrigió. El proceso de verificación consiste en la extracción de ciertas características de la imagen, con las cuales se realiza, basándose en su ordenamiento, un árbol de decisiones para el reconocimiento. Las características para el proceso de verificación son:

1. Ancho medio del carácter: es el ancho horizontal del carácter en la zona media.
2. Altura final: es la altura del último pixel del enlace final del carácter.

Los errores mencionados anteriormente ocasionan en la etapa de comparación equivocaciones en el reconocimiento debido a que se modifican los valores de los vectores característicos, ocasionando principalmente confusiones entre ciertos caracteres que difieren entre ellos por uno o dos elementos solamente, lo cual se corregirá en el proceso de verificación. También surgen errores en los que la mayoría de los elementos del vector difieren y en este caso, habrá que repetir el trazo.

En la verificación si un carácter es reconocido como "a", habrá que verificar que no se trate de una "o". Lo mismo sucede con las letras "e" e "i". La verificación se realiza considerando el árbol de decisiones, mostrado en la Figura 5.2.

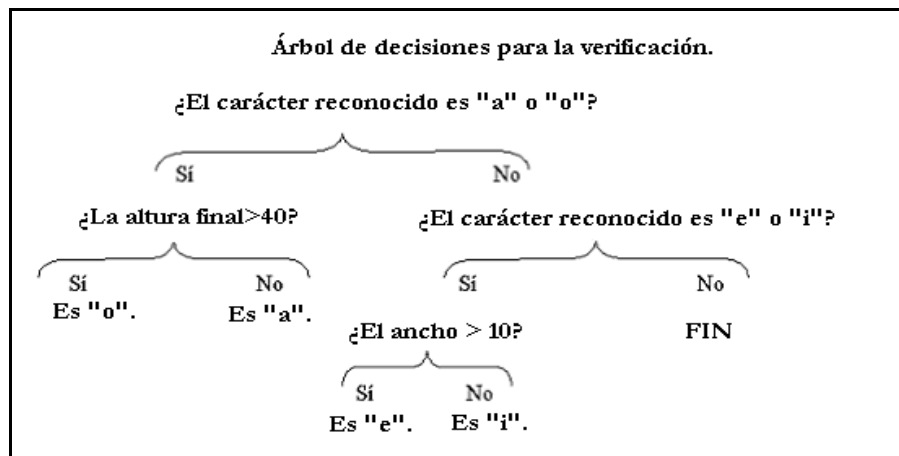


Figura 5.2. Árbol de decisiones para el método por trayectorias

Como podemos observar, del proceso de verificación se obtiene el reconocimiento del carácter prueba; es así como se verifica el pre-reconocimiento, corrigiéndolo en caso de los errores mencionados o confirmando la identidad del carácter pre-reconocido. Como ejemplo de este método de reconocimiento de caracteres, tenemos la imagen del carácter de prueba "a" en la Figura 5.3, el cual será reconocido por este método.

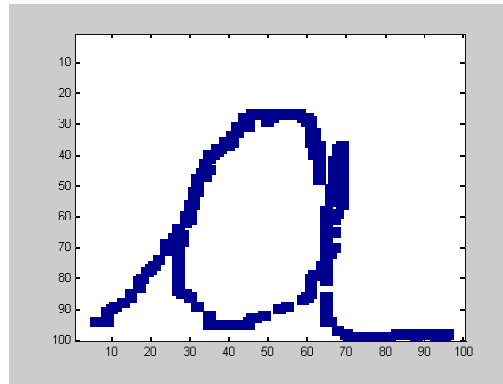


Figura 5.3 Carácter de prueba para el reconocimiento por trayectorias

Con las características extraídas para el carácter anterior, se obtuvo el siguiente vector:

$$v = [1, 1, 3, 0.18, -0.5, 0.1, 0, 0, 0, 0.8, 0.02, 0.18, 0, 0, 0, 3.25]$$

que, como recordamos, tiene la siguiente estructura:

$$v = [\text{ini}, \text{fin}, \text{trayec}, m(1), m(2), m(3), m(4), m(5), m(6), \text{lt}(1), \text{lt}(2), \text{lt}(3), \text{lt}(4), \text{lt}(5), \text{lt}(6), \text{cor}]$$

La comparación para este proceso de reconocimiento, realiza un recorrido por los vectores obtenidos de la base de datos. Esos vectores son obtenidos por la extracción de las mismas características, para los caracteres prototipo y no para los caracteres prueba. Como resultado del proceso de comparación el algoritmo encuentra aquel vector que sea igual o varíe menos con respecto al vector “v” de la prueba.

La base de datos está conformada como una matriz, en la que cada renglón es un vector que guarda las características extraídas para cada una de los caracteres prueba, en forma similar a lo siguiente:

Vector(1, :) = [ini, fin, trayec, m(1), m(2), m(3), m(4), m(5), m(6), lt(1), lt(2), lt(3), lt(4), lt(5), lt(6), cor] %a

Vector(2, :) = [ini, fin, trayec, m(1), m(2), m(3), m(4), m(5), m(6), lt(1), lt(2), lt(3), lt(4), lt(5), lt(6), cor] %e

Vector(3, :) = [ini, fin, trayec, m(1), m(2), m(3), m(4), m(5), m(6), lt(1), lt(2), lt(3), lt(4), lt(5), lt(6), cor] %i

Vector(4, :) = [ini, fin, trayec, m(1), m(2), m(3), m(4), m(5), m(6), lt(1), lt(2), lt(3), lt(4), lt(5), lt(6), cor] %o

Vector(5, :) = [ini, fin, trayec, m(1), m(2), m(3), m(4), m(5), m(6), lt(1), lt(2), lt(3), lt(4), lt(5), lt(6), cor] %u

Al recorrer esta matriz buscando el vector cuya diferencia con respecto al vector obtenido para la prueba “v”, sea la mínima, se obtiene un vector de diferencias a partir de las siguientes ecuaciones:

$$\text{Dif}(1,1) = \text{sum}(\text{Vector}(1,:)-v)'$$

$$\text{Dif}(2,1) = \text{sum}(\text{Vector}(2,:)-v)'$$

$$\text{Dif}(3,1) = \text{sum}(\text{Vector}(3,:)-v)'$$

$$\text{Dif}(4,1) = \text{sum}(\text{Vector}(4,:)-v)'$$

$$\text{Dif}(5,1) = \text{sum}(\text{Vector}(5,:)-v)'$$

El vector de diferencias obtenido tiene una forma similar a la siguiente:

$$\text{Dif} = [1 \ 5 \ 6 \ 2 \ 4]$$

Encontrando el elemento mínimo dentro de ese vector de diferencias, el algoritmo puede asignar el carácter reconocido. En este caso se reconocería a un carácter “a”.

Nuevamente introducimos un proceso de verificación, con un árbol de decisiones como en el reconocimiento por envolventes.

Una vez que el sistema identifica al carácter de prueba como un carácter “a”, sigue un ciclo de diferenciación entre “a” y “o” y entre “e” e “i”. En este ciclo se establecen las mismas características que en el caso previo:

1. Si el ancho del carácter es menor a 10 píxeles, el carácter es una “i”.
2. Si su altura final es mayor de 40 píxeles es una “o”.

En este caso, dado que la altura final del carácter es 27, el carácter es verificado efectivamente como una letra a.

5.1.2.1 Reconocimiento por trayectorias optimizado (RTO)

El método de reconocimiento por trayectorias, utiliza una base de datos formada por el conjunto de las imágenes de los caracteres prototipo. Cuando el reconocimiento se realiza, todas esas imágenes son leídas por el bloque de extracción de características, junto con la imagen del carácter capturado. Lo anterior con la finalidad de encontrar los vectores tanto de los prototipos, como el de la prueba, y compararlos.

Sin embargo, las imágenes de los prototipos son constantes y la única imagen que cambia es la de la prueba, por lo que obteniendo una base de datos de vectores, a partir de la extracción de características de las imágenes prototipo guardadas en la base de datos de imágenes, podremos realizar en cada reconocimiento sólo la lectura y análisis de una sola imagen (la de la prueba) y recurrir a una base de datos de vectores para la comparación, evitando así, la repetición innecesaria del proceso de extracción para cada carácter prototipo. Esta es precisamente la optimización hecha al reconocimiento por trayectorias, obteniendo con ello, un reconocimiento optimizado en cuanto a tiempo y repetición de procesos.

5.1.3 Comparación y reconocimiento por vectores (RV)

Este método de reconocimiento, utiliza una serie de características que son determinantes en la escritura, tales como el ancho del carácter, la altura, los puntos máximos, mínimos y medios, la

presencia o ausencia de ciertos trazos, así como las áreas que encierran los trazos de las letras. Como podremos observar a continuación, las siete características contenidas en estos vectores permiten la diferenciación entre caracteres, pues como resultado de cada proceso de extracción de características en la base de datos se obtiene un vector o un conjunto de vectores específicos para cada carácter. La comparación entre el vector resultante del proceso de extracción para la imagen de la letra a reconocer (prueba), con la base de datos creada para este reconocimiento, arroja al carácter reconocido.

Como ya se ha explicado anteriormente, el formato de los vectores obtenidos para este tipo de reconocimiento es el siguiente:

$$\text{Vectorp} = [\text{AH AV H PF AM AC APM}]$$

Vectorp = [Ancho Horizontal, Ancho Vertical, Altura, Posición Final, Ancho Medio, Abierto o Cerrado, Ancho del Máximo]

Todos los vectores de la base de datos se guardan en una matriz “vc”, que se muestra en la Tabla 5.1.

CARÁCTER	VECTORES	CARÁCTER	VECTORES
a	vc(1,:)= [2 2 0 0 2 1 2] vc(2,:)= [1 2 0 0 2 1 2] vc(3,:)= [2 2 0 1 2 0 1]	h	vc(45,:)= [2 1 1 0 1 0 2] vc(46,:)= [2 2 1 0 1 0 2] vc(47,:)= [2 2 1 0 2 0 2] vc(48,:)= [1 2 1 0 1 0 2] vc(49,:)= [1 1 1 0 1 0 2] vc(50,:)= [1 0 1 0 2 0 2] vc(51,:)= [2 0 1 0 2 0 2] vc(52,:)= [2 1 1 0 2 0 2]
e	vc(4,:)= [2 2 0 0 1 1 2] vc(5,:)= [2 2 0 0 1 1 1] vc(6,:)= [1 2 0 0 1 1 2] vc(7,:)= [2 2 0 0 0 1 2] vc(8,:)= [0 2 0 0 1 1 2] vc(9,:)= [2 2 0 0 0 0 2] vc(10,:)= [1 2 0 0 1 1 1] vc(11,:)= [2 2 0 0 1 1 0] vc(12,:)= [1 1 0 0 1 1 1] vc(13,:)= [2 1 0 0 1 1 2] vc(14,:)= [1 1 0 0 1 1 2]	n	vc(53,:)= [2 1 0 0 2 0 1] vc(54,:)= [2 0 0 0 2 0 2] vc(55,:)= [2 2 0 0 2 0 2] vc(56,:)= [2 1 0 0 2 0 2] vc(57,:)= [2 0 0 0 2 0 1] vc(58,:)= [2 2 0 0 2 0 1] vc(59,:)= [2 1 0 0 1 0 1] vc(60,:)= [1 0 0 0 2 0 2] vc(61,:)= [1 1 0 0 0 0 2] vc(62,:)= [1 1 0 0 2 0 2] vc(63,:)= [2 1 0 1 2 0 2] vc(64,:)= [2 1 0 0 1 0 2] vc(65,:)= [2 0 0 0 1 0 2] vc(66,:)= [2 0 0 1 2 0 2] vc(67,:)= [2 0 0 0 2 0 0]
i	vc(11,:)= [0 1 0 0 0 1 0] vc(12,:)= [0 2 0 0 0 1 0]	r	vc(54,:)= [1 0 0 1 1 0 0] vc(55,:)= [1 0 0 1 0 0 0]
o	vc(22,:)= [2 2 0 1 2 1 2] vc(23,:)= [2 2 0 1 2 0 2] vc(24,:)= [1 2 0 1 2 1 2] vc(25,:)= [2 1 0 1 2 1 2] vc(26,:)= [2 2 0 1 0 1 2]	s	vc(88,:)= [0 2 0 0 1 1 0] vc(89,:)= [1 2 0 0 1 1 0] vc(90,:)= [0 1 0 0 1 1 0] vc(91,:)= [2 2 0 0 2 1 0] vc(92,:)= [0 2 0 0 2 1 0] vc(93,:)= [1 2 0 0 0 1 0] vc(94,:)= [1 2 0 0 2 1 0] vc(95,:)= [1 2 0 0 1 1 0] vc(96,:)= [0 2 0 0 1 1 1] vc(97,:)= [0 2 0 0 2 1 2] vc(98,:)= [1 2 0 0 2 1 2] vc(99,:)= [0 2 0 0 2 1 1] vc(100,:)= [0 1 0 0 2 1 0] vc(101,:)= [0 2 0 0 1 1 1] vc(102,:)= [0 2 0 0 2 1 2] vc(103,:)= [1 2 0 0 2 1 2]

			vc(104,:)= [0 2 0 0 2 1 1] vc(105,:)= [0 1 0 0 2 1 0] vc(106,:)= [1 2 0 0 2 1 1]
u	vc(27,:)= [2 0 0 0 2 1 0] vc(28,:)= [2 1 0 0 1 1 0] vc(29,:)= [2 1 0 0 2 1 0] vc(30,:)= [1 1 0 0 1 1 0] vc(31,:)= [1 1 0 0 2 1 0] vc(32,:)= [1 0 0 0 2 1 0] vc(33,:)= [1 0 0 0 1 1 0] vc(34,:)= [1 1 0 1 2 1 0] vc(35,:)= [2 0 0 0 1 1 0]	t	vc(107,:)= [0 2 1 0 0 1 2] vc(108,:)= [1 2 1 0 0 1 2] vc(109,:)= [0 2 1 0 1 1 2] vc(110,:)= [0 1 1 0 0 1 2] vc(111,:)= [0 2 1 1 2 1 2] vc(112,:)= [1 2 1 1 2 1 2]t
c	vc(36,:)= [1 2 0 0 0 1 2] vc(37,:)= [0 2 0 0 0 1 1] vc(38,:)= [0 2 0 0 0 1 2] vc(39,:)= [0 2 0 1 2 1 2]	v	vc(113,:)= [2 0 0 1 1 0 0] vc(114,:)= [2 0 0 1 1 1 0] vc(115,:)= [2 0 0 1 2 1 0] vc(116,:)= [2 0 0 1 2 0 0] vc(117,:)= [1 0 0 1 2 1 0] vc(118,:)= [2 0 0 1 0 0 0]
d	vc(40,:)= [2 2 1 0 2 1 2] vc(41,:)= [2 2 1 0 1 1 2] vc(42,:)= [1 2 1 0 2 1 2] vc(43,:)= [2 2 1 1 2 1 2] vc(44,:)= [2 2 1 0 2 1 0]		

Tabla 5.1. Base de datos de vectores

Una vez obtenido el vector de características para el carácter de prueba, el proceso de comparación consiste en hacer un recorrido por los vectores de la base de datos, encontrando el vector que sea igual al obtenido para el carácter prueba. Cada carácter en la base de datos está representado por un conjunto de vectores, quienes engloban todas las posibles variaciones en la escritura del carácter, con un determinado estilo. El vector para el carácter de prueba obtenido para este caso, fue el siguiente:

$$vp = [2 \ 2 \ 0 \ 1 \ 2 \ 0 \ 1]$$

Reconocimiento

Realizando el recorrido por la base de datos se encuentra que este vector corresponde al elemento tres, que lo relaciona con el carácter “a”. El resultado, entonces de este proceso será el reconocimiento de la letra “a”.

Cuando el proceso arroja un vector que no se encuentra en la base de datos, se considera a ese carácter como erróneo o desconocido, ya sea por descuido del autor o por representar al carácter en un estilo no considerado en la base de datos. En la interfaz, esto se representa con un signo de interrogación. Mientras más estilos se manejen, será necesario incluir en la base de datos más vectores característicos.

5.2 Reconocimiento de palabras

Es conveniente recalcar que el reconocimiento de caracteres aislados, difiere del reconocimiento de palabras, debido a que para este último caso se involucra además la segmentación y el centrado del carácter en el área de reconocimiento. Como ya se ha mencionado con anterioridad, el reconocimiento de una palabra involucra la segmentación de la misma en su mínima unidad de reconocimiento, es decir, en caracteres. El reconocimiento de palabras hecho por el sistema consiste

en el reconocimiento de cada uno de los caracteres que la forman, de tal manera que al finalizar el reconocimiento de todos los caracteres, la palabra haya sido reconocida.

Una vez realizada la segmentación de la palabra en caracteres aislados, se extraen sus características, para posteriormente entrar al bloque de comparación y reconocimiento. El método que utiliza el sistema para reconocer los caracteres provenientes de las palabras, es el Reconocimiento por Vectores (RV), debido a que al realizar las pruebas de validación para caracteres aislados, este reconocimiento obtuvo la eficiencia más alta, como se explica en el Capítulo 7.

Pero si nos preguntamos cuál es la diferencia exacta en el proceso de reconocimiento entre caracteres aislados y los que forman parte de una palabra, la respuesta la encontraremos en los enlaces y puntos de corte presentes en las letras provenientes de la segmentación de la palabra. Si el resultado del proceso de segmentación muestra al carácter con un cierto desplazamiento en su punto de corte, esto ocasionará que muchas de las características que se extraen del carácter varíen de tal forma que se produzcan confusiones en el reconocimiento, al no haberse considerado tal corrimiento.

De esta manera, si los algoritmos de reconocimiento se aplican directamente con las características obtenidas para los caracteres aislados, sin considerar corrimientos, ni otras variantes, se generará un error en el reconocimiento. Por esto es importante considerar los efectos de la segmentación en los procesos posteriores. Para ello es necesario analizar el proceso de segmentación para conocer de una mejor manera sus resultados y poderlos considerar más adelante.

También podemos observar que los puntos de corte de enlaces por la segmentación, pueden estar presentes tanto en la parte inferior como en la parte superior para una misma letra, dependiendo de la letra con la que se enlace al formar la palabra. Ciertos parámetros de una letra se modifican al cambiar la letra adjunta con la que se enlaza. Esto debe ser considerado en el diseño de las estrategias de extracción, comparación y reconocimiento. Lo anterior se ilustra en las Figuras 5.4 y 5.5, donde se muestra la palabra “dos” formada por caracteres sueltos y por caracteres enlazados.

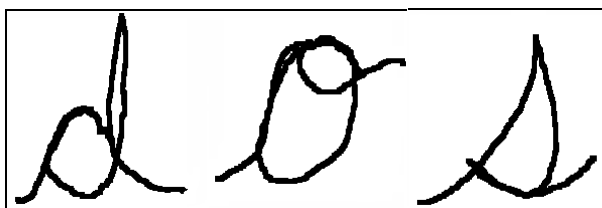


Figura 5.4. La palabra “dos” formada por caracteres sueltos

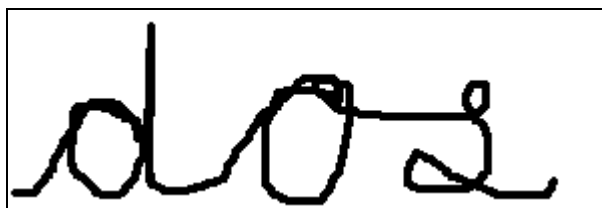


Figura 5.5. La palabra “dos” formada por caracteres enlazados

Se observa que el parámetro que define la posición inicial correspondiente a la letra “s”, varía dependiendo de si se escribe en forma enlazada o no, por lo que el reconocimiento deberá considerarlo. Al observar ambas formas de escribir la misma palabra con caracteres sueltos y enlazados, respectivamente, podemos darnos cuenta de que la letra “s” es la que presenta una mayor diferencia; cuando se trata de caracteres aislados, su enlace inicial empieza en la parte baja, mientras que al formar parte de la palabra dicho enlace comienza en la parte superior.

Además, podemos apreciar la diferencia de tamaño que existe entre la letra “d”, que ocupa todo el alto del espacio disponible de escritura, y los caracteres bajos, como la “s” y la “o”, que son letras más pequeñas que ocupan máximo tres cuartas partes de la altura total de la zona de escritura. En algunos otros casos, cuando no se respete esta diferencia en alturas entre caracteres bajos y altos, se presentará también un error en el reconocimiento.

Con estos ejemplos, se comprende que no se trata solamente de segmentar y reconocer cada carácter de una palabra, sino también de adecuar los algoritmos de reconocimiento con los de segmentación y extracción para considerar todas las posibles variaciones de los caracteres dentro de una misma palabra y en todas las palabras del diccionario. Como ya se ha mencionado, el algoritmo de reconocimiento que el sistema emplea para el reconocimiento de palabras es el Reconocimiento por Vectores (RV), que es utilizado también para el reconocimiento de la totalidad de caracteres aislados del diccionario. Este proceso de reconocimiento se explicará a continuación, ahora para el caso de palabras.

5.2.1 Reconocimiento de palabras utilizando RV

Una vez que la palabra a reconocer ha sido segmentada, cada carácter aislado pasa a este bloque para ser reconocido y ordenado de tal forma que el despliegue del reconocimiento de cada uno de ellos, corresponda al reconocimiento total de la palabra.

Vamos a explicar ahora el reconocimiento por vectores, cuyos caracteres a reconocer provienen del método de segmentación por localización de extremos superiores. En la Figura 5.6 tenemos la palabra “uno”, la cual ha sido segmentada por dicho método.

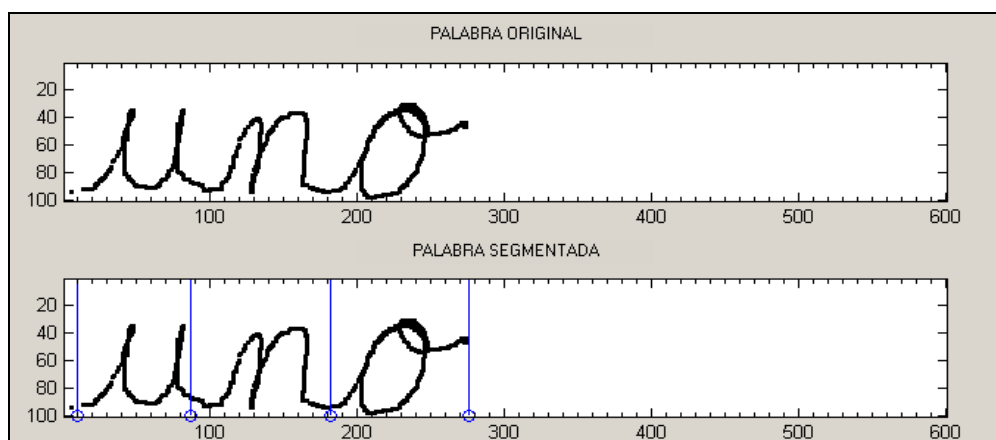


Figura 5.6. Segmentación por localización de extremos superiores

Después de segmentar y centrar, se tienen en la Figura 5.7, las imágenes de los tres caracteres que serán reconocidos individualmente.

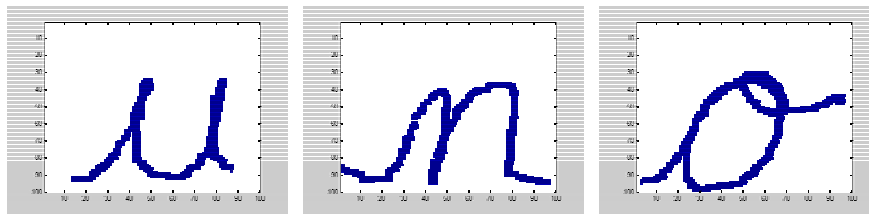


Figura 5.7. Caracteres a reconocer

Una vez que realizamos pruebas de reconocimiento para palabras con el algoritmo de reconocimiento por vectores, sin considerar las variaciones en los enlaces y puntos de corte para caracteres, encontramos una tasa menor de reconocimiento correcto, debido, como ya lo explicamos, a las variaciones de los caracteres dentro de una palabra. Por esta razón, necesitamos considerar esas variaciones, lo que nos ayudará a mejorar la eficiencia del reconocimiento.

En la base de datos se almacenan otros vectores de 7 elementos que incluyen las variaciones mencionadas de corrimiento del punto de corte y los enlaces. Al considerarlas se incrementó el número de vectores de características asociados a cada carácter. La estructura de estos vectores sigue siendo la misma, recordando que sus características son las siguientes:

1. Ancho horizontal
2. Ancho vertical
3. Carácter alto o bajo
4. Punto final arriba o abajo
5. Ancho a la mitad horizontal del carácter
6. Carácter abierto o cerrado en la parte inferior
7. Ancho alrededor del punto máximo

Con estas siete características se evitan las confusiones entre caracteres, además, se agregan en la base de datos más vectores para un mismo carácter, que puede tener asignados varios de estos vectores característicos, en los cuales se registran las variaciones de un mismo carácter al escribirse dentro de una palabra, con lo que se logra que este método sea tolerante a variaciones en la escritura.

En la Tabla 5.2 se muestran algunos de los vectores característicos obtenidos mediante la base de datos, para ilustrar los siguientes ejemplos.

CARÁCTER	VECTORES
	CARACTERÍSTICOS
a	vc(1,:)= [2 2 0 0 2 1 2] vc(2,:)= [2 2 1 1 2 1 2]
e	vc(5,:)= [2 2 0 0 1 1 2] vc(6,:)= [2 2 0 0 1 1 1]
i	vc(11,:)= [0 1 0 0 0 1 0] vc(12,:)= [0 2 0 0 0 1 0]
o	vc(16,:)= [2 2 0 1 2 1 2] vc(17,:)= [2 2 0 1 2 0 2]
u	vc(22,:)= [2 1 0 0 1 1 0] vc(23,:)= [2 1 0 0 2 1 0]
c	vc(30,:)= [1 2 0 0 0 1 2] vc(31,:)= [0 2 0 0 0 1 1]
d	vc(35,:)= [2 2 1 0 2 1 2] vc(36,:)= [2 2 1 0 1 1 2]
h	vc(40,:)= [2 1 1 0 1 0 2] vc(41,:)= [2 2 1 0 1 0 2]
n	vc(49,:)= [2 2 0 0 2 0 2] vc(50,:)= [2 1 0 0 2 0 2]
r	vc(54,:)= [1 0 0 1 1 0 0] vc(55,:)= [1 0 0 1 0 0 0]
s	vc(64,:)= [0 2 0 0 1 1 0] vc(65,:)= [0 1 0 0 1 1 0]
t	vc(69,:)= [0 2 1 0 0 1 2] vc(70,:)= [1 2 1 0 0 1 2]
v	vc(74,:)= [2 0 0 1 1 0 0] vc(75,:)= [2 0 0 1 1 1 0]

Tabla 5.2. Algunos vectores característicos de la base de datos

Para explicar el método de reconocimiento por vectores, vamos a considerar el caso de la palabra de prueba “uno” que hemos analizado en las secciones anteriores. Una vez realizada la segmentación se obtiene a los caracteres u, n y o aislados. El paso siguiente es extraer sus características individuales para obtener los vectores de características. El reconocimiento en sí consiste en comparar el vector característico del carácter de prueba, con cada uno de los vectores característicos de la base de datos, hasta encontrar el vector con el cual es idéntico, asignando a la prueba el número de dicho vector, reconociendo así al carácter. En caso de que el vector de la prueba no coincida con ninguno de la base de datos, se dice que el carácter no ha sido reconocido.

En el ejemplo de la palabra, el primer carácter de prueba generó el vector característico [2 1 0 0 2 1 0], y al realizar la comparación con la Tabla 5.2, el vector con el que coincide es con el número 23, lo cual lo identifica como una letra “u”.

Para el segundo carácter de prueba, se obtuvo el vector [2 1 0 0 2 0 2], mientras que el resultado de la comparación lo identifica con el vector 50, que representa a un carácter n.

Finalmente, el último carácter de prueba generó el vector [2 2 0 1 2 1 2], que coincide con el vector 16 de dicha tabla, por lo que es reconocido como una letra o. Cada uno de estos resultados se expresa como una imagen del carácter reconocido en una fuente de escritura dentro de la interfaz

diseñada para tal fin, colocándose en su orden de aparición en la palabra, representando a la palabra reconocida, tal como se muestra en la Figura 5.8.

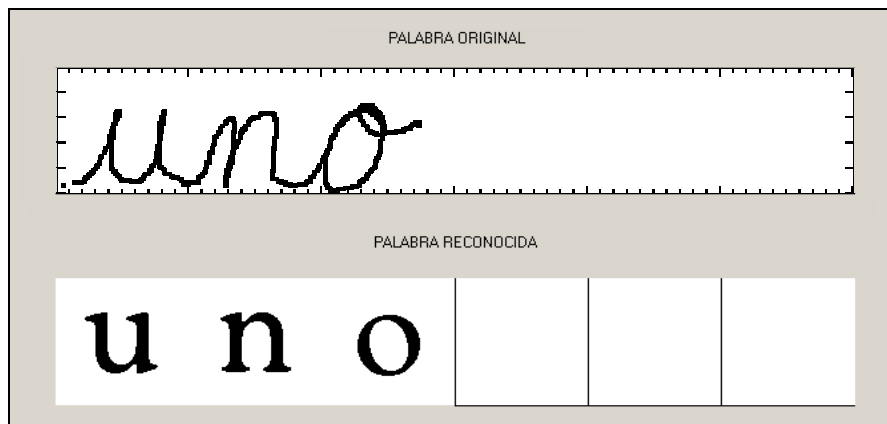


Figura 5.8. Resultados del reconocimiento de palabras por el método de vectores

Es de esta manera como el sistema realiza el reconocimiento de palabras. El proceso de comparación, nos permite hacer casi inmediatamente dicho reconocimiento. Se observa también, la importancia de la extracción de características, como un proceso fundamental para la comparación posterior. Otra observación importante es la diferencia que existe entre el reconocimiento de caracteres aislados y enlazados, ya que para este último caso, se requieren procesos adicionales, tales como la segmentación y el centrado.

La funcionalidad y eficiencia de cada uno de los procesos de reconocimiento se analizará en el Capítulo 7 de pruebas y validación del sistema.

Capítulo 6

Aplicación específica del RCM

A continuación se describe el funcionamiento de la interfaz del Sistema de Reconocimiento de Caracteres Manuscritos, el cual hemos implementado a fin de realizar una comparación clara de los algoritmos de reconocimiento de caracteres aislados, así como ilustrar el funcionamiento del algoritmo de reconocimiento de palabras. En este caso, también tiene como propósito mostrarnos un paso anterior a dicho reconocimiento, el cual es la segmentación de la palabra en la unidad mínima a reconocer, es decir, caracteres.

La interfaz de este sistema permite que el usuario lo utilice de un modo sencillo, ya que lo único que se requiere es que la escritura a reconocer sea capturada en la interfaz correspondiente, mientras que todos los procesos de reconocimiento posteriores son realizados automáticamente por el sistema.

Describiremos ahora las pantallas que componen al sistema. En primer lugar aparece la pantalla de presentación del programa, la cual se muestra en la Figura 6.1.

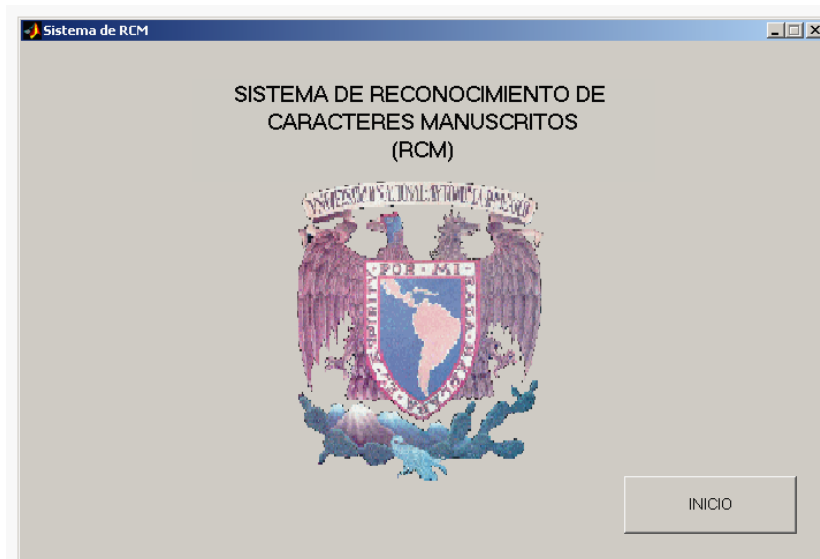


Figura 6.1. Presentación del sistema

Una vez que presionamos la opción INICIO, aparece la pantalla mostrada en la Figura 6.2, en la cual podemos elegir entre analizar un carácter o una palabra.

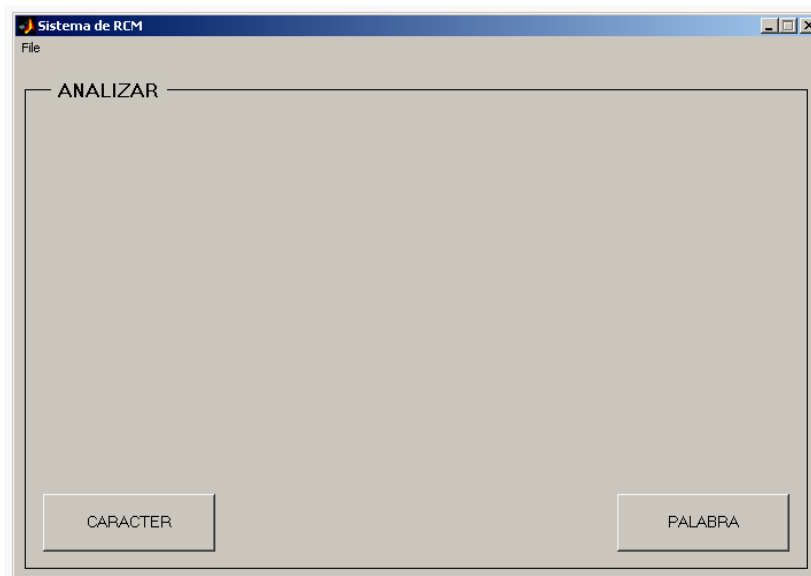


Figura 6.2 Selección de análisis

Al seleccionar la opción **CARACTER**, aparece la pantalla mostrada en la Figura 6.3, en la cual aparecen las instrucciones a seguir para realizar el reconocimiento de los caracteres aislados del diccionario.

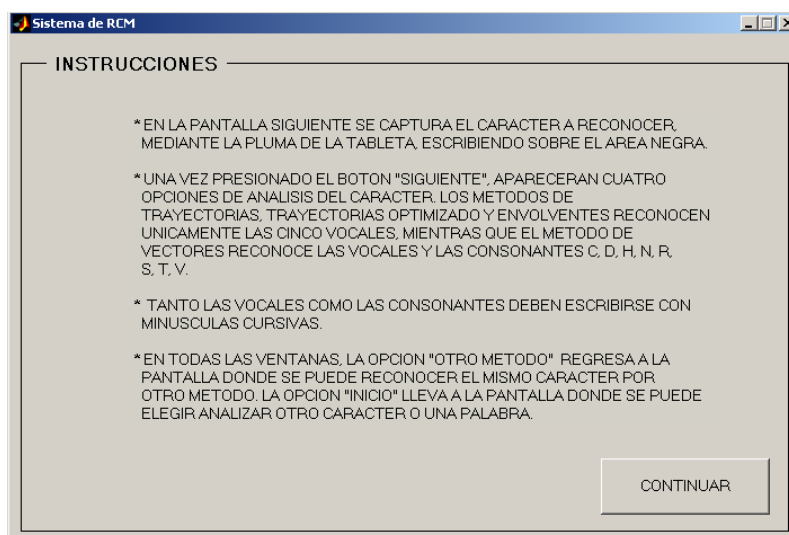


Figura 6.3. Instrucciones para el análisis de caracteres

Después de leer las instrucciones, al presionar CONTINUAR aparecerá la pantalla mostrada en la Figura 6.4, en la que se captura el carácter por medio de la pluma integrada a la tableta digitalizadora, escribiendo con ella dentro del área activa (negra) de la interfaz.

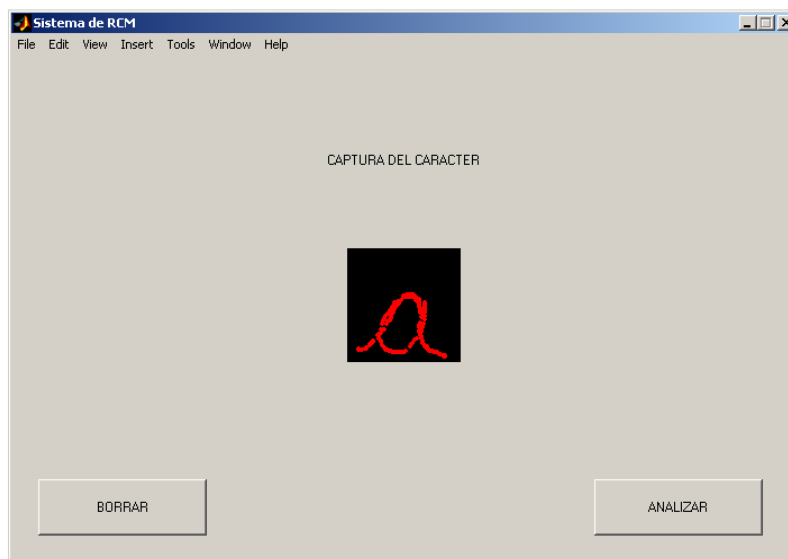


Figura 6.4. Captura de caracteres

Con la opción BORRAR podemos borrar todo lo que haya dentro del área activa y escribir de nuevo el carácter que queremos analizar. Una vez que estamos de acuerdo con la escritura, elegimos la opción ANALIZAR, con lo que aparece la pantalla mostrada en la Figura 6.5, donde se muestra el carácter que ha sido capturado.

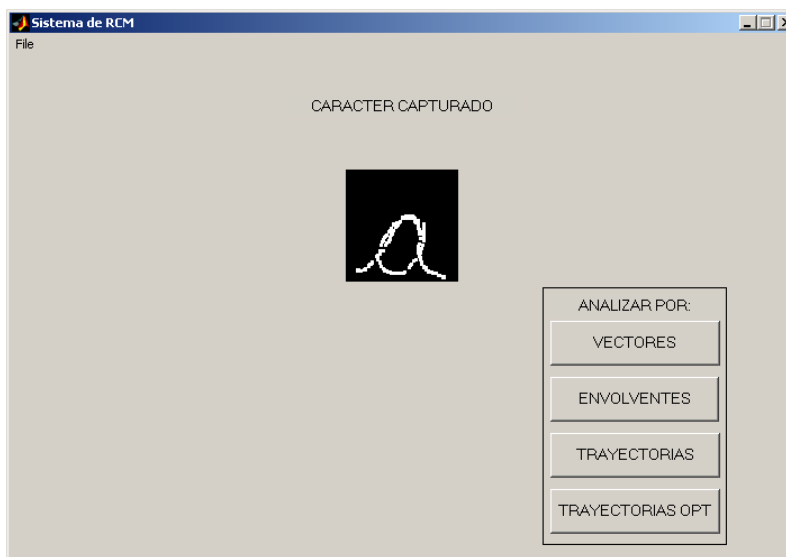


Figura 6.5. Selección de métodos de análisis del carácter

Como podemos observar, en la parte inferior derecha tenemos la opción de elegir entre cuatro métodos de reconocimiento: por vectores, por envolventes, por trayectorias o por trayectorias optimizado.

Estos cuatro métodos reconocen vocales, mientras que el método de vectores también reconoce consonantes, lo cual puede indicarse en la pantalla de instrucciones que vimos anteriormente.

Al escoger el reconocimiento por VECTORES tenemos la pantalla mostrada en la Figura 6.6.



Figura 6.6. Reconocimiento por vectores

En esta pantalla tenemos dos opciones: OTRO METODO, lo cual nos permite analizar el mismo carácter por cualquiera de los otros tres métodos, o bien la opción INICIO, que nos lleva a la pantalla de la Figura 6.2, donde podemos elegir entre analizar otro carácter o una palabra.

Después de elegir OTRO METODO y elegir ahora el método de reconocimiento por ENVOLVENTES, tenemos la pantalla mostrada en la Figura 6.7.



Figura 6.7. Reconocimiento por envoltentes

Del mismo modo, para el método de TRAYECTORIAS tenemos la pantalla mostrada en la Figura 6.8, donde se muestra el reconocimiento del mismo carácter.



Figura 6.8. Reconocimiento por trayectorias

En esta pantalla tenemos, del mismo modo que en el método anterior, la opción de analizar el mismo carácter por otro método o volver a la pantalla de inicio.

Por el método de TRAYECTORIAS OPTIMIZADO tenemos el resultado mostrado en la Figura 6.9 para el mismo carácter.

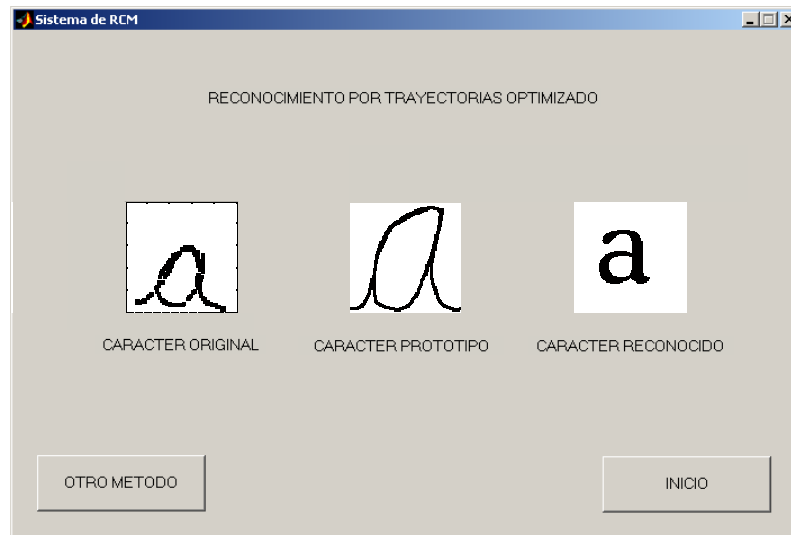


Figura 6.9. Pantalla de reconocimiento por trayectorias

En los cuatro métodos de análisis obtenemos como resultado el carácter original, el carácter prototipo de la base de datos con el que coincide dicho carácter, y un carácter escrito en un formato que puede ser reconocido a simple vista para confirmar el reconocimiento.

Para todos los métodos, eligiendo la opción **INICIO**, podemos analizar otro carácter o analizar una palabra. Si elegimos analizar una palabra, al igual que con la opción para analizar caracteres, aparece una pantalla de instrucciones, ver la Figura 6.10, las cuales debemos leer y seguir para un mejor desempeño de los algoritmos que conforman el reconocimiento de palabras.

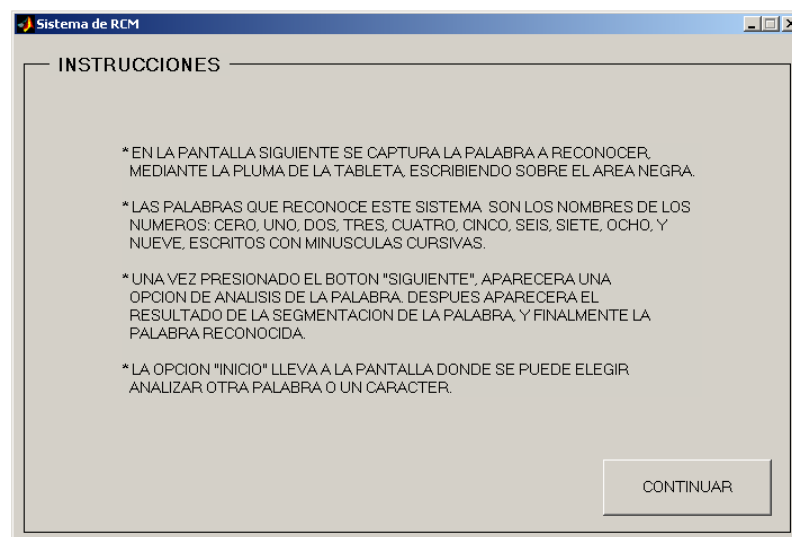


Figura 6.10. Instrucciones para el análisis de palabras

Una vez presionado el botón CONTINUAR, aparece la pantalla mostrada en la Figura 6.11, en la cual se captura la palabra del mismo modo que con los caracteres, ahora en un área activa adecuada al tamaño de las palabras elegidas para su reconocimiento.



Figura 6.11. Captura de palabras

Al presionar la opción ANALIZAR tenemos ahora, como se muestra en la Figura 6.12, la pantalla donde se muestra la palabra que ha sido capturada. Como podemos ver, tenemos ahora un solo método de reconocimiento, por VECTORES, con segmentación por localización de extremos superiores.

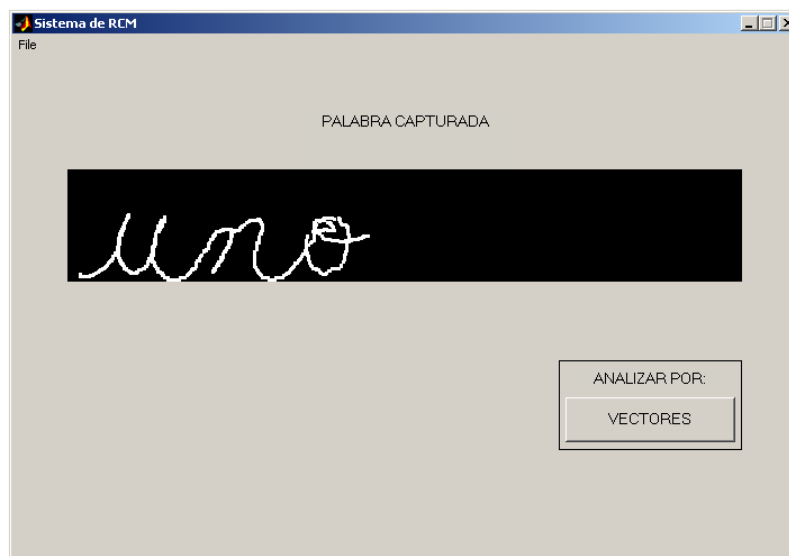


Figura 6.12. Pantalla previa al análisis de la palabra

Al escoger el método de análisis disponible, tenemos en primer lugar la pantalla mostrada en la Figura 6.13, donde se muestra la palabra original y la palabra segmentada por localización de extremos superiores.

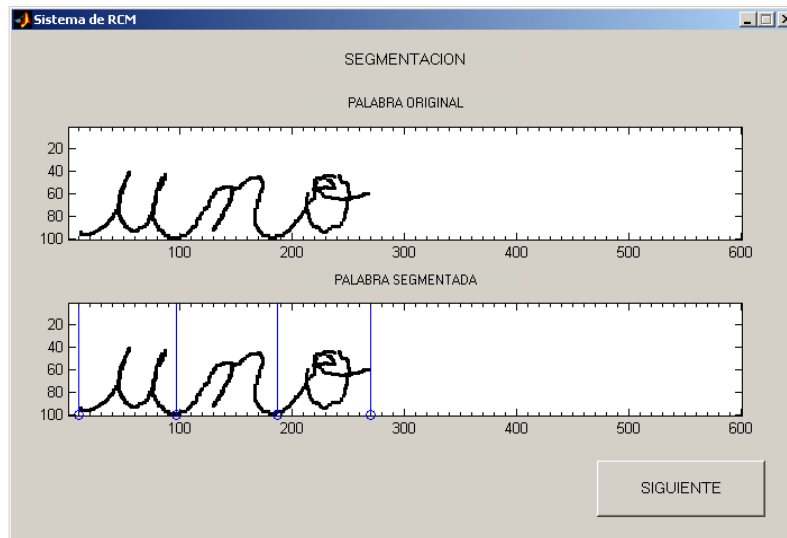


Figura 6.13. Segmentación de la palabra

Al presionar SIGUIENTE, tenemos la pantalla mostrada en la Figura 6.14, donde se muestra el reconocimiento de cada carácter que forma a la palabra.

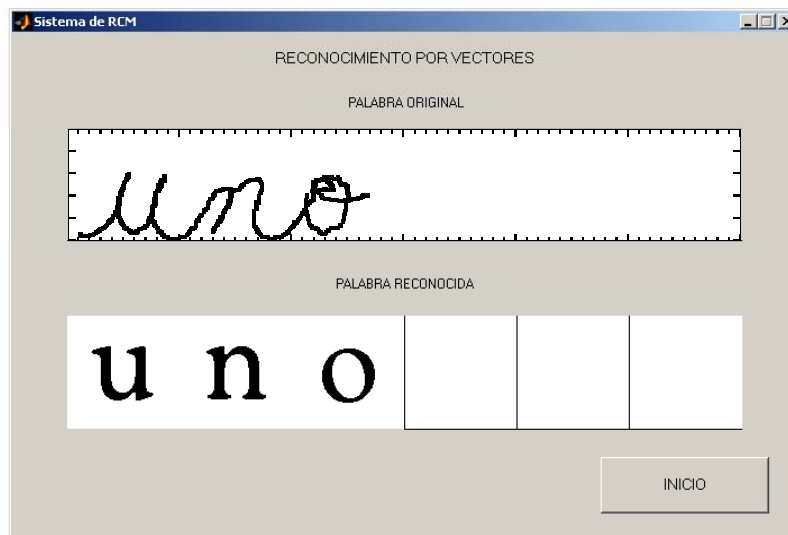


Figura 6.14. Reconocimiento de la palabra

Como ya hemos mencionado, una vez que se ha completado el análisis de la palabra, seleccionando INICIO podemos volver a analizar otra palabra, o bien, caracteres aislados.

De esta manera concluimos la descripción del funcionamiento de la interfaz gráfica de nuestro sistema, cuyo propósito principal es ser una herramienta sencilla de usar por otras personas (usuarios finales), que les permita comparar el desempeño de los diversos algoritmos de reconocimiento que desarrollamos, así como apreciar con claridad la simulación de dichos algoritmos dentro de la secuencia de procesamiento para el reconocimiento de las letras y palabras cortas de nuestro diccionario.

En el Capítulo 7 se mostrarán las pruebas y validación del sistema de reconocimiento de escritura manuscrita que ya hemos presentado, y cuya evaluación se ha visto facilitada mediante el uso del ambiente de simulación (interfaz gráfica) que hemos presentado en este capítulo.

Capítulo 7

Pruebas y Resultados

7.1 Pruebas

Para el reconocimiento de caracteres, tenemos cuatro métodos de reconocimiento: envolventes, trayectorias y trayectorias optimizado, los cuales reconocen vocales, mientras que el método de reconocimiento por vectores es útil para reconocer tanto vocales como consonantes. Para determinar esto, primero realizamos el reconocimiento de todos los caracteres (vocales y consonantes) por los cuatro métodos. Para cada método hicimos diez pruebas por carácter, esto es, 130 pruebas por método en total. Para el reconocimiento de vocales, en los cuatro métodos obtuvimos tasas de eficiencia superiores al 80%, por lo que los incluimos todos en nuestro sistema de reconocimiento.

Para las 8 consonantes del diccionario, tenemos en total 80 pruebas por método. Los métodos de envolventes, trayectorias y trayectorias optimizado obtuvieron tasas de reconocimiento entre 20% y 30%, mientras que el método de vectores obtuvo una eficiencia del 72.5%. Puesto que la eficiencia de reconocimiento de consonantes no fue aceptable sino para el método de vectores, elegimos dicho método para el reconocimiento de consonantes. Puesto que el método por vectores reconoce tanto vocales como consonantes con buenas tasas de eficiencia, podemos concluir que dicho algoritmo es apropiado también para su aplicación en el reconocimiento de palabras.

En dichas pruebas, al tener árboles de decisión en cada método e incorporar condicionantes para distinguir las vocales de las consonantes, las consideraciones hechas hacían que se confundieran las identidades de las vocales con las consonantes, lo cual ocasionaba bajas tasas de reconocimiento. Por ello, al eliminar las consonantes y sus condicionales, se obtuvo una mejora en los métodos de trayectorias, envolventes y trayectorias optimizado en el reconocimiento de vocales, lo cual se puede observar en las tablas dadas más adelante.

Resultado de las pruebas de reconocimiento de caracteres

En las Tablas 7.1, 7.2, 7.3 y 7.4 tenemos los resultados de las pruebas para el reconocimiento de vocales por los métodos de trayectorias, envolventes, trayectorias optimizado y vectores, respectivamente. Para el método de vectores, tenemos en la Tabla 7.5 los resultados de las pruebas para las consonantes que forman parte de las palabras del diccionario: c, d, h, n, r, s, t, v. Por cada letra en cada método se realizaron diez pruebas de escritura, hechas por diferentes personas, para obtener resultados más confiables.

CARÁCTER	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR CARÁCTER
<i>a</i>	9	1	0	90%
<i>e</i>	10	0	0	100%
<i>i</i>	10	0	0	100%
<i>o</i>	8	2	0	80%
<i>u</i>	8	2	0	80%
Total	45	5	0	90%

Tabla 7.1. Resultados del reconocimiento de vocales por trayectorias

CARÁCTER	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR CARÁCTER
<i>a</i>	10	0	0	100%
<i>e</i>	10	0	0	100%
<i>i</i>	9	1	0	90%
<i>o</i>	10	0	0	100%
<i>u</i>	10	0	0	100%
Total	49	1	0	98%

Tabla 7.2. Resultados del reconocimiento de vocales por envolventes

CARÁCTER	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR CARÁCTER
<i>a</i>	10	0	0	100%
<i>e</i>	10	0	0	100%
<i>i</i>	10	0	0	100%
<i>o</i>	10	0	0	100%
<i>u</i>	7	3	0	70%
Total	47	3	0	94%

Tabla 7.3. Resultados del reconocimiento de vocales por trayectorias optimizado

CARÁCTER	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR CARÁCTER
<i>a</i>	9	1	0	90%
<i>e</i>	8	1	1	80%
<i>i</i>	10	0	0	100%
<i>o</i>	10	0	0	100%
<i>u</i>	10	0	0	100%
Total	47	2	1	94%

Tabla 7.4. Resultados del reconocimiento de vocales por vectores

CARÁCTER	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR CARÁCTER
<i>c</i>	10	0	0	100%
<i>d</i>	9	1	0	90%
<i>h</i>	10	0	0	100%
<i>n</i>	9	0	1	90%
<i>r</i>	9	0	1	90%
<i>s</i>	10	0	0	100%
<i>t</i>	9	1	0	90%
<i>v</i>	8	0	2	80%
Total	74	2	4	92.5%

Tabla 7.5. Resultados del reconocimiento de consonantes por vectores

Como podemos observar en las Tablas 7.1, 7.2, 7.3 y 7.4, para los cuatro métodos de reconocimiento de vocales tenemos eficiencias totales superiores al 90%, y tasas de reconocimiento por carácter superiores al 80% en prácticamente todos los casos, alcanzando el 100% de eficiencia en por lo menos la mitad de las vocales. Aunque el reconocimiento de vocales por vectores está en el segundo lugar de eficiencia respecto a los tres métodos restantes, su eficiencia en el reconocimiento de consonantes, mostrada en la Tabla 7.5, aumentó considerablemente respecto a su eficiencia mostrada en las pruebas preliminares mencionadas al principio de la sección 7.2.

Como se observa en dicha tabla, las tasas de reconocimiento por carácter son superiores al 80% en todos los casos, mientras que su eficiencia total es de 92.5%, con lo que confirmamos que este método será de gran utilidad en el reconocimiento de palabras, lo cual analizaremos a continuación.

Resultado de las pruebas de segmentación y reconocimiento de palabras

Para las pruebas del reconocimiento de palabras se realizaron diez pruebas de escritura para el método de reconocimiento por vectores. Se evalúa también el desempeño del algoritmo de segmentación por localización de extremos superiores, considerando si el número de puntos de segmentación, el cual debe ser igual al número de letras en la palabra mas uno, es correcto, así como si dichos puntos están ubicados correctamente, es decir, que no cortan a una letra de manera que la identidad de la palabra pueda verse afectada, ver los resultados mostrados en la Tabla 7.6.

A continuación tenemos los resultados de las pruebas para las diez palabras seleccionadas, correspondientes al desempeño del algoritmo de segmentación por localización de extremos superiores, y al del método de reconocimiento por vectores, cuyo algoritmo es el mismo que se utilizó para el reconocimiento de caracteres aislados, con las modificaciones y adaptaciones mencionadas en la sección 5.2 del Capítulo 5. En cada palabra se analiza el número de caracteres contenidos en ella que han sido reconocidos correctamente, así como el número de palabras totalmente reconocidas, lo cual se muestra en las Tablas 7.7 y 7.8.

PALABRA	PUNTOS DE SEGMENTACIÓN TEÓRICOS	PUNTOS DE SEGMENTACIÓN COLOCADOS	PUNTOS DE SEGMENTACIÓN CORRECTAMENTE COLOCADOS	PORCENTAJE DE EFICIENCIA POR PALABRA
<i>cero</i>	50	50	50	100%
<i>uno</i>	40	40	38	95%
<i>dos</i>	40	40	39	97.5%
<i>tres</i>	50	49	47	94%
<i>cuatro</i>	70	72	68	97.14%
<i>cinco</i>	60	60	60	100%
<i>seis</i>	50	50	50	100%
<i>siete</i>	60	60	59	98.33%
<i>ocho</i>	50	50	50	100%
<i>nueve</i>	60	62	58	96.66%
Total	530	533	519	97.92%

Tabla 7.6. Resultados de la segmentación por localización de extremos superiores

En la Tabla 7.6 tenemos los resultados de la segmentación por localización de extremos superiores, que fue aplicada a las palabras de prueba cuyo reconocimiento se analiza en las Tablas 7.7 y 7.8. Para medir el porcentaje de eficiencia de segmentación por palabra, se utilizó el cociente del número de puntos de segmentación correctamente colocados entre el número de puntos de segmentación teóricos, multiplicado por cien.

Para determinar el número de puntos de segmentación teóricos tenemos por ejemplo, que en la palabra “cero” tenemos 5 puntos de segmentación teóricos, que multiplicados por diez pruebas

realizadas nos da un total de 50 puntos de segmentación teóricos. El número de puntos correctamente colocados es el número de puntos correctos en número y posición en 10 pruebas de cada palabra.

Como podemos ver, la eficiencia por palabra en todos los casos es superior al 90%, mientras que la eficiencia total es prácticamente de 98%, lo cual constituye muy buenos resultados en el desempeño del algoritmo de segmentación por localización de extremos superiores. Esto repercute en el éxito del reconocimiento de palabras, como se muestra en las tablas siguientes.

PALABRA	NUMERO TOTAL DE CARACTERES	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE	CARACTERES NO RECONOCIDOS	TASA DE RECONOCIMIENTO POR PALABRA
<i>cero</i>	40	38	0	2	95%
<i>uno</i>	30	28	0	2	93.33%
<i>dos</i>	30	28	1	1	93.33%
<i>tres</i>	40	37	1	2	92.5%
<i>cuatro</i>	60	56	2	2	93.33%
<i>cinco</i>	50	48	1	1	96%
<i>seis</i>	40	36	2	2	90%
<i>siete</i>	50	45	2	3	90%
<i>ocho</i>	40	39	0	1	97.5%
<i>nueve</i>	50	46	3	1	92%
Total	430	401	12	17	93.26%

Tabla 7.7. Resultados del reconocimiento por vectores con segmentación por localización de extremos superiores

En la Tabla 7.7 podemos observar el desempeño del reconocimiento de las palabras de prueba, en la cual se analiza el resultado de la aplicación del algoritmo de reconocimiento por vectores en cada carácter que forma a dichas palabras. Para determinar la tasa de reconocimiento de caracteres por cada palabra, se consideró el cociente del número de caracteres reconocidos correctamente entre el número total de caracteres, multiplicado por cien.

En este caso, las tasas de reconocimiento por palabra son iguales o mayores al 90% en todos los casos, mientras que la tasa total de reconocimiento es de 93.26%, lo cual es un porcentaje de eficiencia muy bueno, semejante al obtenido en el reconocimiento de caracteres aislados, es decir, vocales y consonantes, lo cual se analizó en las Tablas 7.4 y 7.5.

En la tabla siguiente se muestran los resultados del reconocimiento de palabras, esta vez, considerando a las palabras enteras.

PALABRA	PALABRAS TOTALMENTE RECONOCIDAS	PALABRAS PARCIALMENTE RECONOCIDAS	TASA DE RECONOCIMIENTO POR PALABRA
<i>cero</i>	9	1	90%
<i>uno</i>	8	2	80%
<i>dos</i>	8	2	80%
<i>tres</i>	7	3	70%
<i>cuatro</i>	8	2	80%
<i>cinco</i>	9	1	90%
<i>seis</i>	8	2	80%
<i>siete</i>	7	3	70%
<i>ocho</i>	9	1	90%
<i>nueve</i>	7	3	70%
Total	80	20	80%

Tabla 7.8. Resultados del reconocimiento de palabras

En la tabla anterior tenemos los resultados del reconocimiento de palabras por el método de vectores, y en la cual se muestra el número de palabras totalmente reconocidas, es decir, de aquellas palabras en las que todos los caracteres que las forman han sido reconocidos, y cuya posición dentro de dichas palabras es correcta. Las palabras parcialmente reconocidas son aquellas en las cuales hubo cuando menos un carácter que no fue reconocido, o que fue incorrectamente reconocido.

Como podemos observar, las tasas de reconocimiento por palabra son superiores al 70% en todos los casos, y alcanzan hasta un 90% de eficiencia, lo cual, aunque representa porcentajes de eficiencia menores a los obtenidos en el reconocimiento de caracteres aislados, no significa que el método de reconocimiento usado tenga fallas significativas, sino que sólo se está considerando palabras totalmente reconocidas al calcular la eficiencia total, la cual es de 80%. Sin embargo, como puede verse en la Tabla 7.7, el número total de caracteres reconocidos incorrectamente y no reconocidos es solamente de 6.74%, lo que nos habla de que en caso de error en el reconocimiento de algún carácter, todavía es posible determinar la identidad de la palabra, aplicando otros criterios de reconocimiento.

7.2 Resultados

En esta sección analizaremos los resultados obtenidos del reconocimiento de caracteres, del método de segmentación elegido, así como del reconocimiento de palabras. Para medir la eficiencia de los métodos de reconocimiento, se considera que el reconocimiento ha sido exitoso si los caracteres fueron reconocidos correctamente, y se considera que el reconocimiento ha sido fallido si los caracteres fueron reconocidos incorrectamente, o bien, si los caracteres no han sido reconocidos en absoluto.

7.2.1 Reconocimiento de caracteres

Analizaremos los resultados del reconocimiento de caracteres por los métodos de trayectorias, envolventes, trayectorias optimizado y vectores. Como ya mencionamos, estos cuatro métodos reconocen vocales, mientras que el método de vectores reconoce también consonantes. En primer lugar, analizaremos el desempeño de los métodos de reconocimiento de vocales.

Considerando los totales de las Tablas 7.1, 7.2, 7.3 y 7.4, y tomando en cuenta que el número de pruebas para el conjunto de las cinco vocales, es decir, diez por cada letra, fue de cincuenta para cada método, tenemos a continuación la Tabla 7.9, en la cual se muestran los resultados de los cuatro métodos de reconocimiento de vocales, así como su tasa de eficiencia.

METODO DE RECONOCIMIENTO	TOTAL DE CARACTERES RECONOCIDOS CORRECTAMENTE	TOTAL DE CARACTERES RECONOCIDOS INCORRECTAMENTE Y NO RECONOCIDOS	TASA DE EFICIENCIA
Trayectorias	45	5	90%
Envolventes	49	1	98%
Trayectorias optimizado	47	3	94%
Vectores	47	3	94%

Tabla 7.9. Resultados del reconocimiento de vocales

Para el reconocimiento de consonantes hemos considerado solamente el método de vectores, puesto que en los demás métodos, al incluir en los algoritmos las consideraciones necesarias para reconocer consonantes, éstas causaban mayor confusión en el reconocimiento, obteniendo bajas tasas de reconocimiento, y al eliminar dichas consideraciones se aumentó la eficiencia de los 3 métodos.

Puesto que el método de vectores incluye una serie de características distintivas para cada carácter, aunque la tasa de reconocimiento de vocales no es la más alta de todos los métodos, considera también el reconocimiento de consonantes, con un buen nivel de eficiencia, lo cual es útil para el reconocimiento de palabras.

Para el reconocimiento de consonantes, como hemos dicho, utilizamos el método de vectores, realizando, de la misma manera que para las vocales, diez pruebas para cada carácter, lo cual nos da un total de ochenta pruebas, de las cuales obtuvimos los resultados mostrados en la Tabla 7.10.

NUMERO TOTAL DE CARACTERES	TOTAL DE CARACTERES RECONOCIDOS CORRECTAMENTE	TOTAL DE CARACTERES RECONOCIDOS INCORRECTAMENTE Y NO RECONOCIDOS	TASA DE EFICIENCIA
80	74	6	92.5%

Tabla 7.10. Resultados del reconocimiento de consonantes

Como se mencionó en la sección 7.2, en las pruebas preliminares de reconocimiento de consonantes, los métodos de envolventes, trayectorias y trayectorias optimizado obtuvieron tasas de reconocimiento entre 20% y 30%, mientras que el método de vectores obtuvo una eficiencia del 72.5%. Como podemos observar en la Tabla 7.10, en los resultados de las pruebas finales la tasa de reconocimiento de consonantes fue de 92.5%, lo que significa un aumento de 20% en el desempeño de este método, lo cual se debe a mejoras en el algoritmo, y cuya efectividad se refleja en los resultados finales obtenidos.

Tanto para el reconocimiento de vocales como de consonantes, podemos apreciar porcentajes de eficiencia superiores al 90%, lo cual se traduce en eficiencia de algoritmos, efectividad de criterios de diferenciación de caracteres, y acertada eliminación de consonantes de los métodos de reconocimiento por envolventes, trayectorias y trayectorias optimizado.

7.2.2 Segmentación

Para analizar los resultados del algoritmo de segmentación por localización de extremos superiores que utilizamos, debemos considerar el número de puntos de segmentación teóricos, los cuales son para cada palabra igual al número de letras en la palabra mas uno, puesto que colocamos un punto que señala el inicio de la palabra, los puntos de segmentación entre letras, y el punto final de la palabra.

En las pruebas realizadas registramos el número de puntos que se han colocado en cada palabra, los cuales deben ser iguales en número al número de puntos de segmentación teóricos. Un número menor de puntos indicará que se han tomado juntos dos caracteres, mientras que un número mayor de puntos indicará la sobre-segmentación de algún carácter o ligadura. Dichos puntos de segmentación deben estar correctamente colocados, es decir, que su ubicación no corte el cuerpo del carácter, o bien que no abarquen segmentos de caracteres adyacentes.

Con estas consideraciones, decimos que los puntos de segmentación correctos en una palabra son aquellos que son correctos en número y posición. Las pruebas del algoritmo de segmentación se llevaron a cabo sobre las mismas palabras que sirvieron para evaluar el algoritmo de reconocimiento de palabras, es decir, diez pruebas por palabra. Estos resultados se muestran en la Tabla 7.11.

TOTAL DE PUNTOS DE SEGMENTACIÓN TEÓRICOS	TOTAL DE PUNTOS DE SEGMENTACIÓN CORRECTOS	TASA DE EFICIENCIA
530	519	97.92%

Tabla 7.11. Resultados de la segmentación por localización de extremos superiores

El total de puntos de segmentación correctos se determinó mediante el número de puntos correctamente colocados, ya que éstos son los que deben estar colocados sobre la palabra, y su posición dentro de ella debe ser correcta. El porcentaje de eficiencia obtenido, prácticamente 98%, es muy buen resultado, lo que nos habla de una correcta selección del algoritmo de segmentación incorporado al sistema, el cual considera las características particulares de la escritura cursiva manuscrita que analizamos, y aunque contempla ciertas restricciones a dicha escritura, las cuales fueron descritas en la sección 3.2 del Capítulo 3, no limita de manera importante la naturalidad en los trazos realizados por cada autor.

7.2.3 Reconocimiento de palabras

Como ya hemos mencionado, el reconocimiento de palabras se hace carácter por carácter, analizando cada uno de ellos por el método de reconocimiento por vectores. Para cada palabra se hicieron diez pruebas, por lo que tenemos un total de cien palabras a reconocer. En primer lugar, en la Tabla 7.12 tenemos la estadística del número total de caracteres que han sido reconocidos en esas cien palabras, ya que con esta estadística podemos determinar el número de palabras totalmente reconocidas, es decir, aquellas en las cuales todos sus caracteres han sido reconocidos.

NUMERO TOTAL DE CARACTERES	CARACTERES RECONOCIDOS CORRECTAMENTE	CARACTERES RECONOCIDOS INCORRECTAMENTE Y NO RECONOCIDOS	TASA DE EFICIENCIA
430	401	29	93.25%

Tabla 7.12. Resultados del reconocimiento por vectores carácter por carácter en cada palabra

En la tabla anterior, podemos observar que solamente 29 de los 430 caracteres a reconocer dentro de las diez pruebas realizadas por cada una de las diez palabras del diccionario, no fueron reconocidos, lo cual representa un error de 6.75%, lo cual es un buen resultado, que repercute directamente en el número de palabras completamente reconocidas, ya que mientras menor sea el

número de caracteres no reconocidos o reconocidos incorrectamente, mayor es el número de palabras con todos sus caracteres reconocidos correctamente, y por lo tanto, reconocidas completamente.

Finalmente, en la Tabla 7.13 tenemos los resultados del reconocimiento de palabras; esto quiere decir que se considera el reconocimiento correcto de cada uno de los caracteres que las forman y su ordenación correcta en la interfaz del sistema.

NUMERO TOTAL DE PALABRAS	PALABRAS TOTALMENTE RECONOCIDAS	PALABRAS PARCIALMENTE RECONOCIDAS	TASA DE EFICIENCIA
100	80	20	80%

Tabla 7.13. Resultados del reconocimiento de palabras

En la tabla anterior observamos que la tasa de eficiencia en el reconocimiento de palabras es de 80%, este porcentaje es bueno en sí mismo, pero además, debemos agregar que las palabras parcialmente reconocidas no deben descartarse totalmente al evaluar el éxito del reconocimiento, ya que en aquellas palabras cuyo número de caracteres no reconocidos no sea mucho mayor al número total de sus caracteres, es posible que mediante otras herramientas, tales como información lingüística, contexto, comparación con un diccionario base, u otros criterios de diferenciación dentro de los algoritmos, sea posible que la información obtenida de ellas aún pueda ser útil para el reconocimiento de la identidad de la palabra entera.

De esta manera hemos presentado los resultados obtenidos en el desempeño del sistema de reconocimiento de escritura manuscrita desarrollado. Como pudimos observar, las tasas de eficiencia alcanzadas en cada parte del proceso son satisfactorias, lo cual nos permite cumplir efectivamente con una parte importante de los objetivos que nos hemos planteado desde el principio de este trabajo.

En el capítulo siguiente se analiza el desempeño de cada parte del sistema presentado, desde la captura hasta los resultados del reconocimiento. Por otro lado, se plantean las fortalezas y debilidades de dicho sistema, tanto respecto a sí mismo como respecto a sistemas descritos en otros trabajos, los cuales se encuentran en la revisión bibliográfica que realizamos. Lo anterior nos permitirá realizar una evaluación cualitativa y cuantitativa del sistema, así como el planteamiento de aplicaciones y perspectivas de trabajo a futuro.

Capítulo 8

Conclusiones y Perspectivas

Conclusiones sobre la captura

El bloque de captura le permite al sistema la obtención de los datos necesarios para nuestros propósitos tanto de forma generales como para nuestros objetivos particulares. Nuestro sistema de captura es fundamentalmente un sistema de captura de imágenes de mapa de bits, el cual podemos modificar para capturar un ancho mayor o menor en los trazos de los caracteres, al ser un sistema de comunicación vía puerto serie la recepción de los datos es en cascada, lo cual nos permite un flujo constante de información, algo que es importante es que el sistema muestra los trazos de la escritura en línea (tiempo real).

Esto nos permite observar como se va desarrollando nuestra escritura y nos da un punto de referencia sobre el lugar, tamaño y secuencia de los caracteres que estamos escribiendo, en contraparte con esto perdemos en la velocidad de captura, por lo que nuestra escritura tiene que ser de una forma un poco más lenta de lo habitual, por lo que recomendamos un tiempo de entrenamiento para realizar de una manera satisfactoria nuestra escritura en el sistema, como ya hemos comentado nuestra captura es de imágenes de mapa de bits, por lo que las características que podemos utilizar para el reconocimiento se basan en este tipo de características, esto no quiere decir que solo pueda ser por este medio, también se puede realizar la captura de otro tipo de características de la escritura, como son la presión, la velocidad, la escritura virtual, etc.

Conclusiones sobre la interfaz gráfica

Se enfocó la interfaz a las necesidades de visualización, facilidad y entendimiento de cada paso, de acuerdo a nuestros procesos principales, sin saturar al usuario con datos e imágenes resultados de procesos intermedios a los bloques principales del sistema, como son captura, cálculo de puntos de segmentación, centrado de las imágenes, extracción de características comparación y reconocimiento.

En cada ventana de la interfaz se muestran tanto instrucciones, opciones, resultados, nuestra forma de desplegar resultados del reconocimiento es carácter por carácter, mostrando una imagen del carácter reconocido en un formato prediseñado, en caso de no lograr un reconocimiento, esto es que el sistema no asocia ningún carácter prototipo a la prueba, el sistema despliega un signo de interrogación "?", lo cual es muy práctico y entendible.

Conclusiones sobre el algoritmo de segmentación

De la Tabla 7.6 del Capítulo 7, donde se analizan los resultados de la segmentación por localización de extremos superiores, podemos ver que en algunos casos, el número de puntos de segmentación colocados es menor al número teórico de puntos. Esto quiere decir que puede haber puntos que no están separando a dos o más caracteres. En ese caso, dichos puntos no están bien colocados, o bien, faltan puntos de segmentación, por lo que no se les considera dentro del número de puntos de segmentación correctos.

En algunos otros casos, podemos apreciar que el número de puntos colocados excede al número de puntos teóricos, lo que se interpreta como sobre-segmentación de algún carácter o ligadura. Estos puntos sobrantes también son considerados como incorrectos.

Ambas situaciones afectan directamente la eficiencia del algoritmo de reconocimiento de palabras implementado. Al comparar la Tabla 7.6 con la Tabla 7.7, donde se analizan los resultados del reconocimiento de palabras por vectores con segmentación por localización de extremos superiores, observamos que aquellas palabras con mayor o menor número de puntos de segmentación respecto al número de puntos teóricos, tienen un mayor número de caracteres reconocidos incorrectamente, así como un mayor número de caracteres no reconocidos. Esto afecta el número de palabras totalmente reconocidas, lo que nos hace concluir que el éxito en el algoritmo de segmentación influye directamente en el éxito del reconocimiento de palabras.

Con esto no queremos decir que el éxito en el reconocimiento de palabras dependa exclusivamente de la eficiencia del algoritmo de segmentación, ya que éste depende también de la eficiencia de los algoritmos de reconocimiento, entre otros factores, lo cual se analizará más adelante.

Por otro lado, un factor importante en el éxito del algoritmo de segmentación, es su correcta elección, es decir, que dicho algoritmo debe ser eficiente al aplicarse sobre el tipo de escritura que se está analizando, considerando sus características, ya que en este caso, la escritura cursiva manuscrita debe poder ser escrita con la mayor naturalidad posible, con la inclinación propia de su escritura, los enlaces entre las letras, así como otras características propias de este tipo de letra.

Los algoritmos de segmentación que descartamos, aunque funcionan adecuadamente dentro de sus propias restricciones, no permiten la naturalidad de escritura que hemos mencionado. Sin embargo, los métodos de segmentación por cortes y segmentación gráfica no deben ser descartados totalmente, ya que pueden permitir la correcta segmentación de otros tipos de escritura manuscrita, tales como la letra de molde que usamos comúnmente para escribir.

Finalmente, el porcentaje de éxito, obtenido con el algoritmo de segmentación por localización de extremos superiores, es del 97.92%, el cual es un muy buen nivel de eficiencia, por lo cual podemos decir que con este resultado, este método ayuda efectivamente a cumplir el objetivo del reconocimiento de palabras que nos hemos propuesto.

Conclusiones sobre la extracción de características

El objetivo de un bloque de extracción de características dentro de un sistema de reconocimiento de escritura es, como su nombre lo indica, el de extraer la información que defina o represente a cada una de las letras que el sistema deba reconocer. No se extraen todas las

características de la letra, sino solo aquellas que permitan la diferenciación posterior entre los caracteres. Para lograr lo anterior, es necesario conocer cómo es la escritura que el sistema debe reconocer. Este sistema en particular reconoce todas las palabras cursivas formadas hasta por seis letras, que pueden estar aisladas o conectadas. Las letras que este sistema puede reconocer son: a, e, i, o, u, c, d, h, n, r, s, t, v. También es importante conocer las características de la escritura dibujada por el medio de captura utilizado. En este sistema en particular, las letras se capturan con ayuda de una tableta gráfica digital y una interfaz diseñada para tal fin. Lo que se obtiene de la captura son imágenes en formato bmp, y una vez adecuadas, de ellas parte el proceso de extracción.

Hay que recordar que este sistema, permite el reconocimiento de letras aisladas por cuatro métodos, mientras que solo uno de ellos se aplica para el reconocimiento de palabras. Dependiendo del método por el cual el sistema realice el reconocimiento, se obtiene un conjunto específico de características a extraer. Esto se debe a que fueron diferentes los resultados obtenidos del análisis previo que se realizó para definir dicho conjunto. Para el Método de Reconocimiento por Envoltentes, por ejemplo, el bloque de extracción obtiene las envoltentes superiores e inferiores del carácter, además del promedio de puntos de corte, mientras que para los Reconocimientos por Trayectorias, de esas mismas envoltentes, el sistema extrae algunas propiedades específicas de las trayectorias que las forman, como las direcciones de inicio y de fin del trazo, el número de trayectorias que lo componen, las pendientes y longitud de cada una de ellas, y el promedio de puntos de corte. Hay que notar además que aunque una característica se utilice en distintos métodos, el peso de ella en el proceso puede ser diferente, tal es el caso de esta última característica de promedios de puntos de corte.

Además de los métodos anteriores, el sistema cuenta con un cuarto método, el Método de Reconocimiento por Vectores, que se aplica tanto a letras aisladas como a palabras. En el proceso de extracción, se obtiene una serie de características de la letra por regiones, tales como los anchos horizontal, vertical, medio, a partir del punto máximo; la altura, la posición del fin del trazo, y si dicho carácter es cerrado o abierto.

Los métodos de extracción para cada tipo de reconocimiento, parten de la matriz generada por la imagen capturada en formato bmp. La programación de los métodos fue relativamente sencilla. Los códigos se realizaron en Matlab versión 6.5. Fue importante identificar exactamente a cada una de las características para proponer el método a seguir para su extracción. En algunos casos fue suficiente realizar un barrido de la matriz, aunado a un conteo. En otros, se trabajó con ciclos anidados, promedios, sumas, etc. Para algunos de los métodos se realizaron procesos de obtención de una función de la imagen, en otras las características se obtuvieron directamente del análisis de la imagen. Haciendo una comparación entre los procesos de extracción, se obtuvo la tabla mostrada en la Tabla 8.1.

Método de extracción	Características Extraídas	Número de características	Elementos de comparación	Origen de la extracción	Base de datos
Envoltentes	<ul style="list-style-type: none"> • Envoltente superior • Envoltente inferior • Promedio puntos de corte 	3	Vector de 201 elementos	Matriz de la imagen	Imágenes
Trayectorias	<ul style="list-style-type: none"> • Dirección inicio • Dirección fin • Número de trayectorias • Pendiente de cada trayectoria • Longitud de cada trayectoria • Promedio de puntos de corte 	16	Vector de 16 elementos	Envoltente	Imágenes
Trayectorias Optimizado	<ul style="list-style-type: none"> • Dirección inicio • Dirección fin • Número de trayectorias • Pendiente de cada trayectoria • Longitud de cada trayectoria • Promedio de puntos de corte 	16	Vector de 16 elementos	Envoltente	Vectores
Vectores	<ul style="list-style-type: none"> • Ancho horizontal del cuerpo • Ancho vertical del cuerpo • Altura del carácter • Posición de termino del trazo • Ancho de la zona media • Cerrado o abierto, zona inferior • Ancho horizontal, punto máximo 	7	Vector de 7 elementos	Matriz de la imagen	Vectores

Tabla 8.1. Comparación entre los distintos métodos de extracción.

De la tabla anterior, podemos observar por qué el método obtenido para el Reconocimiento por Vectores fue el mejor; tanto el número de características a extraer como a comparar es reducido. Además la extracción parte directamente de la matriz, sin necesidad de realizar un proceso previo. Todo el análisis es por regiones, y es relativamente sencillo, ya que basta con hacer barridos de la matriz y localizar puntos o medir anchos. Las características se guardan en vectores pequeños, que son fáciles de manejar y referenciar.

Otro punto importante en la extracción es diferenciar los procesos de obtención de características para letras aisladas, de los aplicados para caracteres provenientes de la segmentación de una palabra. En este último caso se requiere de la aplicación de procesos de adecuación a cada una de las letras, una vez que ha sido segmentada la palabra. Lo anterior requiere del estudio del resultado del proceso de segmentación, con la finalidad de encontrar las diferencias entre los dibujos de un carácter aislado y de uno proveniente de la segmentación de la palabra. A veces es necesario, un proceso posterior a la segmentación, para corregir ciertos detalles. En este sistema, el único reconocimiento para palabras se realiza por el Método de Reconocimiento por Vectores. Ese proceso de reconocimiento, involucra un método de este tipo para centrar el carácter dentro de la zona de

reconocimiento y quitarle cualquier corrimiento que por la segmentación pudiera presentar. Es importante tener en cuenta las propiedades que se alteran por la aplicación de procesos previos a la extracción para casos particulares, como el reconocimiento de una palabra, debido a que dichos cambios afectarán los resultados esperados, si es que no se les considera.

La extracción de características es un bloque dentro del sistema que involucra varios procesos, tanto de análisis previo de la escritura, de las herramientas de captura, de los efectos de procesos previos y de la extracción en sí, así como aquellos que serán aplicados posteriormente. Cuando se trabaja con un bloque de este tipo, es conveniente adecuar tanto la entrada, como la salida a los otros bloques pensando en evitar la repetición de procesos y lograr la concordancia entre variables. En general esta es una de las etapas más importantes del sistema, junto con la segmentación previa y la comparación posterior. Y van tan relacionadas, que sin el buen funcionamiento de una, el sistema no lograría el adecuado reconocimiento de la escritura.

Conclusiones sobre las bases de datos

Nuestras bases de datos son de dos tipos de modelo de alto nivel, lo cual nos proporciona una buena calidad de datos, en un caso la base es sobre datos de tipo entidad, lo cual nos proporciona datos que representan objetos o conceptos del mundo real, esto en el caso de la base de datos de imágenes de tipo bmp, mientras que para las otras dos bases de datos las tenemos sobre datos de tipo atributo, esto es que representan una o varias propiedades de interés que describe a un objeto o concepto real.

Para la base de datos de imágenes no se uso compresor alguno, ya que la base de datos no es muy grande, y no necesitamos liberar espacio para un funcionamiento mejor, pero para un sistema que requiriera un abecedario más grande si sería conveniente tener un compresor de imágenes, o tener dichas imágenes en otro formato de imagen, ya que el formato bmp es de los que ocupa más memoria por cada imagen, respecto a los controladores de la base de datos, de haberlos construido representarían un gasto innecesario, tanto de trabajo, de memoria y de recursos, ya que nuestras bases de datos son no relacionales, además de que al tener cada etiqueta (a, c, d, e, h, i, n, o, r, s, t, u, v), un solo dato asociado a ella, el cual es una imagen en formato bmp, no es necesario un lenguaje de búsqueda estructurada.

Esta base de datos, nos proporciona una independencia entre los datos situados en la base y los programas, esto significa que se podrían cambiar los datos, sin que este cambio afecte a los programas que los utilizan, siempre y cuando conserven su formato, esto es debido a que se encuentran en un archivo independiente al programa, aunque estos provoca que debemos tener cuidado sobre la ruta seguir para leer la base de datos.

Para la base de datos vectorial utilizada para el método de trayectorias optimizado la obtenemos mediante la base de datos de imágenes, como ya se menciono anteriormente, por lo que comparte algunas características con ella, ambas son no relacionales y que a cada etiqueta le corresponde un solo vector en este caso, por lo cual no es necesario un lenguaje de búsqueda estructurada, en comparación, esta base de datos no es independiente a los programas, ya que se encuentra dentro de ellos (dentro del código), pero esto mismo nos proporciona una mayor rapidez del sistema, también se elimina el tener que llamar a un archivo externo, y la memoria a utilizar se reduce considerablemente, de tener una base de datos con imágenes en las cuales cada una necesita alrededor de 29.3 kbits a una en la cual solo se utiliza una línea de código con un vector de dieciséis

elementos (esta base de datos es la que requiere menos memoria), la diferencia es evidente, también se reduce el tiempo en el proceso, así como la excepción de algunos procesos ya que esta base de datos es de alto nivel, con datos asociados de tipo atributo, esto es, que representan las características del objeto y por lo tanto ya no aplicamos los procesos de extracción de características, ya que en el vector se encuentran las características de los prototipos a comparar.

En el caso de la base de datos para el método de vectores se utiliza una interfaz hecha con el único fin de obtener esta base de datos, como habíamos mencionado anteriormente para un carácter tenemos más de un vector representativo, pero al manejar los datos de forma que a cada etiqueta le corresponde un solo vector, tendremos una base de datos no relacional, sin tener la necesidad de desarrollar un lenguaje de búsqueda estructurada, a pesar de que esta base de datos no es la que requiere menos memoria si es la que nos representa de una mejor manera las posibles variaciones de la escritura del carácter, además de tener la ventaja de que al encontrarse dentro del programa (código), no realizamos una búsqueda en un archivo externo, lo que vuelve más rápido al sistema, con la desventaja de que el sistema pierde independencia con relación a la base de datos.

Conclusiones sobre la comparación y el reconocimiento

La comparación se fundamenta principalmente en la obtención de la diferencia entre las características extraídas para los caracteres prototipo y las extraídas para el carácter a reconocer (carácter prueba). Por otro lado, el reconocimiento se basa en la identificación del carácter que menos difiere con respecto a los caracteres prototipo. Para la cuantificación de esa diferencia, se emplearon distintos procedimientos, de acuerdo a los métodos de reconocimiento utilizados.

Método por Envolventes

Para la comparación utilizada en el Método de Reconocimiento por Envolventes, fueron buenos los resultados obtenidos para ciertas letras, principalmente para aquellas cuyas envolventes presentan mayores variaciones, como en los casos de las vocales “e”, “i”, “u”, sin embargo, para las letras “a” y “o”, ese resultado se volvía confuso. La eficiencia hasta ese momento era del orden del 90%, para las vocales “e”, “i”, “u”, y del 60% para las letras “a” y “o”. Para mejorar ese resultado, en este último caso se adaptó un proceso posterior de validación del reconocimiento. Este proceso, incluye la extracción de un parámetro denominado “posición final del trazo”, considerando que para el caso de una vocal “a” el trazo final se encuentra en una posición inferior a la de la letra “o”. Con la adaptación de ese proceso, la eficiencia general se incrementó al orden del 90%. Por ello, concluimos que aunque este método por sí solo no alcanza una eficiencia aceptable, debido a la gran semejanza existente entre las envolventes para ciertas letras, el método es conveniente para lograr la diferenciación en un conjunto pequeño de caracteres, como ocurrió en los casos de la “e” y la “a”, o la “i” y la “o”. Si las necesidades de reconocimiento no involucran muchos caracteres, este método es recomendable aunado a un proceso de validación final que distinga de entre un conjunto mucho más pequeño de letras. En nuestro caso, este método no fue conveniente debido a que nuestras necesidades de reconocimiento iban un poco más allá.

Métodos por Trayectorias

Los resultados obtenidos para los procesos de comparación y reconocimiento utilizados en el Método de Reconocimiento por Trayectorias, fueron del orden del 90%, sin embargo el proceso es largo, pues deben extraerse 16 parámetros. Además el método es aplicado nuevamente a las

envolventes del carácter, que si bien varían bastante para ciertas letras, para otras no, ocasionando la repetición o aproximación de ciertos vectores, lo que ocasiona errores en los resultados. En principio, para las letras “a”, “e”, y “o”, la eficiencia del reconocimiento era de 70%, y para las vocales “e” e “i”, del orden del 95%. Nuevamente fue necesario aplicar un proceso de validación, con lo cual la eficiencia alcanzada fue de más del 95%. Sin embargo, se observa que al incrementar los elementos dentro del conjunto de caracteres a reconocer, dicha eficiencia tiende a disminuir. Este método puede aplicarse al reconocimiento de un conjunto pequeño de caracteres, tales como el de vocales, obteniendo una buena eficiencia. También aplicamos este método, como prueba, a funciones obtenidas del recorrido de la pluma al escribir. Para las pocas pruebas que realizamos, la eficiencia fue alta, debido a que las variaciones entre esas funciones son aún mayores. Se espera que este mismo método aplicado a ese tipo de entradas, logre buenos resultados para un conjunto mayor de caracteres. Sin embargo, en el sistema, no se utilizó esto último, debido a que este sistema está pensado para la captura de una imagen de la letra por medio de una tableta digital. Por lo que respecta al Método de Reconocimiento por Trayectorias este fue utilizado en el sistema solo para el reconocimiento de vocales, debido a que, como se menciona en un principio, el método es extenso en comparación con los otros, además de que requiere de un proceso de validación. Por otro lado, se realizó una optimización de este método, cambiando la base de datos de imágenes por una de vectores para agilizarlo, sin embargo, en cuanto a los resultados de reconocimiento obtenido, éstos fueron prácticamente iguales.

Método por Vectores

La eficiencia obtenida para este método fue mayor al 90% para todas las vocales. La extracción de los parámetros se basa en barridos de la imagen buscando cierta información por zonas. No requiere de un método extra de validación. Sólo se extraen 7 parámetros que permiten el reconocimiento de un conjunto amplio de caracteres. Al realizar pruebas incrementando el conjunto de caracteres a reconocer (hasta 13), la eficiencia fue mayor al 90%. En lo que respecta a su base de datos, ésta es de vectores, así que sólo se aplica la extracción para el carácter a reconocer, haciendo más rápido el reconocimiento. Además este método permite la adaptación de los posibles corrimientos del carácter, debidos al proceso de segmentación, que es el problema principal al aplicar el reconocimiento a un carácter proveniente de una palabra segmentada. Por lo anterior, este método fue elegido para aplicarse en el sistema al reconocimiento tanto de vocales como de palabras, logrando para el segundo caso una eficiencia del 80%, para el reconocimiento de la totalidad de la palabra, y de más del 95% para el reconocimiento parcial de más de la mitad de los caracteres que la forman. Además, se logró el reconocimiento no sólo de las palabras del diccionario, sino también de todas aquellas que tienen una extensión de 6 caracteres, escritos con las letras que forman los nombres de los números dígitos, debido a que el reconocimiento de la palabra se realiza mediante el reconocimiento de cada carácter que la forma. Este método nos permitió cumplir con los objetivos planteados para esta tesis, sobrepasándolos debido al incremento en el conjunto de palabras que el sistema puede reconocer, lo que deja abierta la posibilidad de incrementar aún más el conjunto de caracteres de reconocimiento, teniendo la posibilidad de ampliarlo hasta abarcar todas las letras del abecedario, con lo cual podría reconocerse cualquier palabra con una alta eficiencia.

Comparación con otros trabajos

Como sabemos, la cantidad de información respecto al tema de nuestro estudio es enorme, por lo que la comparación de nuestro trabajo con todas las fuentes disponibles es sumamente difícil. Aunque la cantidad de referencias y bibliografía dadas al final de este estudio no es pequeña, es sólo

una parte del total de referencias encontradas en fuentes como la IEEE (Institute of Electrical Electronic Engineers), ya que en otras fuentes seguramente se encuentra una cantidad similar de información.

Dentro de dichas referencias, encontramos algunos estudios con propósitos similares al nuestro; entre ellos el expuesto en el artículo de Andrew W. Senior [20], donde se muestra el funcionamiento de un sistema de reconocimiento de escritura cursiva, que considera la captura de escritura mediante escaneado, normalización y parametrización, reconocimiento mediante una red neuronal recurrente, y un modelo de Markov de lenguaje para reconocer palabras. Este artículo es importante para nosotros puesto que el objetivo de realizar el proceso completo, desde la captura hasta el reconocimiento, es parecido al nuestro, con sus respectivas diferencias.

En dicho trabajo se alcanza una tasa de eficiencia de 87% en la tarea de reconocer un vocabulario abierto, y se menciona que otros autores han conseguido, para otros sistemas (para varios tamaños del vocabulario) tasas de error de 52% [63], 50% [64], y 30 % [65]. Como ya hemos analizado, nuestras tasas de eficiencia son mayores al 80% en cualquier caso, lo cual es muy satisfactorio para este tipo de trabajos, considerando la metodología usada, y tomando en cuenta el reducido tamaño de nuestro vocabulario, aunque debemos mencionar que también podemos reconocer palabras que estén fuera de él, y que puedan ser formadas por las letras contempladas en nuestro estudio, y de un tamaño no mayor a 6 caracteres. En la conclusión de dicho artículo [20] se menciona que es difícil la comparación de resultados con otros investigadores, a causa de las diferencias en los detalles experimentales, el tipo de escritura analizada, y el método de recolección de datos.

Como se observa en las tablas de resultados del Capítulo 7, los porcentajes de eficiencia que obtuvimos en el reconocimiento de caracteres son superiores al 90% en todos los casos, mientras que el reconocimiento de palabras tiene una eficiencia del 80%. Estos resultados son muy favorables, y debemos explicar las razones de estas tasas de eficiencia. En el artículo de W. Senior [20] que mencionamos, una conclusión importante es que una parte del éxito de los algoritmos de reconocimiento se debe al tamaño del diccionario a reconocer, esto es, la cantidad de palabras, caracteres, números, entre otros.

En nuestro caso, mientras menor es el diccionario, mayor es la eficiencia, y viceversa, ya que tenemos 13 caracteres y 10 palabras por reconocer, y la cantidad de caracteres no reconocidos y mal reconocidos es baja. Esto se ve claramente en los algoritmos de reconocimiento de caracteres aislados, en los cuales, eliminando las consonantes y sus criterios de reconocimiento aumentó la eficiencia de los mismos. Sin embargo, el algoritmo de reconocimiento por vectores demostró una buena eficiencia al reconocer vocales y consonantes, aunque su eficiencia no haya sido la mayor de todos los métodos.

Respecto al trabajo de García Navarrete [26] que mencionamos en el Capítulo 3, al evaluar sus métodos de segmentación basados en características geométricas y densidades de píxeles obtiene 74% de éxito en los mejores de los casos, probados sobre 40 palabras. En nuestro trabajo, obtuvimos una eficiencia de 97.92% con el método de segmentación por localización de extremos superiores, lo cual nos habla de un mejor desempeño de nuestro algoritmo, el cual también se basa en características geométricas de la escritura. La principal diferencia entre ambos trabajos es que las palabras de nuestro estudio se basan en caracteres prototipo pre-establecidos, cuyas características son consideradas en nuestro algoritmo de segmentación y a las cuales debe ajustarse el usuario de

nuestro sistema, mientras que el de García Navarrete se basa en escritura procedente de telegramas, la cual requiere un proceso previo de normalización. Otra ventaja de nuestro sistema es que el proceso de segmentación, así como el resto del sistema es automático, mientras que en dicho trabajo [26] la segmentación es un proceso semiautomático, en el cual, sobre la imagen de la escritura, un usuario coloca marcas donde cree que hay puntos de segmentación en la palabra.

Por último, en el artículo de Charles C. Tappert [16] se hace un análisis sobre los resultados del reconocimiento de palabras cursivas sobre cuatro métodos distintos, uno con segmentación externa, otro más con una segmentación especial, y dos con segmentación interna. Para reducir la dificultad del problema sólo se consideraron letras minúsculas en dichos sistemas. Los resultados de estos estudios tienen un porcentaje de eficiencia alrededor del 90% con un diccionario de 26 caracteres. En nuestro caso, obtuvimos una tasa de 80% en el reconocimiento de palabras con un diccionario de 13 caracteres. Este porcentaje es muy bueno, considerando que en dichos estudios los métodos de reconocimiento empleados (análisis de características, pareado de curvas (elastic matching) de direcciones y alturas, procesamiento jerárquico) son mucho más complejos que los nuestros, lo cual habla sobre la ventaja que le da a nuestro algoritmo ser sencillo pero eficiente. En este artículo también se menciona que las aproximaciones hechas son útiles para vocabularios pequeños.

Conclusiones generales

- En base a las comparaciones de la sección anterior, podemos decir que parte importante del éxito de nuestro sistema se debe a consideraciones como las anteriores, es decir, un reducido vocabulario, el uso exclusivo de letras minúsculas y criterios eficientes de distinción de caracteres. Sin embargo, como es evidente, la efectividad de los algoritmos que desarrollamos se debe de igual modo a una buena selección de las características que distinguen caracteres, una buena segmentación, el uso de prototipos promedio, el periodo de entrenamiento, entre otros. De lo anterior podemos concluir que nuestro sistema de reconocimiento reúne algunas de las mejores cualidades respecto a los otros sistemas evaluados, y que el desempeño fue mejor en ciertos casos, ya que se debe tener presente que en muchos de los estudios que revisamos el tamaño del vocabulario a reconocer es mucho más grande, lo cual incrementa directamente la dificultad en la distinción de la identidad de cada carácter o palabra, entre otros problemas.
- Un factor muy importante que contribuye a que los resultados de los algoritmos de reconocimiento, tanto de palabras como de letras, sean tan altos, es que debe considerarse un periodo de entrenamiento por parte de los usuarios del sistema. Es decir, éstos deben acostumbrarse a escribir con la pluma y la tableta digitalizadora, las cuales producen sensaciones diferentes al escribir sobre la tableta, ya que ésta no ofrece la misma fricción que una hoja de papel y una pluma comunes. Además, la posición y forma de ambos instrumentos pueden llegar a ser incómodas para algunos escritores, por lo que algunos pueden llegar a preferir el uso del ratón, por estar más habituados a su uso.
- Un elemento más al que debe uno acostumbrarse es al tipo de escritura cursiva que se maneja en este estudio, el cual determinamos a través de la obtención de prototipos, los cuales promedian las características de la forma de escritura de varios autores, a fin de hacerla más representativa y reducir de esta manera sus diferencias con la escritura de una persona en específico.

- Aunque nuestros algoritmos están basados en métodos heurísticos (en este caso, esto quiere decir que no se basan en fundamentos matemáticos o teorías y métodos ya establecidos) demostraron tener altas tasa de eficiencia, aún al ser comparados con otros sistemas. Al no emplear herramientas de uso frecuente en este tipo de problemas, tales como modelos de Markov, redes neuronales, entre otros, este es un trabajo que propone una estrategia original en la manera de atacar un problema que ha sido resuelto mediante complejos modelos matemáticos y sofisticadas herramientas de programación por otras personas. Sin embargo, debemos insistir en que la comparación directa con otros sistemas es difícil, debido a la diferencia de objetivos, alcances y herramientas de cada problema en particular.
- Este es un trabajo valioso, ya que en nuestro país no se hacen muchos trabajos de este tipo. La mayor parte de los trabajos que revisamos, disponibles a través del sitio en Internet de la IEEE, se encuentran publicados en inglés, y fueron realizados en instituciones de América del Norte, Asia y Europa. Como mencionamos en el Capítulo 3, un trabajo en una línea similar es el de García Navarrete [26], desarrollado en la Universidad de las Américas, Puebla, entre otros. Sin embargo, creemos que los pocos o muchos trabajos enfocados al reconocimiento de escritura, desarrollados en cualquier dependencia en nuestro país, deben ser difundidos, con la finalidad de conocer el trabajo de otras personas y poder comparar métodos, líneas de investigación y resultados.

Perspectivas

- Como parte de un post-procesamiento de la escritura, a la salida del bloque de reconocimiento, tenemos palabras con algunas letras que no fueron reconocidas. Para incrementar el número de caracteres reconocidos en una palabra, y por lo tanto, el número de palabras totalmente reconocidas, puede aplicarse RIC (Reconocimiento Inteligente de Caracteres), el cual consiste en técnicas de reconocimiento de palabras que utilizan modelos lingüísticos, tales como léxicos o información estadística sobre las secuencias de palabras.
- La posibilidad de que la captura pueda realizarse a una mayor velocidad sin perder información, con lo que se podría dar un grado más de libertad (o naturalidad) a la escritura. Además es posible, utilizando otro tipo de tableta, visualizar los trazos en la pantalla al mismo tiempo que sobre la zona de escritura de la tableta digitalizadora o electrónica. Respecto a los controladores del sistema, los requerimientos no necesitan una mayor complejidad.
- Se abren muchas perspectivas respecto a la interfaz gráfica, ya que en mucho depende el enfoque que se le puede dar al sistema, como puede ser para recuperar textos manuscritos y editarlos en computadora, enseñanza y práctica de la escritura, restauración de textos dañados, etc.
- Una perspectiva de trabajo futuro sobre las mismas líneas de nuestro trabajo es el reconocimiento de todas las letras del alfabeto, así como un mayor número de palabras, lo cual implica extraer características y desarrollar criterios útiles para su distinción. Del mismo modo, aumentando el área de captura de la interfaz, o bien, con un sistema distinto de captura interactiva de la escritura pueden reconocerse no sólo palabras, sino frases o párrafos, incluyendo signos de puntuación, números y signos, lo cual puede ser útil en la recolección automática de notas a mano, con un proceso como el explicado para pasar la escritura

reconocida en un documento electrónico de texto, tal como un block de notas o un documento de Word.

- Algunas otras aplicaciones de este sistema y otros similares son: utilización en un servicio de información como base para analizar solicitudes de información y acoplarlas a una base de datos de texto o imágenes, a fin de seleccionar la información precisa; generación de correspondencia y demás documentación comercial; producción y gestión de documentación; servicios de traducción asistida por computadora; traducción en línea de comunicaciones electrónicas; globalización y localización de sistemas informáticos y de sus interfaces de usuario. La mayor parte de estas aplicaciones puede disponer de interfaces de usuario en lenguaje natural, por ejemplo el habla, para incrementar su utilidad.

En conclusión, podemos agregar que en la elaboración de este trabajo, tanto en la investigación documental como en la propuesta y desarrollo de nuestros algoritmos, nos dimos cuenta del enorme número de investigadores que trabajan en el área del reconocimiento de escritura, así como de las dificultades que implica la comparación de nuestro trabajo con trabajos desarrollados por otras personas, dada la gran cantidad de fuentes y de información que puede obtenerse de ellas. Finalmente, el mensaje principal que queremos transmitir con este trabajo es que tanto en nuestra Universidad como en nuestro país deben hacerse más trabajos de este tipo, que puedan ser aprovechables por otras personas, ya sea como aportaciones originales o bien, como una continuación de nuestro trabajo, el cual ofrece muchas perspectivas de seguimiento y mejora de algoritmos y demás elementos que lo conforman.

Referencias y Bibliografía

- [1] Graphics File Formats, Kay David C., Levine John L., McGraw Hill, 1992.
- [2] M. Bern, D. Goldberg, “Scanner-model-based document image improvement”, Proceedings of the 2000 IEEE International Conference on Image Processing, vol. 2, pp. 582-585, Septiembre 10-13 de 2000.
- [3] T. Yamasaki, T. Hattori, “A new data tablet system for handwriting characters and drawing based on the image processing”, IEEE International Conference on Systems, Man, and Cybernetics, vol. 1, pp. 428-431, Octubre 14-17 de 1996.
- [4] K. Kimura, M. Kanoh, K. Kawashima, K. Mohri, M. Takagi, L.V. Panina, “Data tablet utilizing Matteucci effect of amorphous magnetic wire matrix”, IEEE Transactions on Magnetics, vol. 27, no.6, pp. 4861-4863, Noviembre de 1991.
- [5] J. Bresenham, “Pixel-processing fundamentals”, IEEE Transactions on Computer Graphics and Applications, vol. 16, no. 1, pp. 74-82, IEEE Journal (Tutorial), Enero de 1996.
- [6] V. Lacroix, “The primary raster: a multiresolution image description”, Proceedings of the 10th IEEE International Conference on Pattern Recognition, vol. 1, pp. 903-907, Junio 16-21 de 1990.
- [7] C.S. Foshee, “Mixed image format environments: comparisons, tradeoffs, and issues”, Proceedings of the Fourth International Conference on Image Management and Communications, pp. 269-276, Agosto 20-24 de 1995.
- [8] H. Stoldt, A.J.P. Theuwissen, F.F. Vledder, P.G.M. Centen, A. Mierop, A.C.M. Kleimann, H.L. Peek, D.W.E. Verbugt, P.B. Hartog, R.H.S. de Gruyter, “CCD imagers for broadcast applications”, International Electron Devices Meeting (invited paper), pp. 899-902, Diciembre 8-11 de 1996.
- [9] Laurence O’Gorman, Rangachar Kasturi, Venu Govindaraju, “Document image analysis: A primer,” Sadhana, vol. 27, issue 1, pp. 3–22, publicación bimestral, impreso en India, Febrero de 2002.
- [10] Chun Wong, A.M.K. Cheng, “An approach for imprecise transmission of TIFF image files through congested real-time ATM networks”, Proceedings of the 22nd Annual Conference on Local Computer Networks, pp. 420-429, Noviembre 2-5 de 1997.

- [11] W. Lovegrove, D. Elliman, "Text block recognition from TIFF images", IEEE Colloquium on Document Image Processing and Multimedia Environments, pp. 4/1-4/6, Noviembre 2 de 1995.
- [12] M. Livny, V. Ratnakar, "Quality-controlled compression of sets of images", Proceedings of International Workshop on Multimedia Database Management Systems, pp. 109-114, Agosto 14-16 de 1996.
- [13] C.C. Hsieh, H.W. Chao, B. Chen, P.H. Shih, "Automatic interpretation of cadasters by image analysis techniques", Proceedings of the IEEE International Conference on Image Processing, ICIP-94, vol. 3, pp. 202-206, Noviembre 13-16 de 1994.
- [14] D.J. Jackson, S.J. Hannah, "Comparative analysis of image compression techniques", Proceedings of the Twenty-Fifth Southeastern Symposium on System Theory, SSST '93, pp. 513-517, Marzo 7-9 de 1993.
- [15] Sukhan Lee, Jack C. Pan, "Offline tracing and representations of signatures", IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 4, pp. 755 - 771, Julio-Agosto de 1992.
- [16] Charles C. Tappert, Ching Y. Suen, Toru Wakahara, "The State of the Art in On-Line Handwriting Recognition", IEEE Transactions on Pattern Analysis and machine Intelligence, vol. 12, no. 8, pp. 787-805, Agosto de 1990.
- [17] T. Caesar, J.M. Gloer, E. Mandler, "Preprocessing and feature extraction for a handwriting recognition system", Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 408-411, Octubre 20-22, 1993.
- [18] Ke Han, I.K. Sethi, "Off-line cursive handwriting segmentation", Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 2, pp. 894-897, Agosto 14-16 de 1995.
- [19] E. Anquetil, G. Lorette, "Perceptual model of handwriting drawing application to the handwriting segmentation problem", Proceedings of the Fourth International Conference on Document Analysis and Recognition, vol. 1, pp. 112-117, Agosto 18-20 de 1997.
- [20] Andrew W. Senior, Anthony J. Robinson, "An off-line Cursive Handwriting Recognition System", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 309-321, Marzo de 1998.
- [21] The Student Edition of MATLAB / The MathWorks, The MATLAB curriculum series, Versión 4 para Microsoft Windows, Englewood Cliffs, New Jersey, E.U.A., Prentice Hall, 1995.
- [22] Fundamentals of Signals and Systems Using MATLAB, Edward W. Kamen, Bonnie S. Heck, primera edición, Upper Saddle River, New Jersey, E.U.A., Prentice Hall, 1997.
- [23] Thomas Bayer, "State of the Art in Off-Line Handwriting Applications", IEE Third European Workshop on Handwriting Analysis and Recognition (Ref. No. 1998/440), pp. 14/1 - 14/6, Julio 14-15 de 1998.

- [24] Graphics and GUIs with MATLAB, Patrick Marchand, O. Thomas Holland, tercera edición, Boca Ratón, E.U.A., CRC Press, 2003.
- [25] Richard G. Casey, Eric Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pp. 690-706, Julio de 1996.
- [26] Jorge García Navarrete, "Mejora en el algoritmo de segmentación para el reconocimiento de caracteres de telegramas escritos por el Gral. Porfirio Díaz", Tesis profesional, Universidad de las Américas, Puebla, México, Enero de 2003.
- [27] Rafael Palacios, Amar Gupta, Patrick S.P. Wang, "Feedback-based architecture for reading courtesy amounts on checks", IEEE Journal of Electronic Imaging, vol. 12, no. 1, pp. 194-202, Enero de 2003.
- [28] Chandra J. Weliwitige, Alan Harvey, Andrew Jennings, "Use of Hough Transform for Handwritten Character Slant detection", Proceedings of the 11th Conference of the International Graphonomics Society (IGS2003), Teulings, H.L., Van Gemmert, A.W.A. Editores, Scottsdale, Arizona, EUA, 2-5 de Noviembre de 2003.
- [29] Francisco Pazo Sánchez, Pablo Maldonado López, Manuel Argenis Rodríguez Gutiérrez, "Reconocimiento de Escritura Manuscrita. Caso Práctico: Un reconocedor de vocales", Apuntes de clase, Secretariado de Recursos Audiovisuales y Nuevas Tecnologías de la Universidad de Sevilla, España, Agosto de 2002.
- [30] Professional Group C5 (Human-computer interaction), IEE Colloquium on "What are graphical user interfaces good for?", (Digest no. 1994/021), Computing and Control Division, Londres, Reino Unido, Jueves 27 de Enero de 1994.
- [31] Sahnoun Zaidi, Medghoul Reguia, "Fuzzy Data Base System: A Unified Approach", IEEE International Conference on Humans, Information and Technology, vol. 1, pp. 794 – 799, 2-5 de Octubre de 1994.
- [32] Gary E. Kopec, "The signal data base system SDB", IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '84, vol. 9, pp. 154 – 157, Marzo de 1984.
- [33] S. Manke, U. Bodenhausen, "A connectionist recognizer for on-line cursive handwriting recognition", IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-94, vol. 2, pp. II/633 - II/636, 19-22 de Abril de 1994.
- [34] C.A. Higgins, P.E. Bramall, "A system for cursive handwriting recognition based on human reading models", IEE Colloquium on Handwriting and Pen-Based Input, pp. 7/1 - 7/7, 1994.
- [35] Ma Lin, Li Haifeng, Han Jiqing, P. Gallinari, "Hidden control neural network and HMM hybrid approach for on-line cursive handwriting recognition", Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, vol. 1, pp. 236 – 239, 14-17 de Diciembre de 2003.

- [36] R. Seiler, M. Schenkel, F. Eggimann, "Off-line cursive handwriting recognition compared with on-line recognition", Proceedings of the 13th International Conference on Pattern Recognition, vol. 4, pp. 505 – 509, 25-29 de Agosto de 1996.
- [37] G. Rigoll, A. Kosmala, J. Rattland, C. Neukirchen, "A comparison between continuous and discrete density hidden Markov models for cursive handwriting recognition", Proceedings of the 13th International Conference on Pattern Recognition, vol. 2, pp. 205 – 209, 25-29 de Agosto de 1996.
- [38] Y.H. Tay, M. Khalid, R. Yusof, C. Viard-Gaudin, "Offline cursive handwriting recognition system based on hybrid Markov model and neural networks", Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, vol. 3, pp. 1190 – 1195, 16-20 de Julio de 2003.
- [39] Wenwei Wang, A. Brakensiek, A. Kosmala, G. Rigoll, "Multi-branch and two-pass HMM modeling approaches for off-line cursive handwriting recognition", Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 231 – 235, 10-13 de Septiembre de 2001.
- [40] G. Menier, G. Lorette, P. Gentric, "A genetic algorithm for on-line cursive handwriting recognition", Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, Conference B: Computer Vision & Image Processing, pp. 522 – 525, 9-13 de Octubre de 1994.
- [41] IEE Colloquium on "Handwriting and Pen-Based Input" (Digest no. 1994/065), 1994.
- [42] S. Procter, J. Illingworth, F. Mokhtarian, "Cursive handwriting recognition using hidden Markov models and a lexicon-driven level building algorithm", IEE Proceedings of Vision, Image and Signal Processing, vol. 147, issue 4, pp. 332 – 339, Agosto de 2000, IEE JNL.
- [43] J. Ruiz-Pinales, E. Lecolinet, "Cursive handwriting recognition using the Hough transform and a neural network", Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, pp. 231 – 234, 3-7 de Septiembre de 2000.
- [44] T. Varga, H. Bunke, "Generation of synthetic training data for an HMM-based handwriting recognition system", Proceedings of the Seventh International Conference on Document Analysis and Recognition, vol. 1, pp. 618 – 622, 3-6 de Agosto de 2003.
- [45] Ling Gang, B. Verma, S. Kulkarni, "Experimental analysis of neural network based feature extractors for cursive handwriting recognition", Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN '02, vol. 3, pp. 2837 – 2842, 12-17 de Mayo de 2002.
- [46] G. Rigoll, A. Kosmala, D. Willett, "A new hybrid approach to large vocabulary cursive handwriting recognition", Proceedings of the Fourteenth International Conference on Pattern Recognition, vol. 2, pp. 1512 – 1514, 16-20 de Agosto de 1998.

- [47] S. Manke, M. Finke, A. Waibel, "NPen++: a writer independent, large vocabulary on-line cursive handwriting recognition system", Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 403 – 408, 14-16 de Agosto de 1995.
- [48] B.A. Yanikoglu, P.A. Sandon, "Recognizing off-line cursive handwriting", Proceedings CVPR '94, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 397 – 403, 21-23 de Junio de 1994.
- [49] N. Arica, F.T. Yarman-Vural, "Optical character recognition for cursive handwriting", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, issue 6, pp. 801 – 813, Junio de 2002.
- [50] A.K. Jain, R.P.W. Duin, Jianchang Mao, "Statistical pattern recognition: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, issue. 1, pp. 4 – 37, Enero de 2000.
- [51] T. Haifeng Li Artieres, P. Gallinari, B. Dorizzi, "Topology inference for an ANN/HMM hybrid on-line handwriting recognition system", Proceedings of the 9th International Conference on Neural Information Processing, ICONIP '02, vol. 5, pp. 2479 – 2483, 18-22 de Noviembre de 2002.
- [52] B. Verma, "A contour character extraction approach in conjunction with a neural confidence fusion technique for the segmentation of handwriting recognition", Proceedings of the 9th International Conference on Neural Information Processing, ICONIP '02, vol. 5, pp. 2459 – 2463, 18-22 de Noviembre de 2002.
- [53] Roongroj Nopsuwanchai, D. Povey, "Discriminative training for HMM-based offline handwritten character recognition", Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 114 – 118, 3-6 de Agosto de 2003.
- [54] J. Ruiz-Pinales, E. Lecolinet, "A new perceptive system for the recognition of cursive handwriting", Proceedings of the 16th International Conference on Pattern Recognition, vol. 3, pp. 53 – 56, 11-15 de Agosto de 2002.
- [55] M. Parizeau, A. Lemieux, C. Gagne, "Character recognition experiments using Unipen data", Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 481 – 485, 10-13 de Septiembre de 2001.
- [56] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, G. Rigoll, "Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data", Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR '99, pp. 446 – 449, 20-22 de Septiembre de 1999.
- [57] S. Manke, M. Finke, A. Waibel, "Combining bitmaps with dynamic writing information for on-line handwriting recognition", Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2 - Conference B: Computer Vision & Image Processing, pp. 596 – 598, 9-13 de Octubre de 1994.

- [58] D.G. Elliman, L. Bai, "Using tabu search for segmentation of cursive handwriting", Seventh International Conference on Image Processing and Its Applications (Conf. Publ. No. 465), vol. 2, pp.745 – 749, 13-15 de Julio de 1999.
- [59] R. Plamondon, C.M. Privitera, "The segmentation of cursive handwriting. an approach based on off-line recovery of the motor-temporal information", IEEE Transactions on Image Processing, vol. 8, issue 1, pp. 80 – 91, Enero de 1999.
- [60] Jiang Gao, Xiaoqing Ding, "A method for segmentation of cursive handwritings and its application to character shape extraction", Proceedings of the 2000 International Conference on Image Processing, vol. 2, pp. 696 – 699, 10-13 de Septiembre de 2000.
- [61] K. Fukushima, T. Imagawa, E. Ashida, "Character recognition with selective attention", IJCNN-91-Seattle International Joint Conference on Neural Networks, vol. 1, pp. 593 – 598, 8-14 de Julio de 1991.
- [62] Alejandro Héctor Toselli, "Reconocimiento de Texto Manuscrito Continuo", Tesis doctoral, Universidad Politécnica de Valencia, España, 20 de marzo de 2004.
- [63] R.M. Bozinovic, S.N. Srihari, "Off-Line Cursive Word Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 68–83, Enero de 1989.
- [64] S. Edelman, S. Ullman, T. Flash, "Reading Cursive Script by Alignment of Letter Prototypes," Int'l J. Computer Vision, vol. 5, no. 3, pp. 303–331, 1990.
- [65] B.A. Yanikoglu, P.A. Sandon, "Off-Line Cursive Handwriting Recognition Using Style Parameters," Reporte técnico PCS-TR93-192, Dartmouth College, Abril de 1993.