

Reconocimiento Automático de Texto Braille

Héctor Ferraro, Claudia Russo

hpferraro@hotmail.com

crusso@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática LIDI
Facultad de Informática. Universidad Nacional de La Plata.
La Plata, Buenos Aires, Argentina.

*IV Workshop de Computación Gráfica, Imágenes y Visualización
XII Congreso Argentino de Ciencias de la Computación CACIC 2006*

Abstract

This job presents the development of ad-hoc methods to generate a tool to allow recognize the characters of a Braille text and traduce those in digital text. This kind of tool me be useful to reprint Braille documents, as help to learn Braille, as soon as medium of integration for blind students in all levels or integration in the social boundary of blind people. For this, we develop, in others, two ad-hoc methods: one to segments the scanned image and another to detects the inclination angle of the Braille document at the scanning moment.

Key words: **Digital Images, Automatic recognize, Digital image procesing.**

Este trabajo presenta el desarrollo de métodos ad-hoc para generar una herramienta que permita reconocer los caracteres de un texto Braille y traducirlos a texto digital. Este tipo de herramienta puede ser útil para reimprimir documentos Braille, como ayuda en el aprendizaje de Braille o como medio de integración para estudiantes no videntes en todos los niveles o integración en el ámbito social, de personas no videntes. Para esto, se desarrollo entre otros, un método que segmenta la imagen escaneada a ser tratada y otro para detectar la inclinación de la hoja en el momento del escaneo.

Palabras clave: **Imágenes digitales, Reconocimiento automático, Procesamiento digital de imágenes.**

1. Introducción

La habilidad de reconocer letras y dígitos (caracteres) es fundamental para interpretar lenguajes impresos. Para una computadora un carácter en una página no es más que otra imagen u objeto a ser reconocido. El reconocimiento óptico de caracteres consiste en reconocer de forma automática, letras, dígitos, o algún otro símbolo, de imágenes que han sido ópticamente exploradas en líneas horizontales (imágenes raster).

Braille es el sistema de lecto-escritura que utilizan las personas no videntes. Fue inventado por el francés Luis Braille en 1870. Un documento Braille está formado por una serie de caracteres separados a una distancia standard. Cada carácter está compuesto de seis puntos que pueden estar activos o no, es decir presentar salientes en el papel (erupciones) o no. Cada combinación de puntos activos y no activos en un carácter representa un símbolo distinto. De esta forma, con seis puntos se pueden representar 64 símbolos distintos. Los documentos braille pueden ser escritos a una cara o a doble cara, esta última forma se conoce como Braille inter-punto [3] [4].

El caso del reconocimiento de texto Braille tiene diferencias y similitudes con el reconocimiento óptico de caracteres: Hay que resolver el problema de la inclinación de la hoja. Se utilizan técnicas de umbralización de la imagen para segmentar la imagen. Con respecto a las diferencias, en el caso de reconocimiento de texto Braille se intenta detectar la malla (todos los posibles puntos Braille de la hoja) para luego ver cuales puntos están activos y cuales no y para el reconocimiento propiamente dicho no hace falta utilizar técnicas de clasificación de los objetos a reconocer.

A continuación se plantea un esquema de solución al problema planteado. Se describirán las etapas y algoritmos desarrollados para reconocer los caracteres Braille de un documento Braille y transformarlos en texto digital.

2. Solución planteada:

En la Figura 1 pueden verse las etapas para el reconocimiento de texto braille:

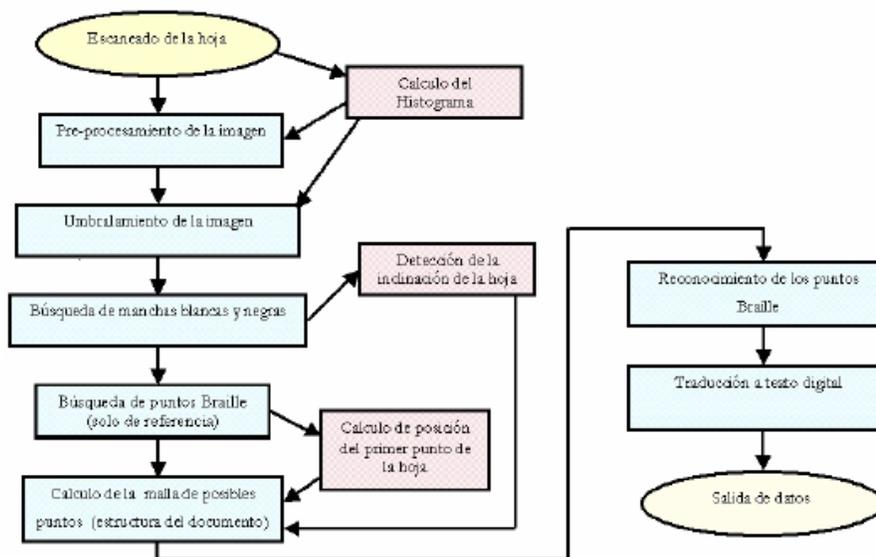


Figura 1. Imagen después de aplicar los umbrales

2.1 Escaneado del documento

Para realizar el escaneado se utilizó el estándar Twain. Este estándar provee una API que puede ser utilizada desde una aplicación determinada. De esta forma la aplicación desarrollada se independiza del escáner y se asegura que funcionará con cualquier escáner que trabaje con Twain (actualmente un gran número de ellos) [8]. Se trabajó con imágenes en escala de grises. La resolución de escaneado recomendada es de 150 dpi.

Características de la imagen escaneada:

Al escanear un documento braille se observan unas zonas brillantes, unas más oscuras y el fondo es de un brillo intermedio. Un punto Braille está formado por una zona brillante y una oscura. Esto queda ilustrado en la Figura 2.

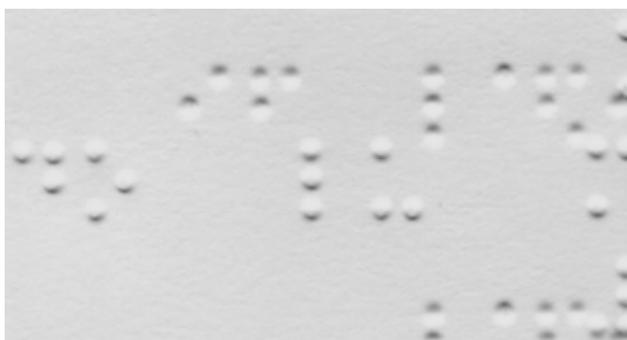


Figura 2. Un trozo de un documento braille escaneado

2.2 Preprocesamiento de la imagen

Es importante disponer de una buena imagen de entrada. Esta etapa puede incluir un conjunto de varias operaciones sobre la imagen adquirida con el escáner a fin de dejar disponible para las etapas posteriores una imagen que pueda ser mejor procesada e interpretada:

- Si la imagen no está en escala de grises habrá que convertirla.
- Si es necesario habrá que ajustar el contraste de la misma.
- Si es necesario habrá que aplicar algún filtro para minimizar el ruido introducido por el escáner al momento de la adquisición.

2.3 Cálculo del histograma

En esta etapa disponemos de la imagen en escala de grises. Se calcula el histograma de luminancias de la misma. Calcular el histograma consiste en contar la cantidad de píxeles de la imagen para cada nivel de gris. Los niveles de gris están en el rango 0..255 (escala de grises). Esta información se guarda en una estructura de datos tipo arreglo a fin de dejarla disponible para ser usada por las etapas posteriores [1].

2.4 Umbralamiento de la imagen

En esta etapa se utilizan la imagen resultado de la etapa de pre-procesamiento y el histograma de luminancias antes calculado. Se aplica sobre la imagen un algoritmo de doble umbral obteniendo como resultado una imagen a tres colores.

Para hallar los umbrales se utiliza el histograma de luminancias calculado anteriormente. Se toman los puntos que dejan por debajo/encima un porcentaje determinado del área total del histograma (colas del histograma). Se han obtenido buenos resultados con un porcentaje del 3 % (para documentos a una cara), de los cuales 1% es la cola izquierda y 2% es la cola derecha del mismo [1] [5].

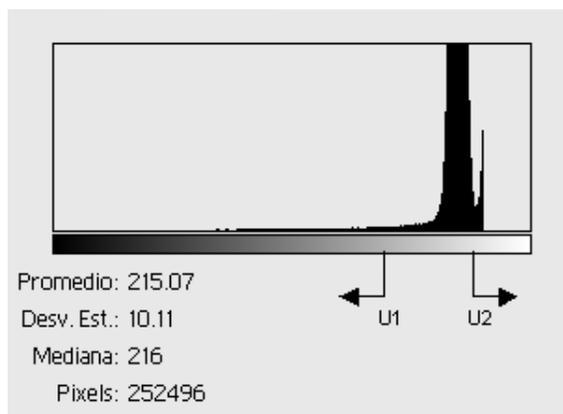


Figura 3. Histograma de luminancias de la imagen original

U1 es el nivel de gris para el cual, el área debajo de la curva que viene dada por la función histograma desde 0 hasta dicho punto, representa el 1% de la cantidad total de píxeles de la imagen.

U2 es el nivel de gris para el cual, el área debajo de la curva que viene dada por la función histograma desde dicho punto hasta 255, representa el 2% de la cantidad total de píxeles de la imagen.

Una vez obtenidos los umbrales se aplica a cada píxel de la imagen una función g.

Suponiendo que representamos la imagen como un función f, donde f(x, y) es el nivel de gris para el píxel de la fila "x" columna "y", la función g a aplicar sería la siguiente:

- Si $f(x, y) \leq U1$ entonces $f(x, y) = 0$ (se reemplaza por un píxel negro)
- Si $f(x, y) \geq U2$ entonces $f(x, y) = 255$ (se reemplaza por un píxel blanco)
- Si $f(x, y) > U1$ y $f(x, y) < U2$ entonces $f(x, y) = 128$ (se reemplaza por el nivel intermedio de la escala de grises)

De esta forma las zonas brillantes se convertirán en manchas blancas, las zonas oscuras en manchas negras y las zonas de brillo intermedio ahora serán grises. Esto puede observarse en la Figura 4.



Figura 4. Imagen después de aplicar los umbrales

2.5 Búsqueda de manchas blancas y negras

En ésta etapa se buscan los centros de las manchas blancas y negras. Se trabaja sobre la imagen umbralada obtenida en el paso anterior.

Explicación del método desarrollado:

Se barre la imagen píxel a píxel, de izquierda a derecha y de arriba abajo. Sean p el píxel en cada paso del proceso, r el vecino superior de p , t el vecino izquierdo de p , q el vecino diagonal superior izquierdo de p y s el vecino diagonal superior derecho de p . La naturaleza de la secuencia barrido asegura que cuando se llega a p los píxeles r , t , q , y s ya han sido procesados.

Teniendo en cuenta los conceptos establecidos arriba se sigue el siguiente procedimiento: Si el píxel p no es del color de manchas buscado se continua hasta la próxima posición de barrido. Si el valor de p es del color de manchas buscado se examinan r , t , q y s . Si ninguno es del color de la manchas a buscar se asigna una nueva etiqueta a p (este píxel es la primera ocurrencia para la mancha a la que pertenece dicho píxel). Si solo uno de r , t , q , s es del color buscado se asigna la etiqueta de ese píxel a p . Si más de uno de estos píxeles es del color buscado y tienen la misma etiqueta se asigna esta a p ; pero si tienen etiquetas diferentes se asigna el valor de una de estas a p y se hace anotación de la equivalencia de etiquetas. Al final del barrido todos los píxeles del color de mancha buscado han sido etiquetados pero algunas de estas etiquetas pueden ser equivalentes. Es decir, una mancha puede estar compuesta por píxeles con etiquetas distintas, pero que son equivalentes [1].

Para solucionar este inconveniente se clasifican todos los pares de etiquetas equivalentes en clases de equivalencia, se asigna una etiqueta diferente a cada clase y luego se da una segunda pasada sobre la imagen reemplazando cada etiqueta por la etiqueta asignada a su clase de equivalencia. Se aprovecha esta segunda pasada para calcular las dimensiones de las manchas y sus respectivos centros.

El proceso de búsqueda de manchas se realiza dos veces, una vez para hallar las blancas y otra para hallar las negras.

2.6 Búsqueda de puntos (solo de referencia)

En esta etapa se buscan algunos puntos Braille, no todos los de la hoja. Estos son solo de referencia y serán necesarios en etapas posteriores del reconocimiento. Se trabaja con la imagen a tres colores, resultado de la etapa de umbralamiento; y las coordenadas de los centros de las manchas halladas en la etapa anterior.

Los puntos de una cara estarán formados por una mancha blanca arriba y una negra debajo (aproximadamente en la misma vertical), mientras que los puntos de la otra estarán formados por una mancha negra arriba y una blanca abajo (aproximadamente en la misma vertical).

Lo importante en esta etapa es no detectar puntos falsos. Los errores se deben a las manchas cercanas que llegan a pegarse al aplicar los umbrales (etapa de umbralamiento). Esto ocurre cuando hay puntos hacia arriba y abajo demasiado cerca.

Explicación del método desarrollado para hallar los puntos:

Se selecciona la primera mancha de la lista de manchas blancas y se verifica que sobre ésta no haya una mancha negra. Si es así se busca si hay una mancha negra debajo de ella. En caso afirmativo se habrá encontrado un punto Braille de la cara “A” de la hoja (se encontró una ocurrencia del patrón mancha blanca arriba, mancha negra abajo). Se calcula su centro con la información de coordenadas de las manchas en cuestión (la blanca y la negra) y se lo guarda en una lista de puntos para cara “A”. Luego la mancha negra (la que está por debajo de la mancha blanca) es removida de la lista de manchas negras.

La distancia máxima para verificar si una mancha está por arriba o por debajo de otra se ajustó experimentalmente y es de alrededor de 1.0 mm. La distancia es pasada a píxeles al momento de la comparación, ya que las coordenadas de los centros de las manchas están expresadas en esa unidad.

Para calcular la cantidad de píxeles que representan una cierta cantidad de milímetros se utiliza la siguiente fórmula f:

$$f(x) = (\text{DPI_RES} * x) / 25,4$$

Donde DPI_RES es la resolución (cantidad de píxeles por pulgada) de la imagen y 25,4 es la cantidad de milímetros que representa una pulgada.

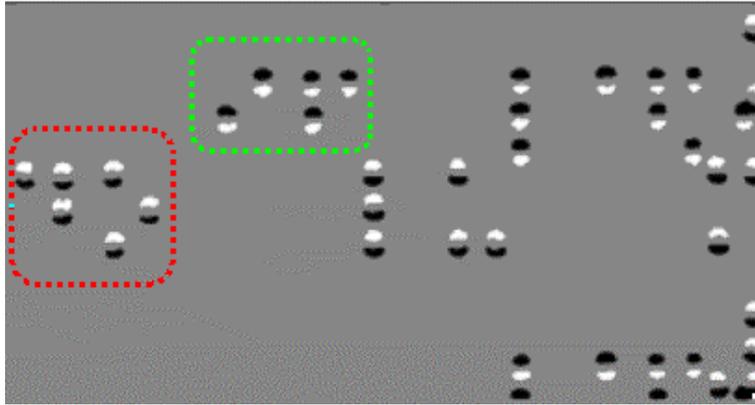


Figura 5. Las manchas encerradas por el cuadro punteado rojo corresponden a la cara “A” de la hoja. Las encerradas por el cuadro punteado verde corresponden a la cara “B” de la misma.

Este procesamiento se repite para cada mancha de la lista de manchas blancas.

Para hallar los puntos de la otra cara (cara “B”) se trabaja con la lista de manchas negras que sobraron. Para cada una se verifica que abajo haya una mancha blanca (ésta será seguro una mancha blanca para la que no se encontró pareja negra arriba y por eso fue descartada en la instancia anterior). En caso afirmativo se calcula el centro del punto hallado con la información de coordenadas de las manchas en cuestión (la negra y la blanca) y se lo guarda en una lista de puntos para cara “B”.

2.7 Búsqueda de los 10 puntos más cercanos al origen

En esta etapa se trabajará con la información de las listas de puntos de cada cara de forma independiente.

Se recorre la lista de puntos de la cara que se está procesando, buscando los diez puntos de coordenadas más cercanas al (0, 0). La coordenada (0, 0) corresponde al borde superior izquierdo de la imagen. Los puntos encontrados se guardan en una lista que será usada en etapas posteriores del reconocimiento.

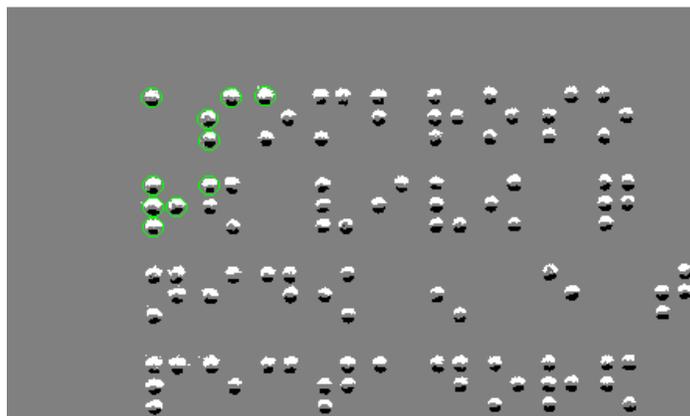


Figura 6. Los puntos Braille marcados con verde son los diez más cercanos al borde superior izquierdo.

2.8 Detección de la inclinación de la hoja

En esta etapa se detecta el ángulo de inclinación de la hoja sobre el escáner. Este valor será de gran utilidad en las etapas posteriores del reconocimiento.

Explicación del método desarrollado:

Se barre la imagen con diferentes ángulos. Para cada ángulo se suman los píxeles negros o blancos de cada fila. Luego se calcula la varianza de las sumas. El ángulo que maximice la varianza es el ángulo de inclinación de la hoja.

Este algoritmo es muy lento. Para aumentar la performance se tuvieron en cuenta algunas consideraciones:

- Se trabaja sobre una mitad de la imagen. Se elige la mitad que tenga mas información de puntos (mas probabilidad de texto Braille). Para realizar esta elección se pueden contar la cantidad de píxeles blancos y negros de cada mitad. Se selecciona la mitad que tenga mas píxeles blancos y negros.
- Se hace una reducción de la resolución horizontal de la imagen de 1 en 5. Es decir al realizar las sumas por fila se avanza de a 5 píxeles. Esto acelera considerablemente el algoritmo y no supone una perdida de precisión.
- Se utilizó un método de aproximación. En cada paso se busca el ángulo de inclinación con más precisión pero dentro del intervalo de error del paso anterior. La misma se hace en tres pasos.

Con estas consideraciones se aumenta notablemente el speed-up del algoritmo desarrollado a la vez que se mantiene la robustez y precisión del mismo [2].

2.9 Calculo de posición del primer punto de la hoja

Las distancias entre puntos en la escritura Braille siguen un patrón definido. Conociendo la resolución de escaneado esas distancias se pueden pasar a píxeles. Además, las filas y columnas estarán inclinadas con inclinación conocida (que podría ser nula). Se usará la suposición de que las filas y columnas son siempre perpendiculares aunque a veces esto no sea así.

Una vez conocido el ángulo de inclinación es posible centrarse en un punto y moverse las distancias adecuadas hasta encontrar todos los demás (por lo menos todos los verdaderos).

Para ello son necesarias dos cosas:

- Que el punto inicial sea verdadero.
- Saber a que posición (de 1 a 6) corresponde dentro de su carácter.

Se utiliza la lista de puntos más cercanos al origen (calculada en una etapa anterior) y se considera cada uno como posible punto inicial. Para cada punto se suponen las 6 posiciones posibles. Después se recorre la malla (llamamos malla al conjunto de posibles puntos de la hoja) contando los puntos de entrada que son “encajados”. La máxima de esas 60 cuentas determina el punto de referencia inicial [5].

Para hacer más robusto este algoritmo frente al ruido se tienen en cuenta dos cuestiones:

- Se permite cierto margen de error en la búsqueda de puntos (encajado).
- No se realiza la búsqueda en toda la hoja sino en un número limitado de filas y columnas a partir del punto de estudio. Esto permite que el algoritmo sea más rápido y además compensa pequeños errores en el ángulo de inclinación calculado.

2.10 Calculo de todos los posibles puntos de la hoja

Como se dijo anteriormente, llamamos malla Braille al conjunto de posibles puntos de la hoja. La construcción de ésta se realiza partiendo del punto de referencia inicial (calculado en la etapa anterior), colocando sobre el una plantilla de tamaño un carácter. Esta plantilla se moverá en todas direcciones hasta los límites de la hoja determinando, de esta forma, todos los posibles puntos. El movimiento de la plantilla se hace teniendo en cuenta las distancias estándar entre caracteres y el ángulo de inclinación detectado.

La creación de la malla es adaptativa: Al posicionar la plantilla se busca en la lista de puntos Braille (calculada en una etapa anterior) para ver cuantos puntos son encajados en esa posición. Si no se encaja ninguno se sigue con el próximo carácter, pero si se encajan puntos (para esa posición) la plantilla se centra basándose en ellos.

El próximo movimiento de la plantilla será relativo a la posición actual de la plantilla adaptada. De ésta forma el resultado no depende en exceso del punto de referencia inicial. Además se corrigen pequeños errores en el calculo del ángulo de inclinación de la hoja.

Los puntos Braille que se hayan detectado erróneamente en la etapa de búsqueda de puntos simplemente estarán descolocados y serán descartados al momento de construir la malla.

2.11 Reconocimiento de los puntos Braille

En la etapa anterior se calculó la malla Braille. Es decir la posición de los posibles puntos Braille. En esta etapa se determinará cuales de estos puntos están activos y cuales no. Se trabaja con la malla y la imagen umbralada a un porcentaje mayor para aumentar el tamaño de las manchas blancas y negras para disminuir el error en el encajado de puntos [5]. La malla está organizada en caracteres Braille (conjuntos de seis puntos Braille).

Se recorre la malla: Para cada carácter en la malla se determina que puntos del mismo están activos. Para ello se inspecciona la imagen por arriba para ver si hay una cierta cantidad de píxeles blanco y por debajo para ver si hay una cierta cantidad de píxeles negros. Si es así se anota el punto como activo, caso contrario será inactivo.

2.12 Transformación a texto digital

A fin de realizar la transformación a texto digital (ascii) es necesaria una estructura para almacenar el alfabeto Braille. Esta estructura es una tabla con 64 entradas donde el índice de la primera posición es 0 y el índice de la última es 63.

En esta etapa sabemos para cada carácter Braille de la malla cuales son los puntos que están activos y cuales no (se realizó en la etapa anterior). Tomando ventaja de ello se realiza el siguiente cálculo para cada uno de los caracteres:

$$c(p) = 32 * v(p_6) + 16 * v(p_5) + 8 * v(p_4) + 4 * v(p_3) + 2 * v(p_2) + v(p_1)$$

donde $v(p_x)$ es 1 si el punto p_x del carácter se determinó como activo y 0 en caso contrario.

El valor $c(p)$ calculado va a estar en el rango de 0 a 63 y se utiliza como índice de acceso para la tabla que contiene el alfabeto. Dependiendo del valor $c(p)$ se realiza un tratamiento distinto al momento de la traducción:

- Si es distinto de 40 y 60 se busca con este índice en la tabla y se obtiene el carácter de texto correspondiente.
- Si es igual a 40 significa que el o los próximos caracteres están en mayúscula [3] [4]. Entonces se inspecciona el próximo carácter de la malla. Si el valor de $c(p)$ para este carácter es distinto de 40 se busca en la tabla con este índice obteniendo el carácter de texto correspondiente y se lo pasa a mayúscula; mientras que si es igual a 40 (esto indica que toda la palabra es mayúscula) se inspeccionan los próximos caracteres hasta que el valor de $c(p)$ sea igual a cero 0 (es un carácter de espaciado) obteniéndose los caracteres para los índices $c(p)$ calculados y se los pasa todos a mayúsculas.
- Si es igual a 60 significa que los próximos caracteres son números [3] [4]. En este caso también se utiliza el valor de $c(p)$ como índice, pero de otra tabla del alfabeto. Esto se resolvió así por que la escritura Braille utiliza los mismos símbolos con los que representa las primeras letras del abecedario para representar los dígitos numéricos. De otra forma se tendrían dos posibles valores para el mismo índice sobre la misma tabla.

3. Conclusiones y Resultados Obtenidos:

Utilizando el esquema de solución propuesto en [5] se desarrollaron métodos ad-hoc para las diferentes etapas del reconocimiento automático de texto Braille obteniendo resultados óptimos. En particular se desarrolló un método para detectar la inclinación de la hoja que resultó más eficiente que el propuesto en [5]. Actualmente se está implementando la solución en Java.

4. Líneas Futuras:

Desarrollo de un algoritmo de segmentación para aislar los puntos de las diferentes caras en el caso de trabajar con braille interpunto (hojas doble faz).

Detección automática de distancias. Aunque las distancias entre puntos Braille están estandarizadas, existen varios tipos de hojas Braille con juegos de distancias distintos. Hasta ahora se le ingresa a los algoritmos las distancias entre caracteres y entre puntos de caracteres.

5. Bibliografía y Referencias:

- [1] Digital Image Processing, 2ND ED. Gonzalez, R.C.; Woods, R.E. McGraw-Hill 2002.
- [2] Optical Recognition of Braille writing using standard equipment – IEEE
- [3] <http://www.funcaragol.org/>
- [4] http://members.tripod.com/DE_VISU/brlest.html
- [5] http://www.gpi.tsc.uvigo.es/pub/papers/urs95_2.pdf
- [6] <http://www.sighted.com/english/obr.html>
- [7] Introducción al reconocimiento de patrones clásico. Oscar Bría.
- [8] Twain www.twain.org
- [9] IEEE Transactions on Pattern Analysis and Machine Intelligence
- [10] IEEE - Vision, Image and Signal Processing
- [11] W. Pratt, Digital Image Processing, 2nd Edition. John Wiley and Sons In., U.S.A., 1991.