

Introducción al Análisis Gráfico Para Obtener Frecuencias de Símbolos Escritos en Imágenes de Texto

C. Zambrano¹, P. Chávez²

Facultad Ingeniería en Electricidad y Computación (FIEC)
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral. Apartado 09-01-5863. Guayaquil, Ecuador
Czambrano1@bankguay.com¹, pchavez@fiec.espol.edu.ec²

Resumen

El estudio de culturas antiguas por medio de sus expresiones escritas nos dejan un legado invaluable de nuestra historia, en nuestros tiempos estos estudios cuentan con herramientas de ayuda entre las cuales el procesamiento digital de señales provee una gama amplia de aplicaciones. Este Analizador gráfico es una introducción en el estudio de símbolos de escritura el cual está desarrollado en MATLAB y tiene como objetivo fundamental extraer símbolos diferentes con sus respectivas frecuencias y ubicaciones dentro de una imagen de texto. El Analizador proveerá herramientas básicas para el tratamiento previo como la eliminación de ruido, utilizará un método de segmentación por conexiones nulas, la comparación de matrices con la elección por parte del investigador del nivel de tolerancia y una herramienta de selección para la visualización de los resultados.

Palabras Claves: Escritura, MATLAB, Tratamiento, Segmentación, Comparación, Visualización de Resultados.

Abstract

The study of old cultures using its written expressions leaves us an invaluable historical legacy, in our times these studies have help tools among which the digital signal procesing provides a wide range of applications. This graphic Analyzer is an introduction in the study of written symbols which is developed in MATLAB and it has as fundamental goal to extract different symbols with their respective frecuencies and location inside a text image. The Analyzer will provide basic tools for the previous treatment as the elimination of noise, it will use a segmentation method for null connections, the comparison of wombs with the election on the part of the investigator of the level of tolerance and a selection tool for the visualization of the results.

Keywords: Written Expressions, MATLAB, Treatment, Segmentation, Comparison, Visualization of Results.

1. Fundamentos Teóricos

1.1 Binarización

La mayor parte de los algoritmos para reconocer escritura están escritos a partir de imágenes binarias, por lo que se hace conveniente el paso de una imagen en niveles de gris (o color) a una binaria, además esto permite reducir el volumen de los datos a tratar.

La binarización de una imagen digital consiste en convertir la imagen digital en blanco y negro, de tal manera que se preserven las propiedades esenciales.

Uno de los métodos para poder binarizar una imagen digital es mediante el histograma de dicha imagen. A través del histograma obtenemos una gráfica donde se muestran el número de píxeles por cada nivel de gris que aparecen en la imagen.

Para realizar la binarización, se deberá elegir un valor adecuado dentro de los niveles de grises (umbral), de tal forma que el histograma forme un valle en ese nivel. Todos los niveles de grises menores al umbral calculado se convertirán en negro y todos los mayores en blanco.

1.2 Fragmentación o Segmentación

Una vez obtenida la imagen binaria se deberá fragmentar o segmentar en las diferentes componentes conexas (parte de la imagen donde todos los píxeles son adyacentes entre sí) que la componen. La fragmentación o segmentación de la imagen constituye una de las mayores dificultades del reconocimiento, y se hace necesaria para poder reconocer cada uno de los caracteres de la imagen binaria. La fragmentación o segmentación es la operación que permite la descomposición de un texto en diferentes entidades lógicas. Estas entidades lógicas deben ser lo suficientemente invariables, para ser independientes del escritor, y lo suficientemente significativas para su reconocimiento.

Con la segmentación se debería localizar las zonas de interés y separar por dichas zonas. Las zonas de interés estarán caracterizadas por unos atributos comunes como son dimensión, superficie, densidad, inclinación, longitud del trazo, etc. No existe ningún método genérico para realizar la segmentación de la imagen que sea lo suficiente eficaz para el análisis de un texto. Por ejemplo, uno de los métodos sería separar la imagen en entidades con una conexión mínima con la vecina:

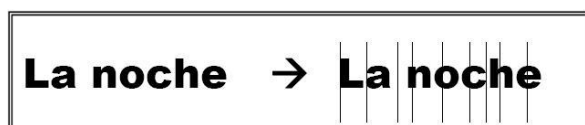


Imagen 1.1: Segmentación de una imagen verticalmente

Aquí se ve reflejado que en las letras “n” y “h” el método fallaría, porque las separaría por la mitad ya que existe una conexión mínima dentro de ellas. Otro ejemplo, sería utilizar un método que segmentase la imagen en entidades que no tuviesen ningún punto en común. En imágenes perfectas daría un resultado favorable. Sin embargo, en la realidad, teniendo en cuenta el ruido, esto resulta complicado.

1.3 Representación Vectorial de los Colores

Las imágenes pueden ser representadas en un mapa de bits que consiste en filas de píxeles donde cada elemento tiene un valor numérico que marca su color. Los colores se codifican en tres bytes, es decir 3 valores numéricos que van de 0 a 255 cada uno representando así su descomposición en los tres colores primarios: rojo, verde y azul. Matemáticamente puede interpretarse un color como un vector en el espacio tridimensional de Rojo, Verde y Azul.

Bajo esta interpretación pueden aplicarse algunos conceptos de la geometría analítica en el tratamiento de colores y en la generación de filtros o transformaciones.

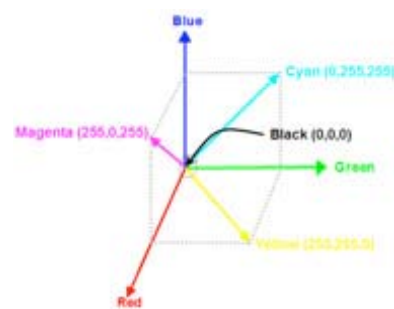


Imagen 1.2 Espacio tridimensional de colores

Una imagen es una codificación en un dominio espacial bidimensional estático y esto nos permite que podamos contar con nuevas imágenes a partir de las originales haciendo uso de transformaciones o filtros aplicados a sus píxeles

1.4 Escala de Grises

En el espacio de colores, los vectores en la dirección del vector (1, 1, 1) representan diferentes tonalidades de gris. Así, cualquier píxel (r, g, b) de una imagen, proyectado sobre este vector nos dará su contribución gris a una nueva imagen que formemos con todas las proyecciones de los píxeles originales.

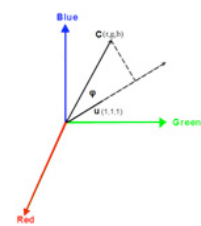


Imagen 1.3. El gris en el Espacio vectorial de Colores

Del álgebra de vectores sabemos que el producto escalar:

$$\mathbf{C} \cdot \mathbf{u} = (r, g, b) \cdot (1, 1, 1) = r + g + b = |\mathbf{C}| \times |\mathbf{u}| \times \cos(\varphi)$$

Que nos da al calcular las magnitudes de los vectores:

$$r + g + b = \sqrt{3(r^2 + g^2 + b^2)} \cos(\varphi)$$

Pero la proyección de C en la dirección de u es:

$$\sqrt{(r^2 + g^2 + b^2)} \cos(\varphi) = \text{Proy } \mathbf{C}_u$$

Entonces:

$$\text{Proy } \mathbf{C}_u = \frac{r + g + b}{\sqrt{3}}$$

El mayor valor que puede tomar esta expresión es $255 \sqrt{3}$ y como debemos cuidar que la magnitud de esta expresión nunca rebase 255 debemos normalizarla multiplicando por $1/\sqrt{3}$, así:

$$\text{Proy } \mathbf{C}_{u_n} = \frac{r + g + b}{3}$$

Es la proyección normalizada de un pixel en la dirección de los grises. De esta manera una imagen a colores puede ser transformada en escala de grises aplicando este concepto a cada pixel.

2. Desarrollo del Analizador

2.1 Alcance

Para el desarrollo del Analizador Gráfico de Símbolos de Escritura, el cual llamaremos a partir de ahora "Analizador", se establecieron algunas definiciones preliminares que marcan el alcance del mismo.

El Analizador conseguirá una introducción básica al estudio de símbolos de escritura, por lo que los resultados estarán limitados a obtener datos específicos tales como:

- Símbolos diferentes hallados en una imagen.
- Cantidad de repeticiones por cada símbolo.
- Ubicación de cada símbolo y sus repeticiones en la imagen.

El Analizador tendrá herramientas propias de preparación de la imagen con el fin de aplicar algunos conceptos del procesamiento digital, estas herramientas serán básicas, entre ellas tenemos:

Conversión de Imagen original en escala de grises y en binaria.

Aplicación de filtro eliminador de ruido "sal y pimienta".

Eliminador de pixeles agrupados y aislados de nuestros símbolos de interés.

El Analizador permitirá al usuario revisar cada símbolo hallado mediante una herramienta de selección incluida.

El Analizador está diseñado para descubrir símbolos en una imagen, por lo tanto en la segmentación no se utilizarán métodos por conexiones mínimas sino por conexiones nulas, por lo tanto es responsabilidad del analista conseguir buenas fotografías que minimicen el error. La razón de esto es a fin de no alterar los símbolos y permitir que sea el observador quien saque conclusiones de los resultados.

2.2 Implementación

Para llegar a los resultados deseados, se ha creado una interfaz que presentará las opciones de preparación de la imagen de una manera dinámica, de forma que estas herramientas estarán disponibles cuando sea posible su uso.

Se utilizó Matlab para el desarrollo y las pruebas han sido realizadas con imágenes descargadas en internet y con fotografías de escritos reales en hebreo. A continuación se explicará la lógica de desarrollo utilizada en el Analizador

2.2.1 Preparación de la Imagen

La imagen original cargada por el usuario, es guardada en una matriz y convertida en escala de grises y en binaria en variables diferentes. Para la binarización se utilizó el umbral calculado por Matlab a través del método de Otsu. En estos momentos el botón de "Escala Grises" aparece como la única opción de manera que al elegirla, se muestra el contenido de la variable que posee la imagen en grises, el botón cambia a la opción "Binarizar" de manera que si se presiona tendremos a la vista la imagen en blanco y negro. Se utiliza una bandera de estado para conocer cuál es la imagen que se está mostrando. Cuando la imagen mostrada está en escala de grises, se permite sólo la opción de "Filtrar Ruido" con el objetivo de verificar lo que ocurre al aplicar el filtro en el modo de escala de grises. Cuando la imagen mostrada es binaria,

están disponibles cuatro opciones que pueden ser aplicadas:

Filtrar Ruido.- Aplica un filtro mediano utilizando una máscara de 3x3 para eliminar ruido de sal y pimienta. Esta máscara recorre toda la imagen pixel a pixel verificando los valores del vecindario de cada uno manteniendo los bordes y eliminando además de pixeles aislados, líneas delgadas que resultan ruido y que atraviesan los símbolos. Debe tenerse cuidado al aplicar este filtro en las imágenes con texto en hebreo pues podría eliminar parte de la integridad de los símbolos afectando el análisis final.

Eliminar Pixeles.- Sólo elimina grupos de pixeles aislados. Por medio de una barra de deslizamiento se escoge la cantidad de pixeles agrupados que deseamos sean eliminados.

Análisis de Bordes.- Se incluyó para una observación de los símbolos por bordes. Es muy vistoso cuando tenemos imágenes que buena resolución. Se utiliza el algoritmo de Canny para detección de bordes y se crea una nueva variable para guardar la imagen final por si el usuario desea aplicar al Análisis final sobre los bordes y no sobre símbolos llenos.

Análisis de Plantillas.- Es el propósito final de la preparación de la imagen que lo veremos a continuación.



Imagen 2.1 Preparación de la Imagen

2.2.2 Análisis Digital Automático

Esta opción es la principal de nuestro Analizador, ya que es aquí donde de manera automática se realiza la segmentación de la imagen y la búsqueda de símbolos con sus frecuencias.

Luego de cada tratamiento la imagen queda almacenada lista para ser tomada por esta función, una vez cargada la imagen según el último tratamiento, se utiliza la técnica de segmentación por etiquetas, es decir a cada símbolo continuo se le asigna una etiqueta numérica para discriminar entre ellos.

La discriminación utilizará lo que he llamado “nivel de correlación”, la función pregunta al usuario el nivel que desea utilizar de la siguiente manera:



Imagen 2.2 Correlación

La correlación es la similitud que deben tener los símbolos entre sí al ser analizados. Ya que esta función del Analizador realizará un barrido y comparará símbolo por símbolo verificando la similitud, se permite al usuario escoger en tres niveles:

Alto.- Que equivale a una similitud del 75%. Este nivel es recomendable por si queremos utilizar el analizador en imágenes donde los símbolos escritos están muy bien dibujados de manera que la comparación entre unos y otros tiene un umbral del 0.25 de error.

Medio.- Equivalente al 50%. Este nivel es el recomendado para símbolos con menor nitidez, en las pruebas realizadas ha sido eficaz en este nivel debido a la calidad de las imágenes utilizadas.

Bajo.- La similitud es del 30%. Es decir un gran nivel de tolerancia. Esta correlación podría ser útil para símbolos muy diferentes entre sí y que muestren una mala caligrafía de manera que el Analizador pueda agrupar los aproximados al 30% en similitud.

Una vez escogida esta opción, la función procede a revisar el tamaño máximo de los símbolos etiquetados, para ello se realiza un barrido total revisando el largo y ancho en pixeles de cada símbolo acumulando el máximo valor en las respectivas variables asignadas las mismas que al final recibirán un incremento forzado del 10%. Estos valores máximos serán utilizados para crear las plantillas de cada símbolo etiquetado, el incremento del 10% es para dejar holgura en las plantillas correspondientes a los símbolos de mayor tamaño.



Imagen 2.3 Tamaño de Plantillas

En el siguiente paso, se vuelve a realizar el barrido total obteniendo cada símbolo etiquetado realizando un recorte del símbolo en la imagen general y rellenando con ceros hacia las cuatro direcciones para completar el tamaño calculado de la plantilla en el paso anterior. Cada una de estas plantillas se irán grabando en un arreglo de matrices de (Mry, Mcx, Num) dimensiones, donde Mry y Mcx son los valores máximos de largo y ancho calculado, en nuestra

imagen anterior corresponden a los valores Y3 y X8 y Num es la cantidad de símbolos etiquetados que marcará el número de plantillas que se crearán. De esta manera las plantillas quedan grabadas a manera de un libro donde una está detrás de la otra.

Luego de cada plantilla que es reconocida y agregada al arreglo, se presenta esta imagen en la parte inferior de la interface del Analizador y se hace una pequeña pausa para una breve observación.

Una vez culminado este proceso de crear las plantillas, se procede a la creación de la matriz de correlación. Esta matriz es un arreglo de 2 dimensiones de Num x Num, cuyo tamaño será la cantidad de símbolos etiquetados. Los índices de las filas y columnas corresponderán a cada una de las plantillas agregadas al arreglo en el paso anterior. Las coordenadas (i,j) contendrán el valor que indicará si la plantilla "i" tiene o no un nivel de correlación con la plantilla "j" mayor al escogido por el usuario previamente.

El proceso consiste en la comparación de las plantillas del arreglo de matrices, cada una es comparada con todas las demás y el valor de la comparación es ubicado en la coordenada (i,j) de la matriz. La comparación hecha en Matlab utiliza la siguiente fórmula matemática:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

Donde A y B representan las matrices a ser comparadas y los valores \bar{A} y \bar{B} son los promedios de cada una de las matrices. Se realiza un cálculo de similitud entre ambas matrices y r es el resultado que oscila entre 0 y 1. Es un escalar de tipo real que representa el porcentaje de similitud entre ambas matrices. Es aquí donde utilizamos el nivel de correlación elegido por el usuario, ya que al calcular para cada par de plantillas (i,j), si el valor resultante es mayor al nivel elegido, éste es reemplazado por "1", caso contrario se sobrescribe un "0".

i,j	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
01	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0
05	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
06	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
07	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0
08	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
09	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0
10	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Matriz Correlación

A continuación se realiza la verificación de los símbolos diferentes hallados para lo cual revisamos en cada fila de la matriz de correlación la cantidad de "1" que tenemos, esto nos indica cuántas veces se encontró el símbolo "i" en nuestro arreglo de plantillas.

Para el caso de la matriz que estamos presentando, la fila i = 1 tiene 3 valores verdaderos (1), vemos que estos valores corresponden a las posiciones de j = 1, 2 y 3, de la misma manera ocurre para i = 2 y 3. Debemos comprender que en fila i = 1 el dato que estamos observando es que las plantillas 1, 2 y 3 contienen el mismo símbolo, de esta manera si queremos continuar al siguiente símbolo diferente, debemos obviar las filas i = 2 y 3.

Al revisar la fila i = 4 encontramos verdaderos j = 4, 7, 9 y 10, es decir que la cuarta plantilla contiene al mismo símbolo que las plantillas 7, 9 y 10, entonces utilizando el mismo razonamiento anterior, para continuar revisando símbolos diferentes, ahora debemos obviar las plantillas 7, 9 y 10.

Siguiendo esta secuencia de análisis y anexando estos datos a un vector posición tendremos la ubicación de los símbolos diferentes hallados. El vector de posición que daría de la siguiente manera:

Vector Posición	1	0	0	2	3	0	0	0	0	4	0	5	0	6	0	0
Posición	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17

Luego de este procedimiento, en el vector de posición tenemos la información de cuáles son los símbolos diferentes y cuántos hay en la imagen original, en nuestro ejemplo tenemos que hay 6 símbolos diferentes y se encuentran en las posiciones 1, 4, 5, 11, 13 y 15 y a medida que se realiza esto, se van anidando estas plantillas que poseen los símbolos diferentes ubicados en estas posiciones dentro del arreglo de plantillas, esta imagen anidada será presentada en la parte inferior del analizador. Se incluye un procedimiento de validación para evitar que un símbolo sea asociado con dos o más plantillas únicas y a medida que se realiza esta validación excluyente, se cuentan las frecuencias de cada símbolo y se agregan en un vector de frecuencia de manera que en cada posición "i" de este vector se ubica la suma vectorial de la fila de nuestra matriz de correlación validada. Finalmente terminado este análisis tenemos la siguiente interfaz presentada por el analizador:



Imagen 2.4. Análisis Completo

2.2.3 Revisión de Resultados

Para facilitar la revisión de los resultados finales, el usuario tendrá una nueva opción por medio del Botón “Seleccionar Símbolo”. Al hacer click en este botón aparecerán unos ejes de selección para dirigirlos hacia el símbolo que deseamos revisar:



Imagen 2.5 Revisión de Resultados

El símbolo elegido será resaltado en color rojo y presentado en la ventana superior izquierda y se mostrará la frecuencia del mismo indicando así las repeticiones halladas en la imagen original con la ayuda de la matriz de correlación binaria.

Al contar con la matriz de correlación así como nuestro vector de frecuencias, se hace posible realizar estas variaciones que nos facilitan el objetivo del Analizador que es justamente identificar dónde y cuántas veces se repiten los símbolos en la imagen de escritura hebrea cargada.

Al tener los ejes de selección y hacer click en el símbolo deseado, una variable recoge las coordenadas de las cuales se utiliza el valor de “x” que correspondería al orden de las columnas. Este valor es dividido para el ancho máximo de las plantillas Mcx previamente calculado con lo que obtenemos un valor entero que indica el orden del símbolo anidado.

En el caso que una imagen analizada nos provea más de 10 símbolos diferentes, aparecerán los botones “Siguiente” y “Anterior”.

Cuando se crean los vectores de frecuencia y de posición, se lleva un contador de símbolos diferentes, si este valor supera la cantidad de 10, se presenta sólo los primeros 10 símbolos y se hace visible el botón “Siguiente”. Cada vez que se presiona este botón se avanza de 10 en 10 hasta llegar al final mostrándose en el proceso el botón “Anterior” de manera que se puede retroceder para observar los 10 símbolos previos.

3. Conclusiones

Para probar el analizador se utilizaron imágenes descargadas en internet así como de libros digitales, se aplicó el proceso del Analizador observando los siguientes resultados:



Hebreo00



Hebreo01



Hebreo02



Hebreo03



Hebreo05

Verificando los resultados con la herramienta propia del Analizador tenemos el siguiente cuadro comparativo entre los datos reales y los obtenidos del analizador:

Imagen	Real		Obtenido del Analizador			Total
	Total	Diferentes	Dif Alto	Dif Medio	Dif Bajo	
Hebreo00	17	6	10	6	5	17
Hebreo01	173	24	28	13	5	170
Hebreo02	168	18	22	9	2	165
Hebreo03	37	16	16	5	4	37
Hebreo05	69	13	36	11	8	69

Imagen	Errores	
	Err Total	Err Dif
Hebreo00	0.00%	0.00%
Hebreo01	1.73%	16.67%
Hebreo02	1.79%	22.22%
Hebreo03	0.00%	0.00%
Hebreo05	0.00%	15.38%

“Total” hace referencia al conteo de todos los símbolos en la imagen y “Diferentes” o “Dif” se refiere a los símbolos diferentes que se repiten. Se hicieron análisis utilizando los 3 niveles de correlación y se han resaltado los valores con los que se calculó el error. Luego de esto podemos obtener las siguientes conclusiones:

1.- El valor apropiado para las comparaciones entre símbolos del lenguaje hebreo depende de la calidad de la imagen. En aquellas de mayor nitidez donde principalmente los símbolos tienen una buena caligrafía, se obtuvieron mejores resultados con el 75% es decir la correlación “Alta”, mientras que con aquellas donde el peso de la imagen es pequeño (29K) y la caligrafía hace que los símbolos difieran entre sí, se obtuvo mejores resultados con el 50% es decir la correlación “Media”, en ningún caso la “Baja” fue de utilidad ya que los errores se incrementaban enormemente.

2.- Los símbolos detectados como diferentes entre sí se ven afectados por la elección de etiquetas por imágenes conexas donde no se aplican mayores tratamientos para eliminar las conexiones mínimas. Esta afectación resulta más positiva que negativa en miras al objetivo de esta herramienta, ya que el observador podrá analizar visualmente cada símbolo.

3.- El análisis de símbolos de escritura exige una buena preparación de las imágenes originales, en nuestro caso las imágenes tomadas para prueba estaban bajas en ruido.

4.- Este sistema de análisis permite automatizar un estudio de símbolos donde un especialista podría obtener conclusiones variables en profundidad, ya que es factible descubrir patrones de uso de símbolos en el lenguaje en general o utilizados en cada escrito diferente

4. Recomendaciones

1.- Incluir mayor cantidad de herramientas de preparación de la imagen tales como modificaciones en la luminosidad, adaptaciones geométricas que permitan restaurar una imagen afectada por el ángulo que fue tomada la fotografía, editor de píxeles para eliminar áreas grandes que no son de interés y afectan en el análisis automático.

2.- Agregar herramienta de edición en el análisis de resultados, de esta manera los símbolos que el observador considera deben estar unidos o separados pueden ser trabajados de forma directa a fin de mejorar

las identificaciones de símbolos diferentes y las frecuencias de cada uno en la imagen global.

3.- Utilizar una herramienta de programación de bajo nivel a fin de optimizar los tiempos y recursos invertidos por el ordenador y poder analizar imágenes de mayor peso con gran cantidad de símbolos..

5. Agradecimientos

Agradezco primeramente a Dios y de manera muy especial a mis padres quienes obraron incluso más allá de sus fuerzas para apoyarme en mi formación profesional, así mismo a mi esposa que ha sido mi ayuda y apoyo en todos mis emprendimientos.

6. Referencia

[1] Mario I. Chacon; Procesamiento Digital de Imágenes; Editorial Trillas SA; 2007.

[2] Rafael C. Gonzalez y Richard E. Woods; Tratamiento Digital de Imágenes; Copublicación de Addison-Wesley Iberoamericana, S.A. y Ediciones Díaz de Santos, S.A.; 1996.

[3] Jorge Valverde Rebaza; Obtener el Histograma de una Imagen; <http://jc-info.blogspot.com/2009/02/obtener-el-histograma-de-una-imagen-en->; Junio 2010.

[4] Omar Sánchez; Modelos Control y Sistemas de Visión; <http://omarsanchez.net/histotransf.aspx>; Junio 2010.

[5] Mark Galer, Les Horvat; Tratamiento digital de imágenes; ANAYA MULTIMEDIA; 1 edición 06/2003.

[6] MathWorks – Company ; MathWorks; <http://www.mathworks.com/matlabcentral/fileexchange/>; Septiembre 2009.

[7] Samira Hervella Azouzi; Editor de Imágenes basado en Regiones. Aplicación en entorno Matlab; EUETIT.-Terrassa; 2006.

[8] Diego Barragán, Luis Moreno, Ángel Valdivieso; Matlab – Manejo de Simulink desde Matlab; www.matpic.com; Septiembre 2009.

[9] García de Jalón, José Ignacio Rodríguez, Jesús Vidal; Aprende Matlab como si estuviera en Primero; Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid; 2005.

Ing. Patricia Ximena Chávez Burbano
Septiembre 20 de 2010