

Associative models applied to a Braille-Spanish/Spanish-Braille translator

María Elena Acevedo, Jafeth Alonso, Fabiola Martínez, Karen Ailed Neri

Abstract-- In this work, we present an algorithm that allows the translation Braille-Spanish/ Spanish -Braille. The algorithm applies two tools for pattern recognition and translation of Braille code, such as Mathematical Morphology and Alpha-Beta Bidirectional Associative Memories, respectively. In the case of translation Spanish-Braille, we only used the Alpha-Beta BAM due to this task does not require image processing. The average of the effectivity in the recognition of Braille characters is 96.64% while in translation Spanish-Braille the algorithm showed the 100% of effectivity.

Index Terms--Artificial Intelligence, Associative Memories, Pattern Recognition, Image Processing, Translation.

I. INTRODUCTION

EL código Braille es un sistema de escritura que utiliza patrones de puntos realzados para inscribir caracteres en un papel. Por tanto, esto permite a los débiles visuales leer usando el tacto en vez de la vista [1]. Esta es una forma de que las personas invidentes participen en la cultura de la literatura. Desarrollado en el siglo diecinueve, el código Braille se ha convertido en el alfabeto táctil preminente. Sus caracteres son celdas de seis puntos arreglados en una matriz de 3 x 2. Aunque las celdas del código Braille son utilizadas alrededor del mundo, el significado de cada uno de los 64 posibles caracteres depende del lenguaje que se esté utilizando. Los diferentes lenguajes tienen su propio código Braille que mapean alfabetos, números y símbolos de puntuación a celdas Braille de acuerdo a sus necesidades.

Se estima que, alrededor de mundo; existen 285 millones de personas con discapacidad visual: 39 millones son invidentes y 246 millones tienen baja visión [2]. De acuerdo al censo del 2010 del Instituto Nacional de Estadística y Geografía [3], en México existen más de 5 millones 700 mil personas que presentan cierta incapacidad y la visual ocupa el segundo lugar. Lo anterior nos indica que la debilidad visual es un problema que afecta a mucha gente, por lo que en este trabajo

proponemos un algoritmo que permite la traducción Braille-Español/Español-Braille. Este algoritmo hace uso de dos herramientas de la Inteligencia Artificial tales como: la Morfología Matemática y las Memorias Asociativas para el reconocimiento de patrones.

A continuación se mencionan algunos trabajos relacionados con nuestra propuesta.

Dubus et al [4] propusieron un algoritmo para traducir el código Braille en relieve a tinta negra. Para esto, utilizan técnicas de filtrado y operaciones morfológicas para detectar los relieves. También se propuso un reconocimiento del código Braille [5] utilizando técnicas de convolución. El sistema no funcionaba correctamente si existía una variación en la alineación de las filas o de las columnas. Ng y Lau [6] utilizaron algoritmos de detección de bordes para traducir el código Braille a los lenguajes Español y Chino. La razón de reconocimiento fue aceptable, sin embargo, no se menciona si el sistema fue capaz de reconocer celdas deformes ni tampoco se menciona la efectividad. King [1] propuso un sistema para traducir texto a y desde el código Braille, está basado en un sistema de traducción que combina una máquina de estados finitos con un empate de contexto de izquierda a derecha y un conjunto de reglas de traducción. Se reporta la eficiencia en el tiempo de traducción. También se propuso un dispositivo portable [7] para explorar y traducir el código Braille en texto, el usuario controla la orientación de la exploración y, debido a que es portable, sólo se puede escanear un segmento del documento. Es muy eficiente en la traducción en tiempo real. Blenkhorn y Evans [8] propusieron un método automático para generar documentos Braille desde documentos de Microsoft Word. El documento se maneja como un objeto y mediante el uso de Visual Basic, se toma ventaja de los métodos y los atributos. En [9] se trabajó en un sistema híbrido que utiliza una red neuronal para el reconocimiento de caracteres Braille. El sistema puede manejar una gran cantidad de degradación en la imagen con un proceso de corrección que inserta y borra puntos. Otros algoritmos también utilizan una red neuronal para implementar un traductor Japonés-Braille [10], en este caso, se utiliza una red neuronal para insertar espacios entre las palabras para que la traducción resulte más simple. Para el reconocimiento óptico de del Braille en relieve, se utilizó una red neuronal probabilística [11]. Otra red neuronal se utilizó para traducir caracteres en Inglés a código Braille [12]. Los patrones de entrada de la red neuronal son los códigos ASCII de los caracteres, los patrones de salida son las posiciones de una tabla en donde el código Braille está almacenado. Al-

This work was supported by Instituto Politécnico Nacional (SIP, EDI, COFAA) and SNI.

M. E. Acevedo is with Instituto Politécnico Nacional, ESIME Zacatenco, Mexico City, Mexico (e-mail: acevedo@ipn.mx).

J. Alonso is with the Instituto Politécnico Nacional, ESIME Zacatenco, Mexico City, Mexico (e-mail: jaalonso@ipn.mx).

F. Martínez is with the Instituto Politécnico Nacional, ESIME Zacatenco, Mexico City, Mexico (e-mail: fmartinez@ipn.mx).

K. A. Neri is with the Instituto Politécnico Nacional, ESIME Zacatenco, Mexico City, Mexico (e-mail: karenanere8@hotmail.com).

978-1-5090-2435-3/16/\$31.00 ©2016 IEEE

Salman [13] utilizó un archivo con reglas para traducir texto a Braille y Braille a texto. Los lenguajes a traducir son Inglés y Árabe. En este trabajo, también se traduce un documento Microsoft Word a Braille. Fahiem [14] aplicó una máquina de Turing para el reconocimiento óptico del código Braille para traducirlo a texto Urdu. Se desarrolló una aplicación móvil [15] para traducir mensajes SMS a código Braille. Los resultados que reportan son: 100% de efectividad para texto simple, 97% para texto simple y símbolos especiales y 75% para texto e imágenes. Chakraborty y Mallic [16] propusieron el uso de la máquina de reconocimiento Tessarac de Google para extraer texto de imágenes y después traducir este texto a código Braille. Se dieron cuenta que el reconocimiento era mejor cuando la imagen procesada estaba en escala de grises. Se propuso un guante vibrador [17] para personas invidentes que han perdido la sensación en sus dedos como las personas diabéticas. Primero, el texto en Inglés es convertido a código Braille el cual está almacenado en una base de datos MS Access 2002. El guante tiene seis puntos, uno en cada dedo y otro en el centro de la palma de la mano. Los caracteres se toman de la base de datos y activan los puntos correspondientes en el guante. Padmavathi et al. [18] convierten un documento en Braille a tres alfabetos que son Inglés, Tamil y Hindi. Primero, procesan la imagen escaneada, el rango de la intensidad de los puntos se identifica mediante un histograma y después se realza el texto en Braille para incrementar el contraste de los puntos respect al fondo. Otro trabajo [19], también preprocesa la imagen escaneada para convertirla de Braille a Árabe, se aplican dos filtros morfológicos: bothat y tophat para realzar los puntos. Oda et al. [20] construyeron un modelo de aprendizaje estadístico basado en máquinas de soporte vectorial combinado con una máquina de traducción llamada KUIC (Kobe University Intelligent eBraille Engine for ChaSen), este sistema permite la traducción de información médica en lenguaje Japonés a Braille. Actualmente, algunos traductores en línea permite la traducción del Español a Braille: Duxbury [21], Online Braille Alphabet [22], Braille Translator [23], Libbraille [24]. Podemos observar que los algoritmos de Redes Neuronales han sido aplicados para la traducción de texto a Braille y Braille a texto. También el traductor de Microsoft Word ha sido utilizado. Además, la convolución y las operaciones morfológicas se han utilizado con el mismo propósito junto con otros algoritmos. Sin embargo, los modelos de memorias asociativas no se han aplicado para implementar un traductor de texto a y desde Braille.

En este trabajo se presenta un algoritmo basado en el enfoque de los modelos asociativos en conjunto con operaciones morfológicas (para el procesamiento de los caracteres Braille) para conformar un traductor Español-Braille/Braille-Español que tiene una eficiencia del 96.64% para el reconocimiento de caracteres Braille, mientras que para el reconocimiento de caracteres en español, presenta una efectividad del 100%. Nuestra principal aportación con este trabajo, es un modelo asociativo que muestra una eficiencia adecuada para el reconocimiento de patrones.

II. MÉTODOS Y MATERIALES

En esta sección, se describirán el pre-procesamiento del texto Braille y el modelo de memoria asociativa para el reconocimiento de patrones

A. Morfología Matemática

La Morfología Matemática [25] es una herramienta para extraer componentes útiles de una imagen para representar y describir áreas importantes (contornos, estructuras, etc.). La morfología matemática puede usarse para: el pre-procesamiento de imágenes (supresión de ruido, simplificación de formas), resaltar la estructura de objetos o describir objetos cualitativamente (área, perímetro, diámetro, etc.).

El elemento estructurante puede ser visto como un simple parámetro de forma para los filtros morfológicos.

La operación morfológica más básica es la erosión. Suponga que A es un conjunto (una imagen binaria o parte de ella) y S es un elemento estructurante. Si S es colocado con su pixel de referencia en (i, j) lo denotamos como $S(i, j)$. Entonces la erosión de A con S se define como el conjunto de todas las locaciones de los pixeles para los que S es posicionado en ese pixel que está contenido en A . Esto se denota como $A \ominus S$ y se escribe $A \ominus S = \{(i, j) : S(i, j) \subset A\}$.

Una operación complementaria a la erosión es la dilatación. Se define simplemente como la erosión de complemento de un conjunto. Si A^c denota el complemento de A , entonces la dilatación de un conjunto A por el conjunto S , denotado por $A \oplus S$, está definido por $A \oplus S = (A^c \ominus S)^c$.

B. Memoria Asociativa Alfa-Beta

Una Memoria Asociativa es un sistema que asocia estímulos con sus respectivas respuestas. Los estímulos son los patrones de entrada del sistema mientras que las respuestas son los patrones de salida. Formalmente, cada patrón de entrada se representa por un vector x y forma un asociación con el patrón de salida representado por un vector y .

Si μ es un índice, el conjunto fundamental se representa con: $\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ con p siendo la cardinalidad del conjunto. Los patrones que forman el conjunto fundamental se llaman patrones fundamentales. Si se cumple que $x^\mu = y^\mu, \forall \mu \in \{1, 2, \dots, p\}$, entonces la memoria asociativa \mathbf{M} es autoasociativa, de otra forma es heteroasociativa; en este caso es posible establecer que $\exists \mu \in \{1, 2, \dots, p\}$ por lo cual $x^\mu \neq y^\mu$.

Las memorias asociativas Alfa-Beta [26] son de dos tipos y son capaces de operar en dos modos diferentes. El operador α se utiliza en la fase de aprendizaje y el operador β es la base para la fase de recuperación. Estos operadores se definen como sigue: primero se definen los conjuntos $A = \{0, 1\}$ y $B = \{0, 1, 2\}$, y los operadores α y β se definen en la Tabla I.

Los conjuntos A y B , los operadores α y β , junto con los operadores usuales \wedge (mínimo) y \vee (máximo), forman el Sistema algebraico $(A, B, \alpha, \beta, \wedge, \vee)$ que es la base matemática para las memorias Alfa-Beta.

TABLA I

Operadores Alfa $\alpha: A \times A \rightarrow B$ y Operador Beta $\beta: B \times A \rightarrow A$

x	y	$\alpha(x, y)$	$\beta(x, y)$
0	0	1	0
0	1	0	0
1	0	2	0
1	1	1	1
2	0	---	1
2	1	---	0

La Memoria Asociativa Bidireccional (BAM, Bidirectional Associative Memory) general es una caja negra que opera de la siguiente manera: dado un patrón x , se obtiene su patrón asociado y , y dado el patrón y , se recupera su correspondiente patrón x .

En ambas direcciones, el modelo de la BAM Alfa-Beta [27] está conformado por cuatro etapas.

Por simplicidad, primero se describirá el proceso necesario para una dirección.

La función de la etapa 2 es obtener una y^k como salida ($k = 1, \dots, p$) dada una x^k como entrada. Se asume que como entrada a la etapa 2 tenemos un elemento de un conjunto de p vectores ortonormales. Recordemos que el Linear Associator tiene recuperación correcta cuando trabaja con vectores ortonormales. En este caso, se utiliza una variación del Linear Associator para obtener y^k , partiendo de un vector *one-hot* h^k en su k -ésima coordenada.

Para la construcción del Linear Associator modificado, nos saltamos la fase de aprendizaje y se construye, directamente, la matriz M que representa a la memoria. Cada columna en esta matriz corresponde a cada patrón de salida y^k . De esta forma, cuando la matriz M se opera con el vector *one-hot* h^k , siempre se recupera el correspondiente y^k .

La tarea de la Etapa 1 es: dado un x^k o la versión ruidosa de éste, el vector *one-hot* h^k debe ser obtenido sin ambigüedad y sin ninguna condición. Los algoritmos de las fases de aprendizaje y recuperación se encuentran ampliamente descritos en [27].

C. Traductor Español-Braille/Braille-Español

En esta sección, se describe el procesamiento de la imagen para la detección de los puntos Braille. Después, se presenta la construcción de la memoria asociativa.

La hoja del escrito en Braille fue convertida a formato digital a través del escáner de un multifuncional EPSON Stylus TX130. Tras realizar varias pruebas variando las características de la configuración del scanner, tales como su resolución, su color, y los formatos de imagen permitidos, se decidió establecer una resolución de 100ppp, un formato de imagen tipo JPG y que el escaneo se realizará en escala de grises. Estas características permiten tener una imagen aceptable para poder ser procesada en los siguientes bloques.

Una vez que se ha digitalizado la imagen, es necesario convertirla a un mapa de bits, éste permite la manipulación y edición de los valores de las tres capas de color de cada pixel. Una vez hecho este proceso, la imagen como mapa de bits se binarizó, es decir, se le aplicó una conversión a blanco y negro, para ello se calculó el promedio de los valores de las

capas de color, pixel por pixel. Si el valor del promedio es más cercano a cero se asigna el color negro a ese pixel, en caso contrario se asigna el color blanco. Para determinar la “cercanía” se utilizó un valor a que representa el porcentaje, o bien, el punto de referencia para asignar valores blanco o negro. Por ejemplo, si una imagen es binarizada con un factor del 50% ($a=0.5$) entonces se asignará un valor 0 (color Negro) a todo pixel cuyo promedio de sus valores rojo, verde y azul den como resultado un valor dentro del rango 0 a 127 se establecerá un valor de cero a ese pixel, en caso contrario, cuando todos los valores superen este rango, el valor asignado es un 255, representando al color blanco.

Entre mayor sea el porcentaje a , la binarización será más selectiva, eliminando información innecesaria en la imagen y resaltando únicamente los relieves o puntos de la hoja en Braille.

Posteriormente, se hizo un análisis de densidad de pixeles blancos en cada renglón de la imagen, con el objetivo de descartar puntos en blanco que no representen nada, rescatando los puntos o relieves de la hoja. El algoritmo cuenta en cada renglón el número de pixeles en blanco, si esta cantidad es muy pequeña en comparación al ancho en pixeles de un punto Braille, el renglón completo es iluminado. En caso contrario el renglón permanece intacto.

Como puede observarse, desde el subproceso anterior los renglones correspondientes a un ancho de punto son identificados, por lo que para la detección y selección de un renglón correspondiente a un ancho de casilla Braille es necesario conjuntar tres renglones simultáneos.

Esto arroja como resultado una imagen con una dimensión horizontal igual a la de la hoja original y una dimensión vertical correspondiente a la altura de una casilla en Braille (ver Figura 1).



Fig. 1. Sección de una hoja Braille que representa un renglón de escritura de este sistema.

Esta sección de la imagen será nuevamente analizada para detectar cada casilla, es decir, cada carácter. Las casillas en el renglón (Figura 1) no están determinadas explícitamente en la imagen, por lo que se hace una aproximación de sus dimensiones, partiendo del tamaño de un punto en Braille. Para conocer el ancho y , simultáneamente, el número de columnas en la línea de escritura seleccionada se hace una dilatación vertical de este segmento, en otras palabras se ilumina en blanco cada columna en la cual exista por lo menos un pixel en blanco. El resultado de este proceso se muestra en la Figura 2.



Fig. 2. Dilatación del renglón de escritura Braille

Debido a que un caracter Braille puede o no utilizar las dos columnas de la casilla, no se puede dar por hecho que dos columnas, detectadas en este segmento de imagen, formen una casilla ya que, como puede observarse, las distancias entre una columna es aproximadamente igual en algunos pares, sin embargo existen columnas con una distancia mayor con respecto a su columna sucesiva o previa. Por tanto es necesaria una inspección de cada columna y sus alrededores para determinar si junto con las columnas de su derecha y/o izquierda conforman una celda, además de conocer qué posición ocupa en ella. Debe mencionarse que se hizo una correspondencia de las dimensiones de la casilla Braille en función del tamaño de sus columnas, partiendo de las dimensiones estándar de los caracteres del sistema de escritura para invidentes: Ancho del punto y la columna es 1.5 mm, Distancia entre columnas 3.75 y Ancho de la casilla es 5.5 mm.

Para establecer el ancho de un relieve o punto en la imagen dilatada, se hizo un barrido horizontal en una de sus filas, y se obtuvo el valor mayor del ancho de las columnas en blanco. El funcionamiento del algoritmo se explica a continuación:

1. Se parte de la ubicación de la columna más cercana al lado izquierdo de la imagen. A partir de su inicio se toma una distancia equivalente a tres veces el ancho de una columna (valor mayor previamente identificado), con el fin de abarcar una longitud aproximadamente igual al ancho de una casilla.
2. Se hace un análisis a lo largo de esta nueva subsección para detectar más columnas a la derecha de la ya seleccionada.
3. En caso de que sean dos las columnas en esta región, incluyendo la seleccionada, entonces se considera como una casilla completa, es decir, una casilla que utiliza las dos columnas.
4. En otro caso, cuando sólo exista una columna en la región examinada se hace una comparación respecto a la posición de la columna posterior, si esta distancia rebasa tres veces el ancho de un punto pero es menor que cuatro veces este valor, entonces se considera que la columna se ubica en el lado izquierdo de la casilla Braille, en caso contrario, si esta distancia es aún mayor se considera a la columna en el lado derecho. Para cualquiera de estos casos, se reubica al renglón en la región de la casilla.
5. El proceso se repite hasta que todas las columnas de la figura sean reubicadas, partiendo de la columna inmediata a la última acomodada.

Ya que se ha identificado la ubicación de cada casilla en la línea de escritura, se selecciona una a una para determinar qué caracter representa. Para ello, la casilla es seccionada como una matriz de 3x2, con el objetivo de identificar la posición de los puntos utilizados. Por otra parte, se declara un vector de seis elementos con valores iniciales de 0, vector Braille = [0, 0, 0, 0, 0, 0].

Se hace una inspección en cada subregión de la casilla para identificar si existe un punto o no, en caso afirmativo se asigna un 1 al elemento del vector correspondiente, los demás

continuarán intactos, vector Braille = [0, 1, 0, 1, 0, 0]. Este vector será la entrada para la memoria alfa-beta creada previamente. Cabe señalar que el sistema Braille posee prefijos que indican si los caracteres inmediatos son números mayúsculas, etc., por lo que antes de ingresarlo a la memoria es importante hacer una inspección previa. Al identificar este vector, se resta un 32 binario al carácter viario devuelto por la memoria asociativa, con el fin de escribir el carácter siguiente como una letra mayúscula del código ASCII.

Construcción de la BAM Alfa-Beta

Primero se definen los patrones de entrada y salida: x y y , respectivamente, con los que se va a construir la Memoria Asociativa Bidireccional Alfa-Beta. Los vectores de entrada tienen una dimensión de $n = 6$ y los vectores de salida tienen una dimensión de $m = 7$. La BAM Alfa-Beta almacenará 26 pares de patrones, $p = 26$, los cuales corresponden a la cantidad de letras en el alfabeto latino. Los patrones de entrada representarán los vectores Braille, mientras que los patrones de salida serán cada una de las letras del alfabeto latino, las cuales tendrán que ser convertidas a binario para que la BAM Alfa-Beta pueda manejarlas.

Con estos patrones, ya definidos, se utiliza el algoritmo de aprendizaje para construir la BAM Alfa Beta.

III. EXPERIMENTOS Y RESULTADOS

El algoritmo fue implementado en Visual C# 2012.

El software presenta dos opciones de traducción, por default, aparece primero la traducción Braille-Español. Se ejecutaron varias pruebas con diferentes documentos Braille escritos a mano. El texto original se comparó con la traducción obtenida del software. Los resultados se muestran en la Tabla II.

TABLA II.
Efectividad en la traducción Braille-Español

# Caracteres	#Errores	Efectividad (%)
249	4	98
217	7	96
568	50	91
273	4	98
178	7	96
257	5	98
239	3	98
101	2	98
114	4	96
378	20	94
181	3	98
109	2	98
185	3	98
145	5	96

El promedio de efectividad de nuestra propuesta fue del 96.4%, lo cual es aceptable para la comprensión del documento traducido. Cabe señalar que estos valores se

obtuvieron con el umbral de binarización que permitió una mejor recuperación del texto. Este umbral debe ser modificado en función del ancho de los puntos.

En el caso de la traducción Español-Braille, la efectividad es del 100% de todo el documento, esto sucede porque no se realiza un procesamiento de imágenes para el reconocimiento de cada caracter.

Se debe puntualizar que los experimentos se realizaron en condiciones óptimas de alineación del documento digitalizado. El margen de desviación de la alineación del documento es de $\pm 2^\circ$. Cuando en la elaboración del documento se tienen deficiencias que puedan afectar la alineación, el software no puede realizar la correcta traducción.

IV. REFERENCES

- [1] A. King, "Text and Braille Computer Translation", A dissertation submitted to the University of Manchester Institute of Science and Technology for the degree of Master of Science, 2001.
- [2] World Health Organization, 2014, <http://www.who.int/mediacentre/factsheets/fs282/en/>
- [3] Instituto Nacional de Estadística y Geografía (Mexico) <http://www.inegi.org.mx/>
- [4] J.P. Dubus, M. Benjelloun, V. Devlaminck, F. Waquier and P. Altmayer, "Image processing techniques to perform an autonomous system to translate relief Braille into black-ink, called: Lectobraille", *IEEE Engineering in Medicine and Biology Society 10th annual international Conference*, vol. 4, pp. 1584 – 1585, Nov. 1998.
- [5] J. Mennens, L.V. Tichelen, G. Francois and J.J. Engelen. "Optical Recognition of Braille Writing Using Standard Equipment", *IEEE Transactions on Rehabilitation Engineering*, vol. 2(4), pp. 207-212 Dec. 1994.
- [6] C.M. Ng, V. Ng and Y. Lau, "Regular feature extraction for recognition of Braille", *Third International Conference on Computational Intelligence and Multimedia Applications ICCIMA '99. Proceedings*, pp. 302-306, 1999.
- [7] I. Murray and T. Dias, "A portable device for optically recognizing Braille, Part I: Hardware development". *Seventh Australian and New Zealand Intelligent Information Systems Conference*, pp. 129-134, Nov. 2001.
- [8] P. Blenkhorn and G. Evans, "Automated Braille Production from Word-Processed Documents", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9(1), pp. 81-85, March 2001.
- [9] G. Morgavi and M. Morando, "A neural network hybrid model for an optical braille recognizer", *2nd WSEAS International Conference on Signal, Speech and Image Processing*, pp. 3201- 3206, 2002.
- [10] T. Watanabe, K. Kisa, K. Nishumura and S. Kishida, "Applications of Neural Networks into Japanese-to-Braille Translation", *The 47th IEEE International Midwest Symposium on Circuits and Systems*, pp. III-109 – III-112, July 2004.
- [11] L. Wong, W. Abdulla and S. Hussmann, "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2 pp. 586-589, Aug. 2004.
- [12] M. Y. Hassan and A. G. Mohammed, "Conversion of English Characters into Braille using Neural Network", *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 11(2), pp. 30-37, 2011.
- [13] A. S. Al-Salman, "A Bi-direccional Bi-lingual Translation Braille-Text System", *Comp. & Info. Sci.*, vol. 20, pp. 13-29, 2008.
- [14] M. A. Fahiem, "A Deterministic Turing Machine for Context Sensitive Translation of Braille Codes to Urdu Text", *12th International Workshop on Combinatorial Image Analysis, LNCS 4958*, pp. 342-351, April 2008.
- [15] D. Kumar, F. Khan and S. Islam, "Mobile SMS to Braille Transcription: A New Era to Mobile for the Blinds", *International Conference and Workshop on Emerging Trends on Technology*, pp. 70-74, Feb. 2010.
- [16] P. Chakraborty and A. Mallic, "An Open Source Tesseract based Tool for Extracting Text from Images for Application in Braille Translation for the Visually Impaired", *International Journal of Computer Applications*, vol. 68(16), pp. 26-32, April 2013.
- [17] B. L. Shivakumar and M. Rajasenathipathi, "Braille Glove Vibration System for Diabetic Affected Disabled Persons", *Journal of Engineering and Applied Sciences*, vol. 8(8), pp. 635-641, Aug, 2013.
- [18] S. Padmavathi, K. S. S. Manojna, S. S. Reddy and D. Meenakshy, "Conversion of Braille to text in English, Hindi and Tamil Languages", *International Journal of Computer Science, Engineering and Applications*, vol. 3(3), pp. 19-32, June 2013.
- [19] Z. I. Authman and Z. F. Jebr, "Arabic Braille scripts recognition and translation using image processing techniques", *Journal of College of Education*, vol. 2(3), pp. 18-26, 2012.
- [20] T. Oda, A. Sugano, M. Shimbo, et al., "Improvement in Accuracy of Word Segmentation of a Web-Based Japanese-to-Braille Translation Program for Medical Information", *Journal of Communication and Computer*, vol. 10, pp. 82-89, Jan. 2013.
- [21] http://www.tecnologia.com.ar/productos_software_traductor_braille_duxbury.html
- [22] <http://www.fbu.edu.uy/alfabeto/alfabeto-online.htm>
- [23] <http://www.brailletranslator.org/es.html>
- [24] <http://libbraille.org/translator.php?src>
- [25] C. A. Glasbey and G. W. Horgan, "Image Analysis of the biological sciences". National Remote Sensing Centre Ltd, Farnborough, Hampshire, 1994.
- [26] C. Yáñez and J. L. Díaz de León, "Associative Memories Based on Order Relations and Binary Operators". *Computación y Sistemas*, vol. 6(4), pp. 300-311, 2003.
- [27] M. E. Acevedo, "Alpha-Beta Bidirectional Associative Memories", PhD Thesis, Centro de Investigación en Computación, Mexico City, Mexico, 2006.