



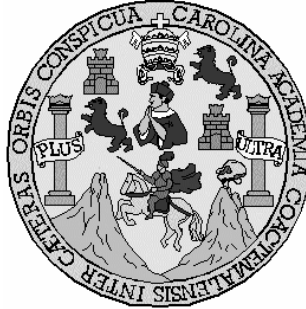
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Técnicas de reconocimiento de patrones aplicadas a glifos Mayas

OSCAR OSWALDO MORALES CALDERÓN
Asesorado por Inga. Elizabeth Domínguez

Guatemala, febrero de 2006

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**TÉCNICAS DE RECONOCIMIENTO DE PATRONES APLICADAS A
GLIFOS MAYAS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

OSCAR OSWALDO MORALES CALDERÓN

ASESORADO POR INGA. ELIZABETH DOMÍNGUEZ

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2006

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olimpo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yasminda Vides Leiva
SECRETARIO	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Marlon Antonio Pérez Turk
EXAMINADOR	Ing. Jorge Armin Mazariegos Rabanales
SECRETARIO	Inga. Marcia Ivonne Véliz Vargas

Guatemala, octubre 2005

Ing. Carlos Alfredo Azurdia Morales
Coordinador Comisión de Trabajos de Graduación
Escuela de Ciencia y Sistemas
Facultad de Ingeniería USAC

Estimado Ingeniero:

Por medio de la presente hago de su conocimiento, que he procedido a revisar el trabajo final de graduación titulado: **Técnicas de reconocimiento de patrones aplicadas a glifos mayas**, elaborado por el estudiante Oscar Oswaldo Morales Calderón y de acuerdo a mi criterio, se encuentra concluido y cumple con los objetivos propuestos para su desarrollo.

Agradeciendo de antemano la atención prestada, me suscribo a usted,

Atentamente.



Inga. Elizabeth Domínguez.

Asesor



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 2 de Noviembre de 2005

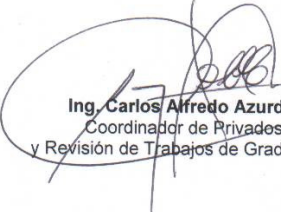
Ingeniero
Jorge Armin Mazariegos Rabanales
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Mazariegos:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **OSCAR OSWALDO MORALES CALDERON**, titulado: "**TECNICAS DE RECONOCIMIENTO DE PATRONES APLICADAS A GLIFOS MAYAS**", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del Revisor y del Licenciado en Letras, del trabajo de graduación titulado **“TÉCNICAS DE RECONOCIMIENTO DE PATRONES APLICADAS A GLIFOS MAYAS”**, presentado por el estudiante **OSCAR OSWALDO MORALES CALDERÓN**, aprueba el presente trabajo y solicita la autorización del mismo.

ID Y ENSEÑAD A TODOS



Ing. Jorge Armin Mazariegos Rabanales
DIRECTOR
INGENIERÍA EN CIENCIAS Y SISTEMAS

Guatemala, 24 de febrero de 2005

Universidad de San Carlos
de Guatemala



Facultad de Ingeniería
Decanato

Ref. DTG.63.05

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **TÉCNICAS DE RECONOCIMIENTO DE PATRONES APLICADAS A GLIFOS MAYAS**, presentado por el estudiante universitario **Oscar Oswaldo Morales Calderón**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos
Decano



Guatemala, marzo de 2006

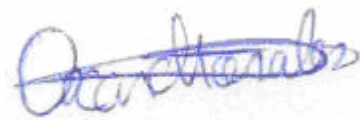
/cc

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

TÉCNICAS DE RECONOCIMIENTO DE PATRONES APLICADOS A GLIFOS MAYAS,

tema que me fuera asignado en la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería con fecha 9 de agosto de 2004



Oscar Oswaldo Morales Calderón

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO.....	VII
RESUMEN.....	IX
OBJETIVOS	XI
INTRODUCCIÓN.....	XIII
1. MARCO TEÓRICO	1
1.1 Arquitectura maya	1
1.2 Numeración	2
1.3 Escritura	3
1.4 Términos y definiciones para el proyecto	4
1.5 Glifos y números de glifo.....	5
1.6 Reconocimiento de patrones.....	6
1.7 Pasos para el reconocimiento de patrones	6
1.8 Subproblemas del reconocimiento de patrones	7
2. IMÁGENES DIGITALES.....	9
2.1 Resolución.....	10
2.2 Dimensiones de píxel	10
2.3 Profundidad de bits	11
2.4 Compresión	14
3. RECONOCIMIENTO DE GLIFOS	17
3.1 Carga de imágenes	18
3.2 Extracción de contornos.....	18
3.2.1 Representación de contornos	20

3.3 Detección de bordes	22
3.3.1 Umbral.....	23
3.3.2 Umbral adaptable	23
3.3.2 Sobel	24
3.3.4 Laplaciano	26
3.3.5 Canny.....	27
3.4 Procesamiento de contornos	31
3.5 Características.....	34
3.5.1 Teoría de los momentos.....	35
3.5.2 Momentos de Orden 0.....	36
3.5.3 Momentos de Orden 1.....	36
3.5.4 Momentos de Orden 2.....	37
3.5.6 Momentos de Orden 3.....	38
3.6 Clasificador	39
3.7 Asignación de puntaje:.....	40
4. MANUAL DE USUARIO	43
4.1 Clasificar una imagen.	44
4.2 La librería.....	46
4.2.1 Navegar en la librería	47
4.2.2 Búsquedas en la librería.....	48
4.2.3 Edición de glifos	48
4.2.4 Eliminación de glifos.....	48
4.2.5 Árbol de glifos	49
4.3 Añadir un glifo a la librería	49
4.4 Opciones del programa.....	51
4.4.1 Opciones generales	51
4.4.2 Umbral.....	52
4.4.3 Umbral adaptable	53
4.4.4 Sobel	53

4.4.5 Laplace	54
4.4.5 Canny	54
4.4.6 Mensajes	54
4.5 Requerimientos del sistema	54
4.6 Instalación	55
CONCLUSIONES	57
RECOMENDACIONES.....	59
BIBLIOGRAFÍA	61
APÉNDICES	63
OpenCV	63

ÍNDICE DE ILUSTRACIONES

Figuras

1. Ejemplo de numeración maya.....	2
2. Ejemplo de escritura maya.....	3
3. Imagen formada por píxeles	9
4. Aumento de una imagen.....	10
5. Dimensiones de píxel.....	11
6. Escala de grises.....	12
7. Imagen binarizada (1 bit) Imagen a escala de grises (8 bits).....	13
8. Capas RGB de la imagen (Rojo, verde y azul)	13
9. Ejemplo de compresión con pérdida.....	15
10. Proceso de clasificación.	17
11. Tipos de conectividad	19
12. Regiones en una imagen binarizada.....	19
13. Vecindario para un píxel dado	21
14. Cadena de la curva: 34445670007654443	21
15. Dx y Dy como mascararas	25
16. Los operadores sobel para Dx y Dy	26
17. Mascara para el Laplaciano de una imagen	26
18. Sectores de gradiente.....	28
19. Pasos del algoritmo de J. Canny	29
20. Comparación entre los diferentes algoritmos.....	30
21. Salida del algoritmo Rosenfeld-Johnston.....	32
22. De izquierda a derecha Imagen de entrada y salida para el algoritmo Teh-Chin	33
23. Ejemplo de figuras equivalentes	34
23. Pantalla principal del programa.....	43

24. Botones de la pantalla principal.....	44
25. Imagen original y los contornos extraídos	44
26. Resultados de la clasificación	45
27. La librería de glifos	47
28. Árbol de glifos.....	49
29. Nuevo glifo en la librería.....	50
30. Opciones del programa	52
31. El programa instalador	55
32. Categoría Preprocessor.....	65
33. Categoría Input.....	65
34. Propiedades para Win32 Debug	66
35. Propiedades para Win32 Release.....	66
36. Insertar un proyecto	67
37. Definir dependencias.....	67

Tablas

I. Cantidad máxima de tonos.....	13
----------------------------------	----

GLOSARIO

B

Bit Contracción de las palabras inglesas *binary digit* -dígito binario-, que designa a la unidad mínima de información que puede representarse físicamente.

Bitmap Un formato de imagen no comprimido dentro del sistema operativo Windows®; los bitmaps ofrecen excelente calidad de imagen pero frecuentemente necesitan gran cantidad de espacio en el disco.

D

Dpi Contracción de las palabras inglesas *dots per inch*, puntos por pulgada, es una unidad de medida estándar de resolución de imágenes.

F

Filtro Fórmula matemática aplicada a una imagen digital. La mayoría de los editores de imágenes ofrecen filtros que pueden hacer cambios radicales en la apariencia de una fotografía.

Fonograma Carácter o símbolo usado para representar una palabra, sílaba o fonema.

I

Ideograma Imagen convencional o símbolo que representa un ser, objeto o una idea, pero no palabras o frases que los representen.

J

JPEG

Formato estándar de compresión de datos de imágenes desarrollado por *Joint Photographic Experts Group* -Grupo conjunto de expertos fotográficos- de ahí el nombre JPEG. En sentido estricto, JPEG no es un formato de archivo sino un método de compresión que se emplea dentro de un formato de archivo, como el formato EXIF-JPEG, habitual en las cámaras digitales. Este formato produce pérdidas parciales de la calidad de la imagen a fin de conseguir índices de compresión muy elevados. Si se configura la cámara digital con un formato JPEG de gran calidad y baja compresión, la pérdida de calidad no suele ser perceptible a simple vista.

L

Logograma

Signo gráfico que representa una palabra sin dar ninguna indicación de su pronunciación.

P

Patrón

Un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas.

Píxel

Punto único en imágenes digitales, una imagen digital esta hecha de miles de píxeles

R

RGB

Contracción de las palabras inglesas *RED GREEN BLUE* -rojo, verde y azul- es un sistema de representación de colores basado en la combinación de estos tres colores para la formación de los demás.

RESUMEN

En este trabajo se propone crear un programa para facilitar la clasificación de glifos usando como entrada una imagen digital, de manera que una persona que quiera identificar un glifo no tenga que revisar dentro de cientos de glifos ya clasificados para encontrar el apropiado, si no reducir la búsqueda en el mejor de los casos a uno y de no ser así, a una cantidad mucho menor, usando como herramienta la computadora y mas, específicamente, técnicas de reconocimiento de patrones.

El primer capítulo esta dividido en dos partes, en la primera parte se dan a conocer generalidades sobre la cultura de los mayas especialmente de su escritura y numeración , en la segunda se explica en forma general en que consiste el reconocimiento de patrones, sus pasos básicos y algunos de los problemas comunes para el mismo. En el segundo capítulo se exponen conceptos fundamentales acerca de las imágenes digitales necesarios para una mejor comprensión de los siguientes capítulos. En el tercer capítulo se desarrolla una explicación del proceso de clasificación usado en el programa, que básicamente consta de, la carga de la imagen, extracción de contornos, representación de contornos, extracción de características, comparación de características y, por ultimo, la presentación de los resultados. El cuarto capítulo es un manual de usuario para las distintas opciones disponibles en el programa.

OBJETIVOS

General

- Creación de un programa que permita clasificar glifos mayas

Específicos

1. Diseño de un modelo que permita extraer las principales características de un glifo
2. Procesamiento previo a las imágenes para simplificar las operaciones siguientes sin perder información importante.
3. Utilización de distintas técnicas de extracción de bordes.
4. Extracción de características principales para reducir la cantidad de información a ser procesada
5. Evaluar las características para clasificar el glifo

INTRODUCCIÓN

El presente trabajo de investigación, referente a la creación de un programa que utiliza técnicas de reconocimiento de patrones para la clasificación de glifos Mayas, se inicia con nociones básicas acerca de la cultura de los mayas dando énfasis a la escritura, ya que, esta se compone de símbolos susceptibles a la clasificación por medio de técnicas de reconocimiento de patrones.

Luego, se explicarán fundamentos de imágenes digitales, como una base al describir los pasos que se tomaron para clasificar las imágenes de los glifos, estos incluyen la reducción de colores de la imagen, la extracción y representación de contornos, la extracción de características y el método de puntuación de éstas, por último se obtendrá una explicación respecto de cómo utilizar el programa.

1. MARCO TEÓRICO

La civilización Maya no constituía un estado unificado, sino que se organizaba en varias ciudades-estado independientes entre si, en un territorio de aproximadamente 400,000 kilómetros cuadrados, situado en la región denominada Meso América, la cual abarca la Península de Yucatán, el estado de Quintana Roo, la Mayor parte de Tabasco y Chiapas, el istmo de Tehuantepec en México, toda Guatemala, Belice, la parte occidental de El Salvador y Honduras, además de una pequeña parte de Nicaragua, este territorio presenta una gran variedad geográfica: montañas, pantanos, planicies, selvas tropicales, bosques de altura, etc., por lo cual existen climas, suelos, lluvias y vegetaciones muy diferentes en la región.

Para su estudio comúnmente la historia maya es dividida en tres etapas

- El periodo preclásico (1600 a.C. al 300 d.C.)
- El periodo clásico (300 al 900 d.C.)
- El periodo posclásico (900 al 1542 d.C.)

1.1 Arquitectura Maya

Las construcciones mayas se hicieron básicamente de madera y piedra. Entre las maderas las más utilizadas fueron la caoba y el zapote, por ser estas muy resistentes a los ataques de las termitas. Entre las piedras usaron caliza, arenisca, mármol. Realizaron todo tipo de construcciones: palacios rectangulares y alargados, templos, juegos de pelota, calzadas que unían las ciudades principales, fortificaciones, baños de vapor.

Se conservan importantes pirámides escalonadas en piedra. En lo alto de las pirámides generalmente se encuentra un templo. Fueron decoradas con pinturas y relieves de una variada gama de colores. Algunos de estos son inscripciones de la escritura jeroglífica maya. Las construcciones más importantes son Copan, Quiriguá, Piedras Negras, Palenque y Tikal.

1.2 Numeración

Una de las aportaciones más geniales de la civilización maya fue su sistema de numeración. En lugar de ser decimal es vigesimal y solamente se necesitan tres símbolos para representar cualquier número: el punto, la raya y una especie de concha para representar el cero. El concepto de cero es una de las grandes aportaciones, que otras civilizaciones, como la hindú, ya conocían, mientras que los europeos de la misma época no descubrieron hasta mucho más tarde. Igual que la cultura occidental y los babilonios, los Mayas utilizan el concepto de lugar valor, pero mientras la cultura occidental utiliza un lugar valor que va en aumento de derecha a izquierda en múltiplos de diez, el lugar valor del sistema Maya aumenta en múltiplos de veinte, por ejemplo; 1, 20, 400, 8000, etc. y en vertical de abajo hacia arriba. La figura 1 ejemplifica este sistema de numeración.

Figura 1. Ejemplo de numeración maya

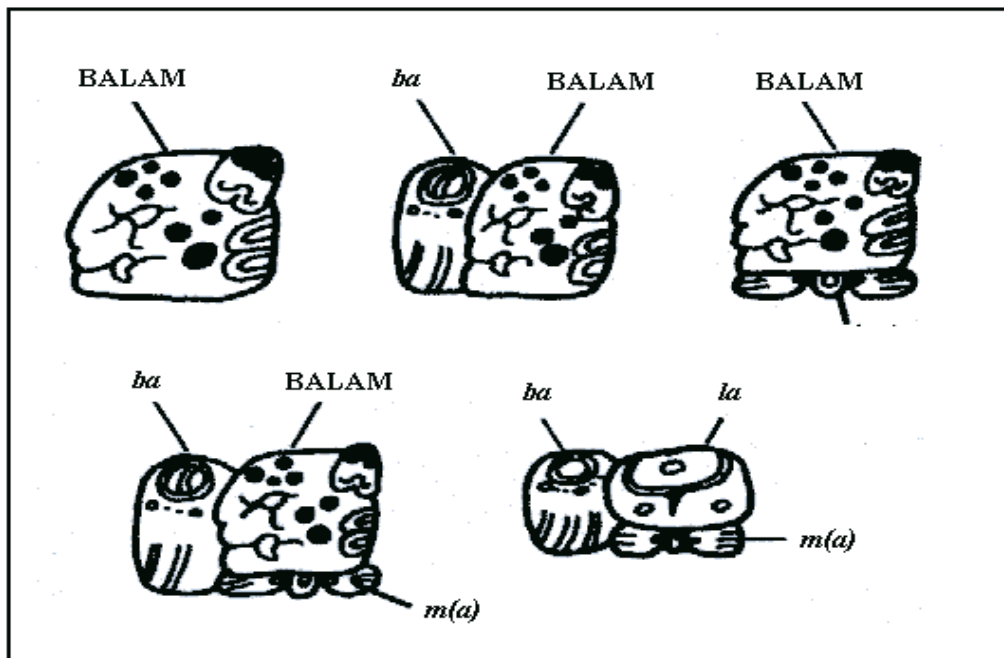
Ejemplos de numeración Maya										
•	••	•••	••••	—	•	••	•••	••••	—	•
1	2	3	4	5	6	7	8	9	10	0
• = 1 x 20 = 20 • = 0 x 1 = 0 Total: 20				••• = 3 x 20 = 60 — = 1 x 10 = 10 Total: 70			— = 10 x 20 = 200 •••• = 9 x 1 = 9 Total: 209			
•• = 2 x 400 = 800 • = 0 x 20 = 0 — = 19 x 1 = 19 Total: 819						•••• = 9 x 8.000 = 72.000 • = 0 x 400 = 0 ••• = 3 x 20 = 60 ••• = 3 x 1 = 3 Total: 72.063				

Fuente: <http://www.proel.org/alfabetos/maya.html>, marzo 2004.

1.3 Escritura

La escritura Maya se supone es del tipo mixto (como la de los egipcios), esto significa que los glifos pueden representar tanto ideas u objetos (ideogramas, pictogramas), como sonidos (fonogramas). Una de las razones de la complejidad de la escritura Maya radica en el hecho que una misma palabra puede escribirse de varias formas. Además, los glifos individuales están con frecuencia unidos entres sí (como en la escritura china), en la escritura Maya lo están tan íntimamente que sólo un experto consigue separar visualmente las partes de las que se compone. La figura 2 muestra cinco formas de escribir la palabra **balam** (jaguar).

Figura 2. Ejemplo de escritura maya



Fuente: <http://www.proel.org/alfabetos/maya.html>, marzo 2004.

1.4 Términos y definiciones para el proyecto

En el presente trabajo la palabra inscripción es usada para referirse al texto jeroglífico que se encuentra en algún medio. O alguna representación del mismo. La palabra transcripción se refiere a representar la inscripción en una forma diferente. La cual conserva la información importante de la inscripción pero puede ser significativamente distinta. Las dos inscripciones y transcripciones son referidas como textos, pero las transcripciones son en ocasiones llamadas cadenas.

La transcripción se basa en la condición que el elemento gráfico de una inscripción son instancias o símbolos de una categoría abstracta o tipo que existe de manera independiente de cualquier inscripción particular, la cual; la transcripción trata de identificar. De tal manera que el transcribir un texto es clasificar sus elementos dentro de las categorías genéricas. Un grafema es un tipo de signo grafico, las inscripciones están hechas de representaciones de grafemas.

Es importante recordar que en este contexto, una transcripción busca únicamente representar la geometría y las propiedades grafemitas de una inscripción tan completamente y precisamente como sea posible. Con muy pocas excepciones no intenta asignar valores a los grafemas. La asignación de valores –fonéticos, semánticos, gramaticos etc.- toma lugar en el nivel de decodificación. Los niveles de transcripción, decodificación, traducción e interpretación deben ser tratados como momentos separados, casi independientes del proceso de investigación epigráfica, uno de los usos de una transcripción digital es que puede ser fácilmente procesado por un programa de traducción, el programa pretende ayudar a una persona a clasificar un glifo dentro de unas categorías previamente definidas, así como la posibilidad de agregar nuevas categorías.

Un glifo esta definido como la unidad mínima de una inscripción, el cual puede estar solo o en combinación con otros glifos, en escritura clásica Maya, uno o mas glifos se combinan para crear bloques, los cuales juntos forman textos. Cuando se usa en referencia a inscripciones la palabra texto se refiere a grupos contiguos de bloques de glifos. A pesar de que dos grupos de bloques pueden ser parte de la misma narración, son tratados como textos separados para propósitos de traducción.

1.5 Glifos y números de glifo

Para la clasificación se propone utilizar el catálogo de glifos de Thompson con algunas modificaciones sugeridas por Kurbujun. Estos números son definidos como los números-g. Además de las amplias categorías definidos en el esquema de Thompson (afijos, principales y rostros) el numero-g no contiene información sobre los glifos, no pretende ser un identificador único de los glifos si no que actué como un índice o un método de búsqueda, los números g se dividen en:

- 1-327 = Glifos afijos identificados
- 328-499 = Glifos afijos recientemente identificados
- 500-856 =Glifos principales identificados
- 857-999 = Glifos principales recientemente identificados
- 1000-1087=Glifos de rostro identificados
- 1088-1299= Glifos de rostro recientemente identificados
- Glifos no identificados = ?.
- Glifos inidentificables =999.
- Glifos inferidos = {numero-g inferido}.

Se entiende como glifos inidentificable aquellos que están erosionados, borrados o simplemente perdidos. Ocasionalmente estos glifos pueden ser identificados por inferencia.

1.6 Reconocimiento de patrones

Se conoce como reconocimiento de patrones a la disciplina cuyo objetivo es clasificar objetos en un número predefinido de categorías o clases, las técnicas de reconocimiento de patrones son una parte importante de los sistemas inteligentes y son utilizados en:

- Toma de decisiones
- Clasificación de objetos
- Reconocimiento facial
- Reconocimiento de caracteres
- Visión por computadora
- Otros...

1.7 Pasos para el reconocimiento de patrones

Una aplicación típica de reconocimiento de patrones, incluye por lo general algunos, o todos de los siguientes pasos.

- Modelado:
Se necesita de una descripción matemática del objeto a clasificar
- Procesamiento previo:
Las imágenes son procesadas previamente para simplificar operaciones posteriores, esto sin que resulte una pérdida de información importante.
- Segmentación:
Imágenes de diferentes objetos son aisladas una de otra y del fondo

- Limpieza de ruidos:
El ruido es causado por la aleatoriedad del sensor, y puede reducir la confiabilidad del trabajo
- Extracción de características:
Reducción de los datos mediante la cuantificación de ciertos atributos, particularidades o características.
- Clasificación:
Los atributos son pasados a un clasificador, que los evalúa, presenta y hace una decisión.

Es importante hacer notar que la mayoría de aplicaciones de este tipo incluyen alguna forma de aprender nuevos patrones. Esto es debido a la imposibilidad de que en el momento de construcción del sistema, se conozcan todas las categorías, sin que puedan aparecer nuevas.

1.8 Subproblemas del reconocimiento de patrones

Para la construcción de un sistema de reconocimiento de patrones se debe de enfrentar una serie de problemas entre los mas comunes tenemos.

- Ruido:
Se conoce como ruido a datos que no forman parte del objeto original
- Selección del modelo y de las características
Existe una gran variedad de modelos (basados en alguna combinación de características), el problema radica en como decidir, rechazarlos, aceptarlos o probar nuevos modelos, y si existe alguna forma de automatizar este proceso.
- Características faltantes:
Como programar al clasificador en caso falte alguna característica
- Segmentación:
Los objetos tienen que ser separados del fondo y de otros objetos.

- Contexto:
Un buen clasificador debe ser capaz de utilizar el contexto para lograr una identificación.
- Invariabilidad:
El clasificador debe buscar una independencia de la posición del objeto, tamaño u orientación.
- Costos y riesgos:
El clasificador es utilizado para recomendar una acción, cada acción tiene asociada un costo o riesgo, el clasificador debe estar diseñado para minimizar este costo o riesgo.
- Complejidad computacional:
El costo computacional debe ser tomado en cuenta ante todo la relación entre exactitud y desempeño

2. IMÁGENES DIGITALES

Las **Imágenes digitales** son fotos electrónicas o escaneadas de documentos, fotografías, escritura manual, escritura impresa o ilustraciones. Se crean a partir de un muestreo de la imagen original para confeccionar un mapa de ella en forma de cuadrícula de puntos o elementos de la figura llamados píxeles. A cada píxel se le asigna un valor tonal (negro, blanco, matices de gris o de color) para asignar este valor se recurre a un promedio de todos los valores de color que están dentro del píxel, por lo que algo de información se pierde, Los píxeles están representados en un código binario. Los dígitos binarios (**bits**) para cada píxel son almacenados por la computadora en una secuencia, con frecuencia se utiliza algún algoritmo para comprimir esta información. Luego la computadora lee los bits para producir la imagen para su visualización o impresión. La figura 3 representa una imagen creada con valores de píxel 0 y 1

Figura 3. Imagen formada por píxeles

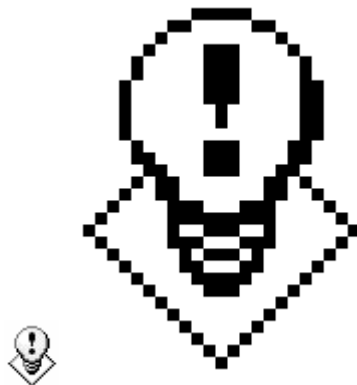
1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Fuente: http://www.pages.drexel.edu/~weg22/can_tut.html, abril 2004.

2.1 Resolución

La **resolución** en las imágenes digitales es la capacidad de distinguir detalles en la imagen, en otras palabras entre mayor resolución mas detalles se pueden observar en la imagen. Por lo general, la frecuencia espacial a la cual se realiza el muestreo de una imagen digital (la frecuencia de muestreo) es un buen indicador de la resolución. Este es el motivo por el cual **dots-per-inch** (puntos por pulgada) (dpi), o **pixels-per-inch** (píxeles por pulgada) (ppi) son términos comunes para expresar la resolución de imágenes digitales. Dentro de ciertos límites, el aumento de la frecuencia de muestreo también ayuda a aumentar la resolución. En la figura 4 puede observarse como al aumentar el tamaño de la imagen pueden observarse los píxeles que conforman la imagen original.

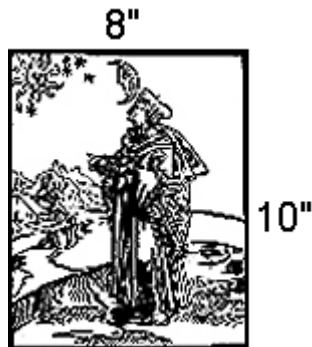
Figura 4. Aumento de una imagen



2.2 Dimensiones de píxel

Las **dimensiones de píxel** son las medidas horizontales y verticales de una imagen, expresadas en píxeles. Las dimensiones de píxel se pueden determinar al multiplicar el ancho y la altura por el **dpi**.

Figura 5. Dimensiones de píxel.



Fuente: <http://www.library.cornell.edu/preservation/tutorial-spanish/intro/intro-03.html>, abril 2004.

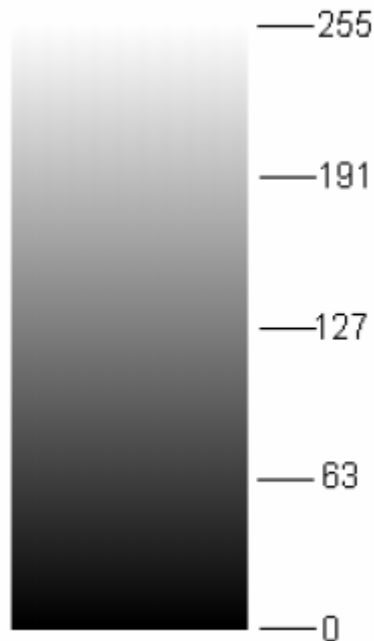
Un documento de 8 x 10 pulgadas que se escanea a 300 **dpi** posee dimensiones de píxel de 2400 píxeles (8 pulgadas x 300 **dpi**) por 3000 píxeles (10 pulgadas x 300 **dpi**).

2.3 Profundidad de bits

La **profundidad de bits** es determinada por la cantidad de bits que son utilizados para definir cada píxel. Entre mas bits sean utilizados para representar el color, tanto mayor será la cantidad de tonos (escala de grises o color) que puedan ser representados. Una imagen digital puede construirse en blanco y negro (forma binarizada), a escala de grises o a colores. Una **imagen binarizada** está representada por píxeles que constan de 1 bit cada uno, con esto solamente se pueden representar dos tonos (típicamente negro y blanco), utilizando los valores 0 para el negro y 1 para el blanco o viceversa. Una imagen a **escala de grises** está compuesta por píxeles representados por más de un bit de información, típicamente se representan usando entre 2 a 8 bits.

Por ejemplo en una imagen en la cual se utilizan dos bits para representar el color de un píxel, solamente existen cuatro combinaciones posibles: 00, 01, 10 y 11. Si se define que 00 representa el negro, luego 11 representa el blanco, entonces 01 es igual a gris oscuro y 10 es igual a gris claro. La profundidad de bits es dos, pero la cantidad de tonos que pueden representarse es 2^2 es decir 4. Con 8 bits, pueden asignarse 256 (2^8) tonos diferentes a cada píxel.

Figura 6. Escala de grises



Una **imagen a color** normalmente utiliza una profundidad de bits entre 8 y 24 bits o más. En una imagen de 24 bits, los bits normalmente se dividen en tres grupos: 8 para el rojo, 8 para el verde, y 8 para el azul esto es lo que normalmente se denomina RGB, por las siglas de estos colores en ingles. Para representar otros colores se utilizan combinaciones de esos bits. Una imagen de 24 bits ofrece 16,7 millones (2^{24}) de valores de color.

En la figura 7 puede observarse la diferencia entre una imagen con una profundidad de bits de uno (dos colores, blanco y negro) con la misma imagen con una profundidad de 8 bits. Mientras que en la figura 8 se puede notar las diferencias entre las distintas capas de una imagen RGB.

Figura 7. Imagen binarizada (1 bit) Imagen a escala de grises (8 bits)



Figura 8. Capas RGB de la imagen (Rojo, verde y azul)



Fuente: <http://www.softintegration.com/products/thirdparty/opencv/>, mayo 2004.

La tabla I muestra la cantidad máxima de tonos que pueden ser representados con distintas profundidades de bits.

Tabla I. Cantidad máxima de tonos

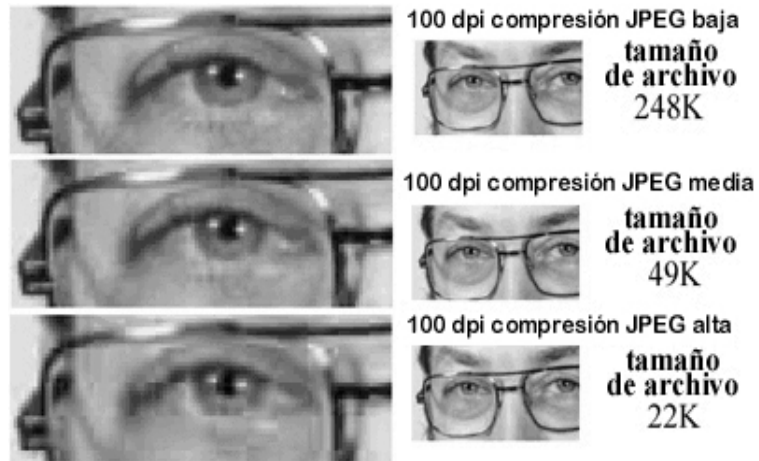
Profundidad	Formula	Cantidad máxima
1	2^1	2
4	2^4	16
8	2^8	256
16	2^{16}	65,536
24	2^{24}	16,777,216
32	2^{32}	4,294,967,296

2.4 Compresión

La compresión es una representación matemática del código binario de una imagen digital sin comprimir, lo que da como resultado una representación resumida de la imagen, esto es común debido a que una imagen sin comprimir suele ser de un tamaño demasiado grande, desperdiciando recursos del sistema.

Los sistemas de compresión se pueden dividir en dos tipos, los sistemas sin pérdida de información o en sistemas con pérdida de información. Los sistemas sin pérdida, abrevian el código binario sin desechar información, por lo que, cuando se muestra la imagen, ésta es idéntica *bit a bit* a la imagen original. Los sistemas con pérdida logran abreviar aun más la imagen, pero con un costo, la pérdida de algo de información, estos algoritmos tratan de desechar únicamente la información con menos importancia, basados en un entendimiento de la percepción visual. La compresión con pérdida típicamente se utiliza con imágenes a colores, y en particular imágenes de colores continuos en donde esta pérdida de información es casi imperceptible al ojo humano.

Figura 9. Ejemplo de compresión con pérdida.



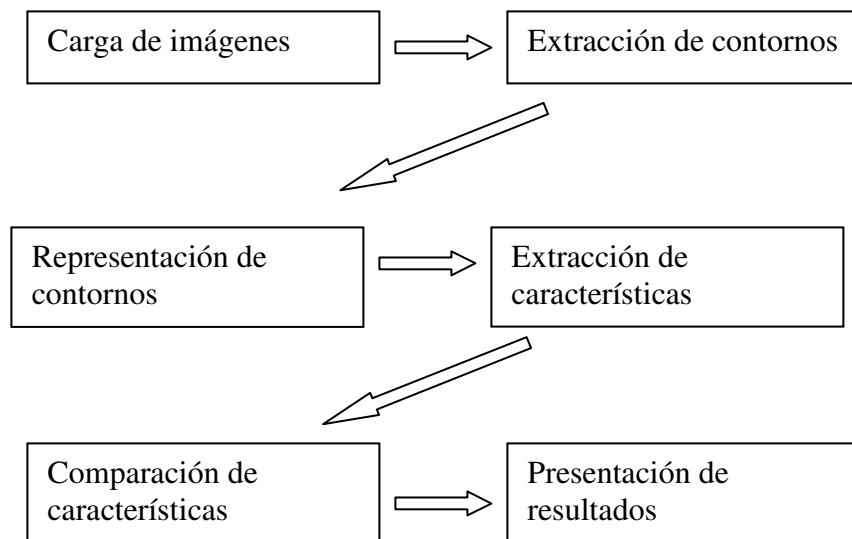
Fuente: Fuente: <http://www.softintegration.com/products/thirdparty/opencv/>, febrero 2004.

En la figura 9 se pueden observar los efectos de la compresión JPEG que utiliza un formato con pérdida de información, a mayor compresión de la imagen mas detalles de la misma se pierden, nótese que en la compresión mas baja se observan mas detalles de la imagen, por ejemplo las pestañas y en la compresión mas alta las pestañas han desaparecido, sin embargo en la compresión mas alta, el tamaño del archivo es mucho menor que en la compresión mas alta.

3. RECONOCIMIENTO DE GLIFOS

El proceso de clasificación consta de varias etapas las cuales en forma resumida son: carga de imágenes, extracción de contornos, representación abreviada de los contornos de la imagen, extracción de características, comparación de las características y por ultimo la presentación de los resultados, en la figura 10, se presenta un esquema del proceso de clasificación de los glifos mayas.

Figura 10. Proceso de clasificación.



Para el desarrollo de este proyecto las siguientes simplificaciones fueron realizadas.

- A pesar que de los glifos normalmente se encuentran en grupos los cuales se denominan bloques, el programa tendrá como ingreso imágenes de glifos aislados.

- El proyecto pretende ser una ayuda para la clasificación de glifos, no es un traductor o un intérprete, aunque puede definirse como un primer paso para la construcción de uno.

3.1 Carga de imágenes

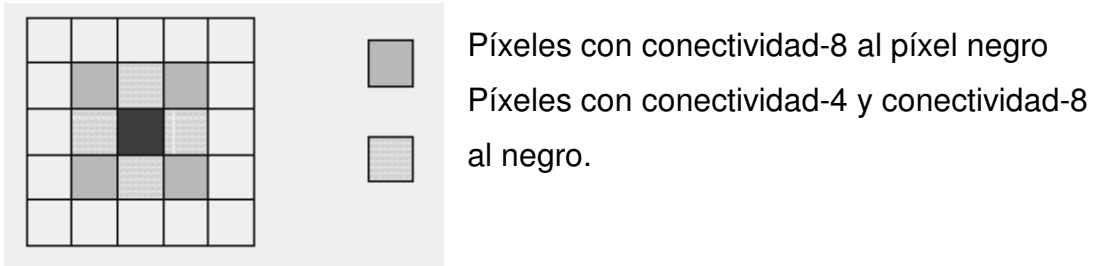
Para la carga de imágenes en el proyecto se permitirá hacer de cualquiera de los siguientes formatos de archivo (mapas de bits – BMP, imágenes - JPEG, JPG, JPE y PNG) estos formatos soportan imágenes con dos colores hasta imágenes con varios millones de colores por lo que el primer paso para simplificar el problema, es convertir la imagen original (posiblemente a colores) a una representación a escala de grises (con máximo de 256 tonos de gris), para hacer esto la siguiente fórmula es utilizada.

$$\text{RGB[A]} \rightarrow \text{Grises: } Y = 0.212671 * R + 0.715160 * G + 0.072169 * B$$

3.2 Extracción de contornos

Los algoritmos que se pueden utilizar en el programa para buscar contornos en imágenes, trabajan únicamente imágenes binarizadas o bitonales, es decir, imágenes en las cuales todos sus píxeles, solo pueden tener dos valores típicamente 0 o 1. Para este tipo de imágenes se pueden definir dos tipos de conectividad, conectividad-4 y conectividad-8. Dos píxeles con coordenadas (x',y') y (x'',y'') tienen conectividad-4 si y solo si $|x' - x''| + |y' - y''| = 1$ y con conectividad-8 si $\max(|x' - x''|, |y' - y''|) = 1$. En la figura 11 se puede observar los distintos tipos de conectividad al píxel negro del medio.

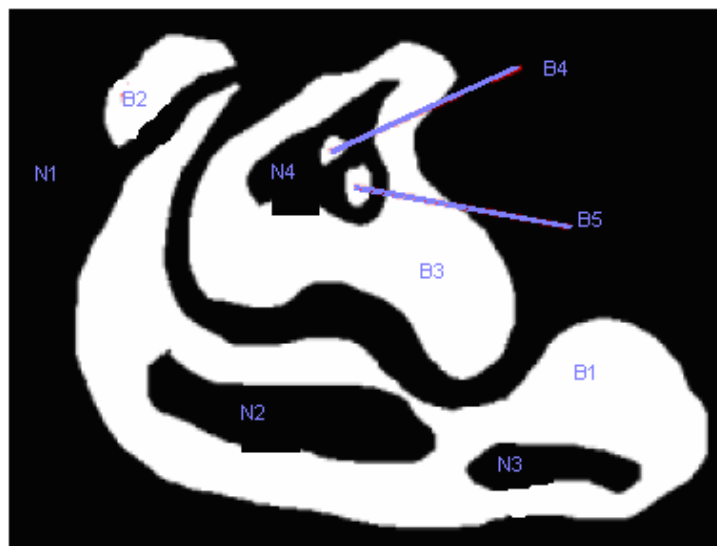
Figura 11. Tipos de conectividad



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, marzo 2004.

Usando estas relaciones es posible separar a la imagen en varios componentes no traslapados, cada componente esta formado por píxeles de igual valor ya sea 1 (usualmente blanco) o 0 (usualmente negro), cualquier par de píxeles, del mismo componente pueden conectarse mediante una secuencia de píxeles con conectividad-4 o conectividad-8, dicho de otra manera existe un camino entre los dos. La figura 12 muestra un ejemplo de regiones en una imagen binarizada.

Figura 12. Regiones en una imagen binarizada



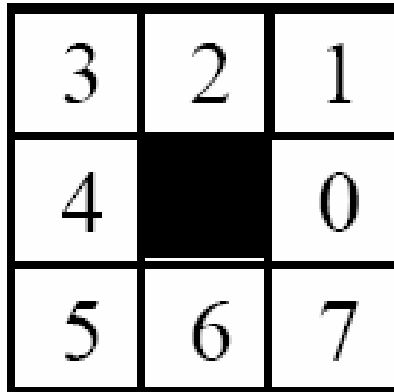
En la figura 12 los Componentes-1 B1, B2, B3 están dentro del marco (componente-0 N1), es decir dentro de N1. Los componentes-0 N2 y N3 están dentro de B1. Los componentes-1 B4 y B5 están dentro de N4, que esta dentro de B3, así que estos componentes están indirectamente dentro de B3, sin embargo ni B4 o B5 se encierran entre si lo que significa que están en el mismo nivel.

En vista de que los componentes-0 son complementarios a los componentes-1, y los componentes-1 están ya sea anidados o no se intersecan, se suele considerar únicamente a los componentes-1, y solamente su estructura topológica es estudiada, considerando a los componentes-0 como el fondo de la imagen. Cada componente-1 tiene solamente un borde externo que lo separa del componente-0 de alrededor, y cero o mas bordes internos que separa los componentes-1 de los componentes-0 que rodea, resulta obvio que el borde externo y los bordes internos dan una descripción completa del componente, es por esta razón, que todos los bordes, también llamados contornos, de todos los componentes junto con información de la jerarquía, conforman una representación compacta de la imagen.

3.2.1 Representación de contornos

Uno de los métodos para la representación de estos contornos se conoce como la cadena de la curva. Para cualquier píxel en la imagen, todos sus vecinos pueden ser enumerados con los números del 1 al 7. En la figura 13 se muestran como se enumeran los vecinos del píxel negro.

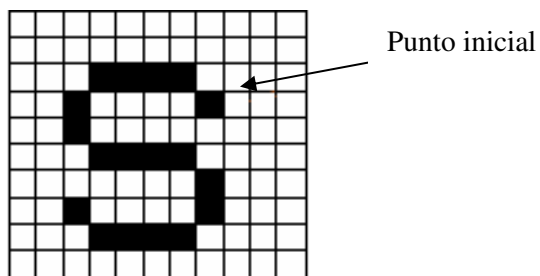
Figura 13. Vecindario para un píxel dado



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

Si se considera al borde como una secuencia de puntos conectados, entonces el borde puede ser descrito como las coordenadas del punto inicial, seguido de códigos (de 0 a 7) que indican la posición del próximo punto relativo al último. En la figura 14 se puede observar como se puede describir totalmente el borde como una secuencia de números que se denomina cadena de la curva.

Figura 14. Cadena de la curva: 34445670007654443



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

Representación Poligonal

Una opción diferente es llamada Representación Poligonal en la cual la curva es codificada como una secuencia de puntos, vértices de una polilínea. Esta alternativa es frecuentemente una mejor elección para manipular y analizar contornos que la cadena de la curva. Sin embargo esta representación es difícil de lograr sin tener mucha redundancia, por lo que algoritmos que aproximan la cadena de la curva con polilíneas pueden ser usados.

3.3 Detección de bordes

En una imagen digital se puede definir a un borde o contorno como la frontera entre dos regiones con niveles de gris relativamente distintos, entonces un píxel pertenece a un borde si existe una marcada diferencia entre su nivel de gris y el de sus vecinos, entre mayor sea la diferencia mejor, es importante señalar que la detección de bordes en una imagen digital reduce significativamente la cantidad de datos y filtra información no útil, mientras que preserva importantes propiedades estructurales de la imagen, haciendo mas sencillas operaciones posteriores, al final se desea una versión binarizada de la imagen donde un tono (usualmente 0) es el fondo de la imagen y el otro tono (usualmente 1) son los bordes o contornos.

3.3.1 Umbral

Una técnica simple para el binarizado de una imagen es aquella que se conoce como el Umbral de una imagen, si se tiene una imagen digital en escala de grises (donde 0 define al color negro y 255 define al color blanco) y se desea obtener una imagen bitonal, se procede de la siguiente manera, primero se crea una imagen con las mismas dimensiones que la imagen original, luego usando un valor límite se compara cada píxel de la imagen original con este, de manera que el valor de la nueva imagen es igual a 1 si el valor es mayor que el límite de lo contrario el valor del píxel es 0.

Para los propósitos del programa existen ciertas limitaciones con el uso de esta técnica, la más importante a tomar en cuenta es, esta técnica no está precisamente buscando bordes en la imagen, lo que busca son aquellos píxeles cuyo valor sea mayor que el valor límite, en segundo lugar es complicado una buena estimación del valor límite ya que este es el mismo para toda la imagen, pero puede ser útil si se conoce de antemano que las imágenes a tratar carecen de ruido, y que los bordes aparecen definidos en la imagen y tienen el mismo (o al menos muy parecido) valor de gris.

3.3.2 Umbral adaptable

Esta técnica ataca uno de los problemas del método anterior el del valor constante del límite, esta técnica calcula este valor para cada píxel de la imagen, tomando en cuenta a los vecinos de este píxel, el algoritmo trabaja de la siguiente manera. Definiendo f_{ij} $1 \leq i \leq I$, $1 \leq j \leq J$ como la imagen de entrada para cada píxel i,j , la media $m_{i,j}$, y varianza $v_{i,j}$ están definidas como.

$$m_{ij} = \sum_{s=-o}^p \sum_{t=-t}^t f_{i+s, j+t}, v_{i,j} = \frac{1}{2p} \sum_{s=-p}^p \sum_{t=-t}^t |f_{i+s, j+t} - m_{ij}|$$

En un vecindario de P x P. Entonces el limite local para el píxel i,j es $t_{ij} = m_{ij} + v_{ij}$ cuando $v_{ij} > v_{min}$, y $t_{ij} = t_{ij-1}$ cuando $v_{ij} \leq v_{min}$ donde v_{min} es el valor mínimo de la varianza ya con este valor limite del píxel se procede de igual manera que en Umbral, este método a pesar de proporcionar el valor limite, sigue sin ser un algoritmo que busque bordes en la imagen, por lo básicamente se puede usar en con las mismas condiciones que el algoritmo anterior.

3.3.2 Sobel

La idea básica para encontrar bordes en una imagen digital es el encontrar píxeles donde exista un cambio brusco entre los niveles de gris, una imagen digital puede definirse como una función de dos variables $f(x,y)$, si esta fuera continua el gradiente se define como.

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Y el modulo como.

$$G_{mag}(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}(x, y) \right)^2 + \left(\frac{\partial f}{\partial y}(x, y) \right)^2}$$

De esta manera es probable que un píxel pertenezca a un borde si el modulo del gradiente es lo suficientemente grande. Por lo regular se compara con un valor determinado con anterioridad y si este es mayor se considera que el píxel pertenece a un borde, la dirección del gradiente es.

$$G_{dir} = \arctg \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$$

Gradiente digital

Para la aproximación del gradiente de una imagen (una función discreta) se puede usar la formula.

$$\nabla_1 f(x, y) = (\nabla_{x1} f(x, y), \nabla_{y1} f(x, y))$$

Donde.

$$\nabla_{x1} f(x, y) = f(x, y) - f(x-1, y)$$

$$\nabla_{y1} f(x, y) = f(x, y) - f(x, y-1)$$

Tanto D_{x1} como D_{y1} pueden expresarse en forma de convolución utilizando las siguientes mascarar.

Figura 15. Dx y Dy como mascarar

0	0	0
0	1	-1
0	0	0

0	-1	0
0	1	0
0	0	0

Los operadores Sobel se usan para aproximar las derivadas de la imagen, por ejemplo para aproximar $D_x(i,j)$ se toma en consideración, el cambio de nivel de gris hacia la derecha de ese punto junto con el mismo cambio hacia la izquierda, además se promedia con los mismos valores calculados en la misma posición de las fila superior e inferior, para lograr una estimación de $D_x(x,y)$. Las mascarar para D_x y D_y son.

Figura 16. Los operadores sobel para Dx y Dy

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Como puede observarse en Dx se le da mas peso a la fila en la que se encuentra el píxel al cual se le esta calculando la derivada, mientras que en Dy el mayor peso esta en la columna del píxel del cual se quiere la derivada.

3.3.4 Laplaciano

El laplaciano de una función bidimensional es una derivada de segundo orden definida como.

$$\nabla^2 f = \left(\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2} \right)$$

Como en Sobel para una imagen digital se calcula por medio de la mascara mostrada en la figura 17.

Figura 17. Mascara para el Laplaciano de una imagen

0	-1	0
-1	4	-1
0	-1	0

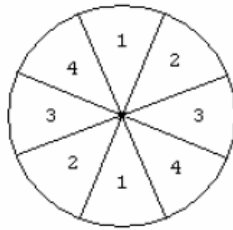
Como puede observarse la suma de los coeficiente es 0 es por eso que la respuesta es cero si el píxel del cual se desea averiguar el laplaciano tiene los mismos valores que los vecinos. El laplaciano de una imagen es sensible al ruido en la misma.

3.3.5 Canny

Se define a los bordes como las fronteras que separan regiones con diferente niveles de claridad o color, J. Canny sugirió un método efectivo para detectar bordes. Toma como entrada una imagen en escala de grises y retorna una imagen binarizada, donde los píxeles con el valor de 1 marcan los bordes detectados, a continuación la explicación de este algoritmo de 4 etapas

- Etapa 1. Suavización de la imagen
La imagen es suavizada mediante una función Gaussiana de un ancho especificado por un parámetro del usuario
- Etapa 2. Diferenciación
La imagen suavizada, de la Etapa 1 es diferenciada con respecto a las direcciones X e Y. De los valores de gradiente computados X e Y. La magnitud del ángulo del gradiente puede ser calculado usando la hipotenusa y el arco tangente.
- Etapa 3. Supresión de no máximos
Después que el gradiente ha sido calculado para cada uno de los píxeles de la imagen, los bordes pueden ser localizados en los puntos donde la magnitud sea la de un gradiente máximo local. Esto es hecho con la supresión de los píxeles con valores no máximos, en otras palabras, puntos para los cuales la magnitud del gradiente no es un máximo local. Sin embargo en los casos de no máximos que sean perpendiculares a la dirección del borde, en vez de los que están en dirección del borde, tienen que ser suprimidos, ya que la fuerza del borde se espera que continúe a lo largo del contorno.

Figura 18. Sectores de gradiente



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

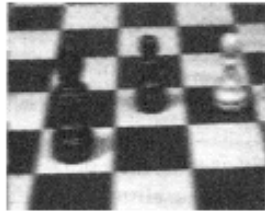
El algoritmo comienza reduciendo el ángulo del gradiente a uno de los cuatro sectores mostrados en la figura 18, el algoritmo pasa un vecindario de 3x3 a través del arreglo de magnitudes.

Cada píxel en el centro del vecindario, es comparado con sus dos vecinos a lo largo de la línea del gradiente dado el valor del sector. Si el valor central es un no máximo es decir no es más grande que sus vecinos este es suprimido

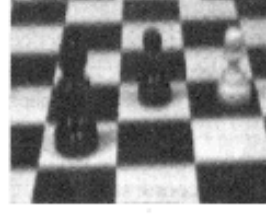
- Etapa 4. Limitación de bordes

El Algoritmo de Canny utiliza una así llamada limitación Histerestica, la mayoría del algoritmos utiliza un solo valor de limite, esto significa que si el valor del borde fluctúa arriba o debajo de este valor la línea aparece como cortada, esto se le conoce como **streaking**. El algoritmo combate este fenómeno definiendo un valor límite superior e inferior para el borde. Considerando un segmento de línea si un valor se encuentra arriba del limite superior es inmediatamente aceptada, si el valor se encuentra debajo del nivel inferior es rechazada, y los puntos que están dentro de los limites son aceptados si están conectados a píxeles que presentan una fuerte respuesta. De esta manera la posibilidad de **streaking** es reducida. J. Canny recomienda un radio de 3 entre el limite superior e inferior. La figura 19 presenta los resultados intermedio y final del algoritmo de Canny.

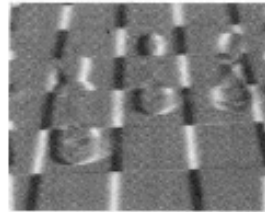
Figura 19. Pasos del algoritmo de J. Canny



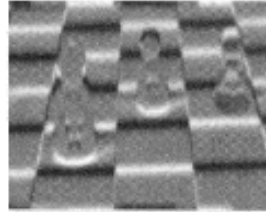
Componente X de la convolución gaussiana



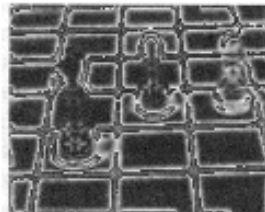
Componente Y de la convolución gaussiana



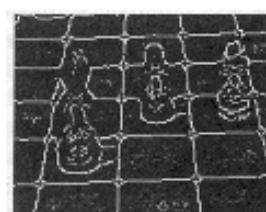
Componente X de la convolución de la derivada gaussiana



Componente Y de la convolución de la derivada gaussiana



Magnitud de la derivada gaussiana



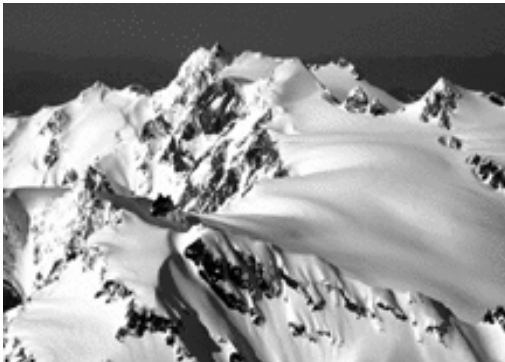
Supresión de los elementos no máximos

Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

A pesar de que este algoritmo es bastante mas complejo que los demás también es el que da mejores resultados, es el menos sensible a ruidos, debido en parte por que no utiliza un valor predefinido para calcular los bordes, y que por medio de la limitación histerestica se reduce el streaking (líneas cortadas) de la imagen.

Figura 20. Comparación entre los diferentes algoritmos

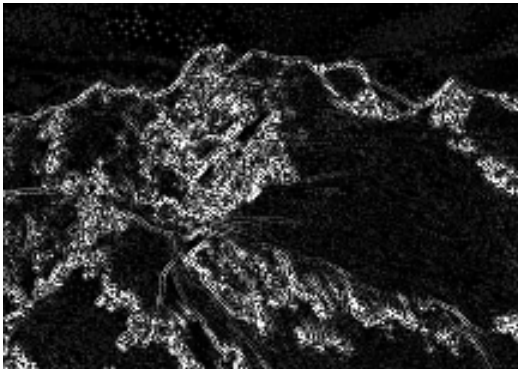
Imagen original



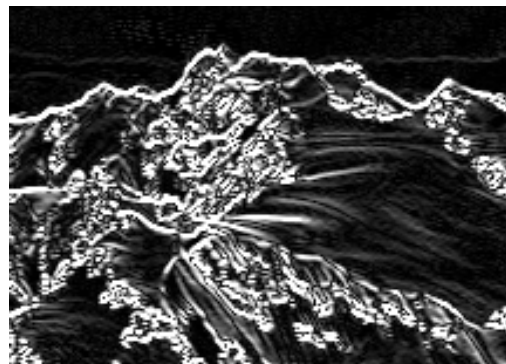
Canny



Laplace



Sobel



Umbral Adaptable



Umbral



Fuente: Imagen original, <http://walpapers.com>, enero 2004.

Como puede observarse en las imágenes anteriores de los 5 algoritmos, el que mejor encuentra los bordes de la imagen es el de Canny.

3.4 Procesamiento de contornos

Después de que todos los bordes han sido recuperados de la imagen, la representación de esta puede comprimirse aun más. La aproximación poligonal es el mejor método en términos de simplicidad de salida para procesamientos posteriores. La idea principal es encontrar y conservar los puntos dominantes, es decir puntos en donde están localizados los máximos locales de curvatura en la curva digital, almacenada en la cadena de la curva o en algún otro forma de representación. En el caso continuo la curvatura es determinada como la velocidad de cambio del ángulo tangente

$$k = \frac{(x'y'' - x''y')}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

En el caso discreto una aproximación simple llamada curvatura L1, es la diferencia entre sucesivas cadenas de la curva.

$$C_i^{(1)} = [(f_i - f_{i-1}) \bmod 8] - 4$$

Este método cubre cambios desde 0 que es la línea recta hasta 4 que corresponde al ángulo mas afilado que es cuando, la dirección es cambiada en sentido contrario. El siguiente algoritmo es usado para conseguir una aproximación más compleja el punto (x_1, y_1) de radio m_1 del vecindario a considerar es seleccionado. Para algunos métodos el valor m_1 es un parámetro del método y es constante para todos los puntos, para otros es calculado automáticamente para cada punto.

Los siguientes valores son calculados para todos los pares (x_{i-k}, y_{i-k}) ($k=1\dots m$).

$$C_{ik} = \frac{(a_{ik} \cdot b_{ik})}{(|a_{ik}| |b_{ik}|)} = \cos(a_{ik}, b_{ik})$$

donde $a_{ik} = (x_{i-k} - x_i, y_{i-k} - y_i)$, $b_{ik} = (x_{i+k} - x_i, y_{i+k} - y_i)$. El siguiente paso es encontrar un índice h_1 tal que $C_{im} < C_{im-1} < \dots < C_{ih} \geq C_{ih-1}$. el valor C_{ih_1} es estimado como el valor de la curvatura del i -esimo punto. Los valores del punto cambian de -1 (línea recta) a 1 (el ángulo mas agudo). Esta aproximación es llamada curvatura del k coseno.

El algoritmo Rosenfeld-Johnston es uno de los primeros algoritmos para determinar puntos dominantes en curvas digitales, el algoritmo requiere el parámetro m , el radio del vecindario que es a menudo $1/10$ o $1/15$ del número de puntos de la curva digital. El algoritmo es usado para calcular los valores de la curva para cada punto y remover los puntos que satisfacen la condición

$$\exists j, |i - j| \leq h_i / 2 : C_{ih_1} < C_{ih_j}$$

Los puntos que quedan son tratados como puntos dominantes. En la figura 21 se muestran de izquierda derecha, la Imagen de entrada y salida del algoritmo Rosenfeld-Johnston

Figura 21. Salida del algoritmo Rosenfeld-Johnston



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

La desventaja del algoritmo es la necesidad de encontrar el parámetro m y el hecho que este es el mismo para todos los puntos lo que da como resultado ya sea una aproximación demasiado burda o demasiado precisa de los bordes. Otro algoritmo es el propuesto por Teh y Chin incluye un método para una selección automática del parámetro m para cada punto. El algoritmo hace varias pasadas a través de la curva y borra algunos puntos en cada pasada. Al principio todos los puntos con un valor de cero en $c_i^{(1)}$ son borrados. Para los otros puntos el parámetro m_i y el valor de curvatura son determinados, después que el algoritmo elimina los no máximos, igual que en el algoritmo Rosenfeld-Johnston borrando puntos cuyas curvaturas satisfacen la condición previa donde para $c_i^{(1)}$ la medida h_i es puesta a m_i .

Finalmente el algoritmo reemplaza grupo de dos puntos restantes sucesivos con un solo punto y grupos de tres o más puntos sucesivos con el primer y ultimo punto. Este algoritmo no requiere más parámetros que la curva a usar. La figura 22 muestra a la derecha la imagen original y a la derecha el resultado de aplicar el algoritmo Teh-Chin.

Figura 22. De izquierda a derecha Imagen de entrada y salida para el algoritmo Teh-Chin



Fuente: *Intel Corporation, Open Source Computer Vision Library Reference Manual*, junio 2004.

3.5 Características

El objetivo del proyecto es clasificar una imagen dentro de un conjunto de imágenes de glifos, o dado el caso agregar la imagen del glifo a la colección para posteriores búsquedas. Una imagen en computadora puede equipararse a una matriz de valores llamados píxeles, por lo que una aproximación obvia a la solución del problema de clasificación, es el comparar los valores de los píxeles correspondientes de la imagen contra los píxeles de las imágenes que sirven de referencia. Este método no es el que brinda mejores resultados, tomando como ejemplo las dos imágenes que se muestran en la figura 23, en la cual se pueden observar dos cuadros de distinto tamaño, la mayoría de personas no tiene problemas para identificar a estas dos imágenes como equivalentes, pero usando este tipo de aproximación existe muy poca correlación entre los píxeles de las dos imágenes.

Figura 23. Ejemplo de figuras equivalentes



Otra aproximación más efectiva hace uso de lo que se denomina como un vector de características, un vector de características es simplemente un conjunto de características de la imagen, entendiendo por característica, como algo de lo que se pueda obtener una medida. Ejemplos de características para una imagen son:

- Alto de la imagen.
- Ancho de la imagen.
- Numero de líneas.
- Numero de esquinas.
- Numero de píxeles.

Como puede notarse una característica por si sola, únicamente puede brindar una idea general de la imagen, sin embargo es posible encontrar que, una cantidad razonable de ellas brinden la información necesaria, para lograr una clasificación aceptablemente buena. Con el uso esta técnica la clasificación de glifos trabaja aproximadamente:

- Determinar el vector de características para el glifo a clasificar.
- Comparar este vector con los ya clasificados.
- Mostrar las imágenes con los vectores mas parecidos.

Las características que actualmente usa el programa son las siguientes

- El numero de regiones cerradas en la imagen.
- Los 7 momentos geométricos de tercer orden, o momentos hu.

3.5.1 Teoría de los momentos

Es una alternativa útil de representar la forma de los objetos, si se tiene un objeto que en la imagen esta representado por todos aquellos píxeles cuyo valor sea distinto a 0, es decir $f(x,y) > 0$ entonces se define como momento de orden p,q como:

$$m_{pq} = \iint x^p y^q f(x, y) dx dy \quad \text{para } p, q = 0, 1, 2, \dots$$

El teorema de representación de los momentos indica que el conjunto infinito de momentos $m_{p,q}$ (para $p,q=0,1,2,\dots$) determinan de manera única a la función $f(x,y)$ y viceversa, una demostración de esto va mas allá de los alcances de este documento. Al tratarse de imágenes digitales el momento de orden $(p+q)$ se define ahora como:

$$M(p,q) = \sum_x \sum_y x^p y^q f(x,y)$$

Siendo de utilidad las funciones que podemos extraer de ellos, ya que son invariantes a las transformaciones geométricas como rotación, traslado y rotación.

3.5.2 Momentos de Orden 0.

Momentos Simples de Orden 0. $M(0,0)$: El momento simple de orden 0 representa el área de la figura en imágenes binarias y la superficie en imágenes en escala de grises. Es la suma de los valores de todos los píxeles. Para ello nos basamos en la fórmula de los momentos simples.

$$M(0,0) = \sum_x \sum_y f(x,y)$$

3.5.3 Momentos de Orden 1

Momentos Simples de Orden 1, $M(1,0)$, $M(0,1)$: Se usan principalmente para hallar el centro de masas de una figura.

$$M(1,0) = \sum_x \sum_y xf(x,y) \quad M(0,1) = \sum_x \sum_y yf(x,y)$$

Momentos Centrales de Orden 1, $MC(1,0)$, $MC(0,1)$: Estos momentos son 0 por definición.

$$U(1,0) = \sum_x \sum_y (x - \bar{x})^1 (y - \bar{y})^0 f(x,y) = M(1,0) - \frac{M(1,0)}{M(0,0)} M(0,0)$$

$$U(0,1) = \sum_x \sum_y (x - \bar{x})^0 (y - \bar{y})^1 f(x, y) = M(0,1) - \frac{M(0,1)}{M(0,0)} M(0,0)$$

Momentos Centrales Normalizados de Orden 1, MCN(1,0), MCN(0,1): Estos momentos son 0 por definición.

3.5.4 Momentos de Orden 2

Aquí es donde comienza el análisis de las imágenes mediante el reconocimiento de formas, estos momentos son de vital importancia en el cálculo los momentos centrales. La densidad de la figura se multiplica por distancias al cuadrado desde el centro de masas o centroide (Inercia).

$$U(p,q) = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

Momentos Centrales de Orden 2.

MC(2,0): Su valor aumentará cuanto mayor sea la componente horizontal de una figura.

MC(0,2): Su valor aumentará cuando mayor sea la componente vertical de una figura.

MC(1,1): Usa las componentes Horizontal y Vertical. Puede ser positivo o negativo dependiendo de donde se encuentre la componente vertical; si la componente vertical se encuentra en los cuadrantes 2º y 4º, entonces será negativo, si por el contrario está en los cuadrantes 1º y 3º entonces será positivo. Teniendo en cuenta esto es fácil deducir que una imagen simétrica respecto a los ejes, el MC(1,1) será 0.

Matriz de Rotación: Los momentos centrales forman las componentes de la matriz de rotación, partiendo de estas componentes se obtendrán tanto el Angulo de Rotación como la Excentricidad.

$$\begin{vmatrix} MC(0,2) & -MC(1,1) \\ -MC(1,1) & MC(2,0) \end{vmatrix}$$

Angulo de Rotación: Se define como el ángulo entre el eje de abscisas y el eje alrededor del cual la figura puede rotar con mínima inercia.

Excentricidad: Si definiéramos una elipse que rodease la figura para que quedase fuera un área de la figura igual al área que quedase dentro de una elipse, y teniendo en cuenta que una elipse tiene 2 focos, la excentricidad se definiría como la distancia entre los dos focos. La excentricidad medida sobre figuras normalizadas se encuentra en un rango entre 0 y 1. Por ejemplo para un cuadrado la excentricidad resulta 0, lo mismo que para un círculo ya que la elipse coincide con el círculo, en cambio para una recta infinita, la excentricidad sería infinita y si estuviera normalizada entonces sería 1.

3.5.6 Momentos de Orden 3

Estos momentos sirven para calcular los momentos invariantes. A partir de los momentos centrales normalizados de orden 2 y 3 se obtienen los siete momentos invariantes. Estos momentos, como su propio nombre indica, se mantienen invariantes ante rotaciones, traslaciones y cambios de escalas de una figura concreta

$$I_2 = (N(2,0) - N(0,2))^2 + 4(N(1,1))^2$$

$$I_1 = N(2,0) + N(0,2)$$

$$I_3 = (N(3,0) - 3N(1,2))^2 + (3N(2,1) - N(0,3))^2$$

$$I_4 = (N(3,0) + N(1,2))^2 + (N(2,1) + N(0,3))^2$$

$$I_5 = (N(3,0) - 3N(1,2))(N(3,0) + N(1,2)) \left[(N(3,0) + N(1,2))^2 - 3(N(2,1) + N(0,3))^2 \right] \\ + (3N(2,1) - N(0,3))(N(2,1) + N(0,3)) \left[3(N(3,0) + N(1,2))^2 - (N(2,1) + N(0,3))^2 \right]$$

$$I_6 = (N(2,0) - N(0,2)) \left[(N(0,3) + N(1,2))^2 - (N(2,1) + N(0,3))^2 \right] \\ + 4N(1,1)(N(3,0) + N(1,2))(N(2,1) + N(0,3))$$

$$I_7 = (3N(2,1) - N(0,3))(N(3,0) + N(1,2)) \left[(N(3,0) + N(1,2))^2 - 3(N(2,1) + N(0,3))^2 \right] \\ + (3N(1,2) - N(3,0))(N(2,1) + N(0,3)) \left[3(N(3,0) + N(1,2))^2 - (N(2,1) + N(0,3))^2 \right]$$

3.6 Clasificador

Todos los pasos anteriores, carga de imagen extracción de bordes, simplificación de contornos, dieron como resultado una representación compacta pero aceptable de la imagen que es sobre la cual se extraen las características anteriormente mencionadas, para la clasificación cada característica es tratada por separado.

Para evitar el tener que calcular estas características para todas las imágenes cada vez que se desee usar el clasificador, se guardan estos valores en un archivo, así solo se tienen que procesar para la nueva imagen, a la colección de imágenes ya procesadas, a este conjunto de características de imágenes le llamaremos la librería. A pesar de que el programa no pretende la traducción de textos mayas también se guarda alguna información epigráfica del glifo, los datos a guardar son:

- El numero-g (definido en el capítulo 1)
- Tipo de glifo
- Sonido
- Significado
- Imagen original

Esta última información es solamente para la interfaz de usuario ya que el clasificador solamente trabaja únicamente con el resumen de las características. Para cada característica el clasificador empieza midiendo la característica de la imagen del glifo a clasificar, luego hace una comparación con los valores de los glifos que existen en la librería, el objetivo de esta comparación es la asignación de un puntaje, entre mas cercanas sean las medidas mayor será el puntaje.

Este puntaje se suma a los puntajes de las características anteriores, al finalizar los glifos con los puntajes mas altos deben de ser los mas parecidos. Hay que tomar en cuenta que no todas las características describen igualmente bien a la imagen, al menos no necesariamente, por eso es posible definir características que puedan aportar una mayor cantidad posible de puntos al puntaje final.

3.7 Asignación de puntaje:

Para comparar a los valores se calculara la diferencia entre ellos, si Z es el valor de la característica en la nueva imagen y X es el valor de la librería, entonces la diferencia D la calcularemos de la siguiente manera

$$D= Z-X \text{ (si } Z>X) \text{ o}$$

$$D= X-Z \text{ (si } X>Z).$$

Aquí tendremos que entre mas cerca de 0 se encuentra D, mas parecida es esa característica, si solamente se tuviera una característica, este calculo seria suficiente para encontrar la imagen mas parecida, pero se están trabajando con varias características (que en conjunto deben de dar una buena descripción de la imagen), por lo que se necesita es crear un puntaje con una valor mínimo y un máximo para cada característica, entonces se define como rango (R), a la diferencia entre el valor máximo y el mínimo del conjunto de valores de una característica especifica.

$$R= \text{máximo}-\text{mínimo}.$$

El siguiente paso es al Rango restarle la Distancia, lo que da como resultado un numero entre 0 y R (el rango), luego se divide el resultado entre el rango, lo que dará como resultado un numero entre 0 y 1. Ya que la finalidad es el que entre mas parecidas sean los valores mas alto sea el puntaje, el puntaje es calculado como.

$$P=1-D/R$$

Esta formula da como resultado un numero entre 0 y 1, si los valores de Z y X son iguales el resultado es 1 (el puntaje mas alto), y si la diferencia entre Z y X es igual al rango (la mayor diferencia posible, con los datos que existen en la librería) el resultado es 0 (el puntaje mas bajo). Ya que existe la posibilidad de que una característica sea un mejor descriptor de la imagen que otra, se añade otra variable a este calculo que denominaremos como el peso. Cuando este valor es alto estamos indicando que la característica un buen descriptor de la imagen (Por defecto en el proyecto todas las características tienen el mismo peso). Y cuando el valor del peso es bajo estamos indicando que existen otras características más importantes al momento de tomar la decisión. Por lo que la formula del puntaje de una característica es:

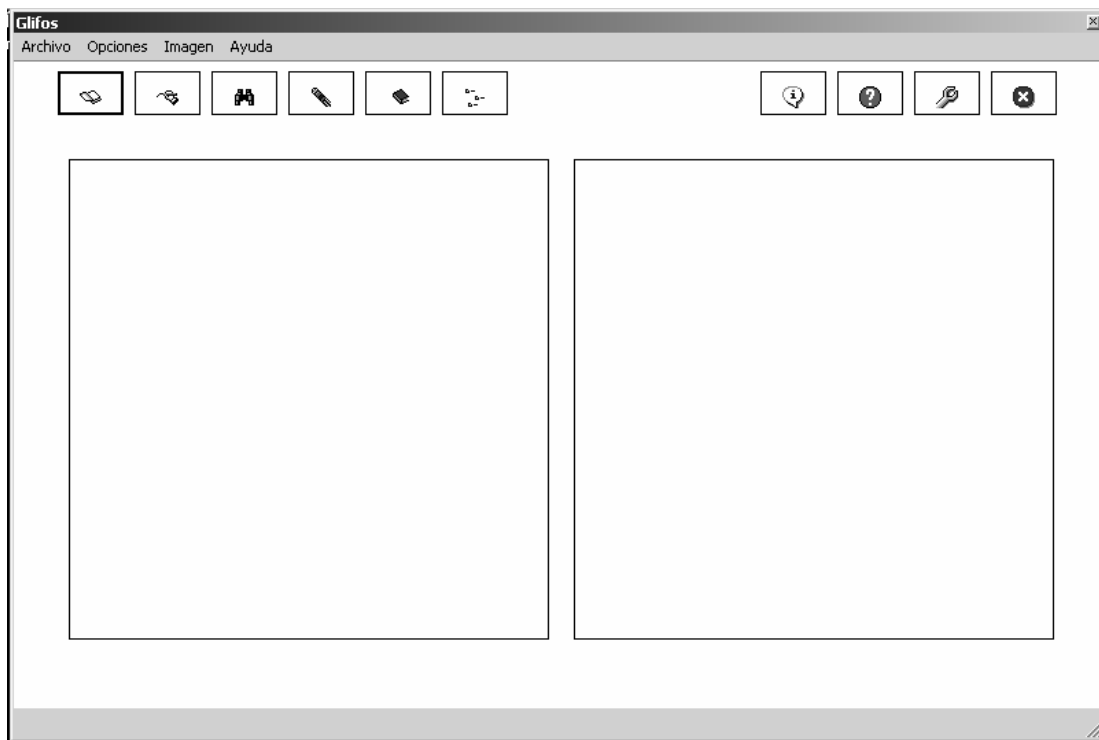
$$P = (1 - D/R) * \text{peso}.$$

El puntaje final es la suma de todos los puntajes individuales. Al finalizar el clasificador, cada glifo en la librería tiene un puntaje, se le muestran al usuario los glifos con los puntajes mas altos, si el usuario lo desea puede ir viendo los demás glifos en orden descendente, y decidir si el glifo que quiere identificar esta o no en la librería, si no lo esta puede agregarlo, lo que significa que la información del glifo será guardada para posteriores comparaciones, si el usuario conoce la información también podrá ingresar, el numero-g, tipo , traducción y sonido del glifo.

4. MANUAL DE USUARIO

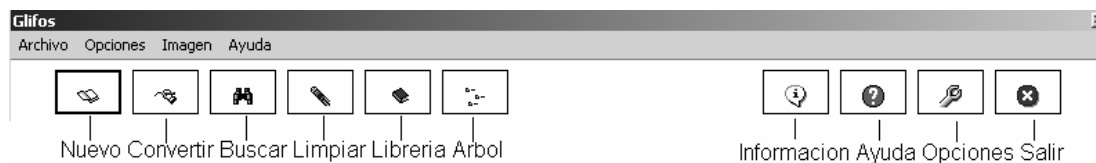
A continuación se explica como se utiliza el programa para el reconocimiento de glifos. Primero se cubre paso a paso el proceso para la clasificación de una imagen. La figura 23 es mostrada cuando el programa da inicio.

Figura 23. Pantalla principal del programa



En la figura 23 se puede notar que la pantalla consta de dos cuadros para imágenes, el de la izquierda es para las imágenes que se desee clasificar, el de la derecha para los resultados de extraer los contornos de las imágenes. En la parte superior un conjunto de botones que realizaran ciertas operaciones o nos mostraran otra ventana. La figura 24 nos muestra estos botones.

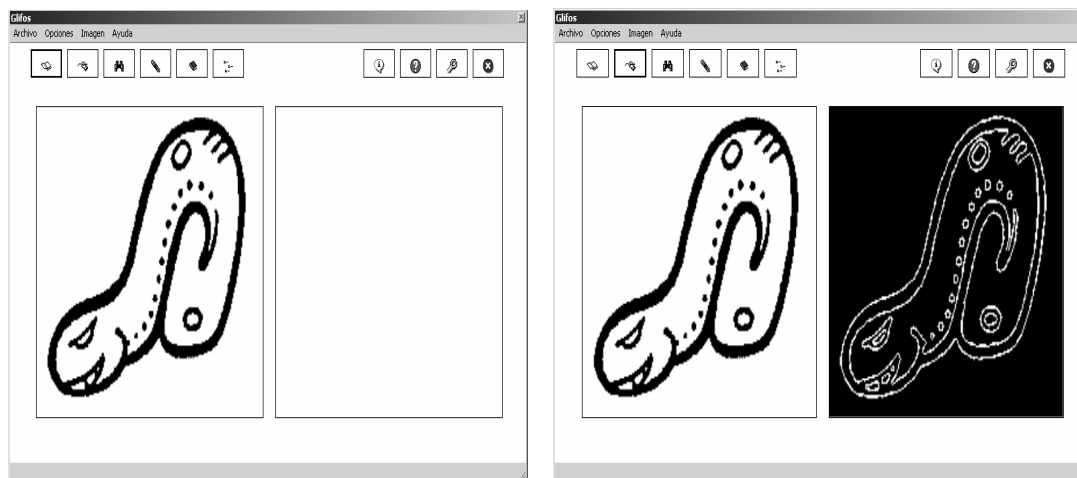
Figura 24. Botones de la pantalla principal



4.1 Clasificar una imagen.

Para iniciar el proceso de selección se da clic el botón Nuevo. Aquí se muestra un dialogo para que se seleccione la imagen que se desea clasificar. El programa soporta únicamente archivos del tipo bmp, jpg o png. Si se tiene archivos que no son de este tipo, la mayoría de programas de edición de imágenes permiten cambiar de un formato a otro, si este es el caso lo mejor es cambiar sus imágenes a formato bmp, ya que este formato no pierde información. Cuando se haya seleccionado el archivo, este se muestra en el primer cuadro, en el formato original de la imagen. Una vez terminado este paso, se esta preparado para el segundo paso que es la extracción de contornos, que se realiza dando clic en el botón Convertir.

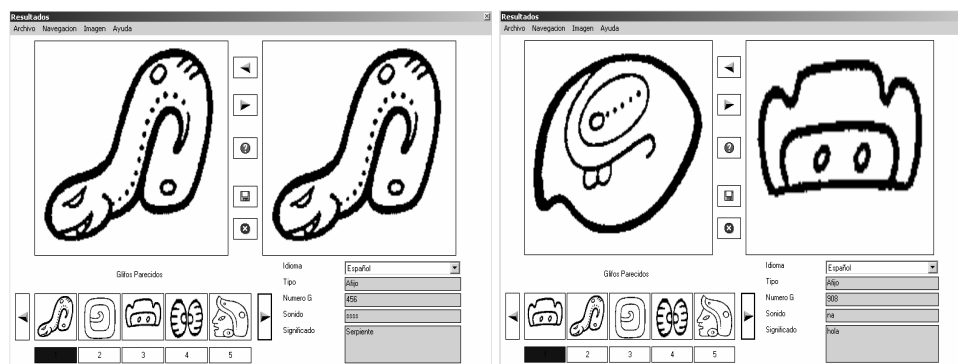
Figura 25. Imagen original y los contornos extraídos



En la figura 25 se puede ver un ejemplo del resultado de los pasos anteriores, si se da clic con el botón derecho del ratón en alguna de las imágenes se mostrara un menú el cual permite ver la imagen en cualquiera de las tres formas siguientes. *Normal* la imagen se muestra en su tamaño real. *Automático* aquí la imagen se acomoda al tamaño del cuadro y *Centrado* igual que en *Normal* solo que centra la imagen en el cuadro. El siguiente paso es la búsqueda de esta imagen dentro de la librería para esto solo se tiene que dar clic en el botón Clasificar, Cuando esto ocurre los siguiente pasos tendrán lugar internamente.

Primero de la imagen binarizada se extraerán los contornos, con estos contornos se procede a medir las características, luego de que se han obtenido todas las características, comienza el proceso de comparar estas características con las características de las imágenes ya en la librería, con el propósito de asignar una puntuación, cuando ya se ha comparado con todas las imágenes de la librería, se procede a ordenarlas del punteo mas alto al mas bajo para de ultimo ser mostradas en ese orden en la pantalla que llamaremos pantalla de Resultados. La figura 26 muestra dos posibles resultados de estas operaciones, el primero que la imagen se encuentre en la librería, y la segunda opción que la imagen no se encuentre.

Figura 26. Resultados de la clasificación



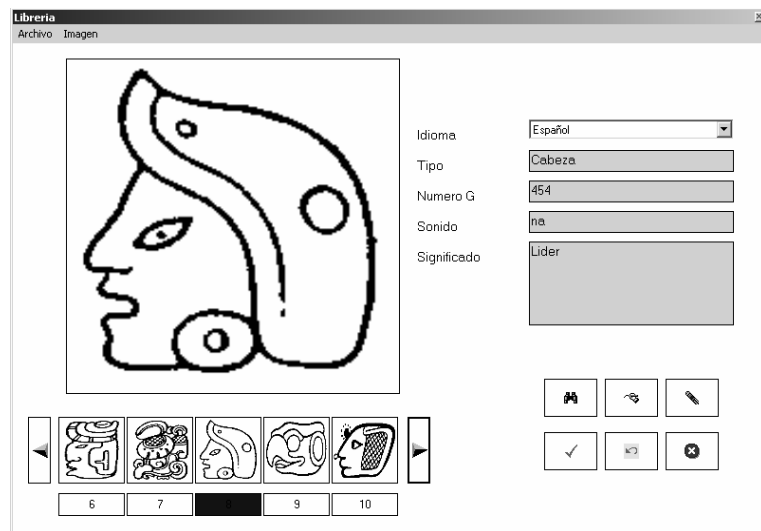
Existen varias opciones aquí, la imagen de la izquierda es la imagen del glifo que se está buscando. Las pequeñas imágenes del fondo pertenecen a la librería y se muestran en orden de puntuación. La imagen de la derecha es una versión aumentada de alguna de las imágenes del fondo. Los datos que se encuentran debajo de la imagen a la derecha corresponden a la imagen de la librería seleccionada. Al principio el programa presenta la imagen con la puntuación más alta, lo que no necesariamente significa que esa sea el glifo buscado. Puede darse el caso de que la imagen no se encuentra en la librería, o que por motivos de, diferentes algoritmos de binarización, ruido, escala etc.. la imagen buscada no sea la de la puntuación más alta.

En vista de esto se puede navegar hacia todas las imágenes de la librería para que el usuario pueda estar seguro de que no se encuentra puede revisar todas las imágenes, en teoría esto no es necesario ya que si el glifo se encuentra este debería de estar entre las primeras imágenes. Para ver las imágenes solo hay que dar clic en la imagen pequeña del fondo. Para moverse hacia delante solo hay que dar clic en el botón de la derecha y para moverse hacia atrás en el botón de la izquierda.

4.2 La librería

Para acceder a la librería solo hay que dar clic en el botón librería de la página principal, lo cual nos mostrara la pantalla de la librería (figura 27).

Figura 27. La librería de glifos



En esta pantalla se muestra información de los glifos ya almacenados, la información que presenta es: el tipo de glifo, su numero g, sonido y significado. Aquí es posible navegar en la librería, hacer búsquedas, editar y eliminar glifos de la misma.

4.2.1 Navegar en la librería

Como puede observarse en la figura 27 en la pantalla existen un conjunto de 5 imágenes pequeñas debajo de la imagen grande, estas son una vista previa de glifos que existen en la librería, por defecto están ordenadas por tipo de glifo, para acceder a un glifo de ese conjunto de imágenes solo se da un clic sobre la vista previa, entonces se mostrara esa glifo, y los datos (de estar disponibles) del mismo. Para mostrar los siguientes 5 glifos solo hay que dar clic en el botón con una flecha a la derecha, y para mostrar los 5 glifos anteriores en el botón con una flecha a la izquierda. Es posible cambiar el idioma en que se muestran los datos, simplemente escogiendo el idioma en el combo que se encuentra a la par de la imagen del glifo.

4.2.2 Búsquedas en la librería

Es posible el realizar búsquedas en la librería para ello solamente tiene que presionar el botón buscar (el que tiene la imagen de un binocular), esto limpiara la información que se este presente y permitirá que se escriban los datos a buscar, tome en cuenta que se puede escribir en uno o mas de los campos disponibles, y de hacerlo en mas de un campo la búsqueda realizara un **y** lógico para unir las diferentes condiciones, es decir que se cumpla el valor en el campo 1 y en el campo 2 etc. Luego de que de escribir la o las condiciones se presiona el botón aceptar (el que tiene una imagen de un cheque) y se muestran los resultados de la búsqueda, como la búsqueda puede dar 0, 1 o mas resultados se puede navegar entre los resultados de la misma manera anteriormente definida.

4.2.3 Edición de glifos

En el momento que se visualizan los datos de un glifo es posible editar los mismos, nótese que solamente se pueden editar los campos de tipo, numero g, sonido y significado no así la imagen, para realizar esto simplemente presione el botón de editar el que tiene una imagen de una mano con un lápiz, esto permitirá la edición de los campos del glifo, para guardar los cambios presione el botón de aceptar, también es posible presionar el botón cancelar en este caso se perderán los cambios y los campos recuperaran los valores previos.

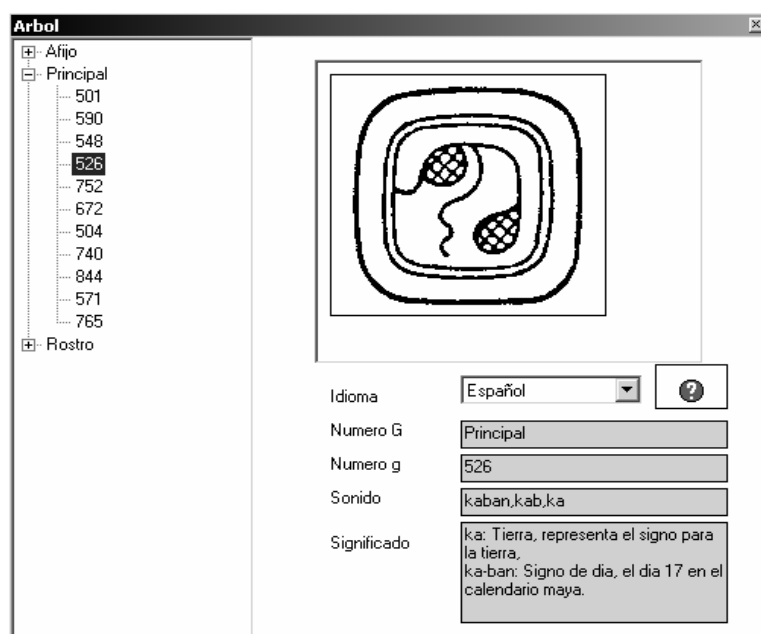
4.2.4 Eliminación de glifos

Si se tiene algún glifo seleccionado es posible eliminarlo de la librería, para realizar esta operación solo hay que presionar el botón de eliminar cuando este seleccionado el glifo que se desea eliminar, el glifo no se eliminara realmente a menos que después se presione el botón de aceptar tome en cuenta que en ese punto ya no se puede recuperar los datos, para cancelar la operación se presiona el botón cancelar.

4.2.5 Árbol de glifos

Una manera alterna de visualizar los glifos de la librería, es un árbol que se crea a partir de los tipos de glifo que estén definidos en la librería, aquí es posible navegar hacia cualquier glifo usando el árbol de la izquierda, y se muestra la imagen y los datos del glifo. Pero hay que tomar en cuenta que en esta pantalla solamente es de consulta, por lo que no es posible la edición, eliminación o búsqueda de glifos, para estas operaciones se puede utilizar la librería. En la figura 28 se muestra esta pantalla.

Figura 28. Árbol de glifos.



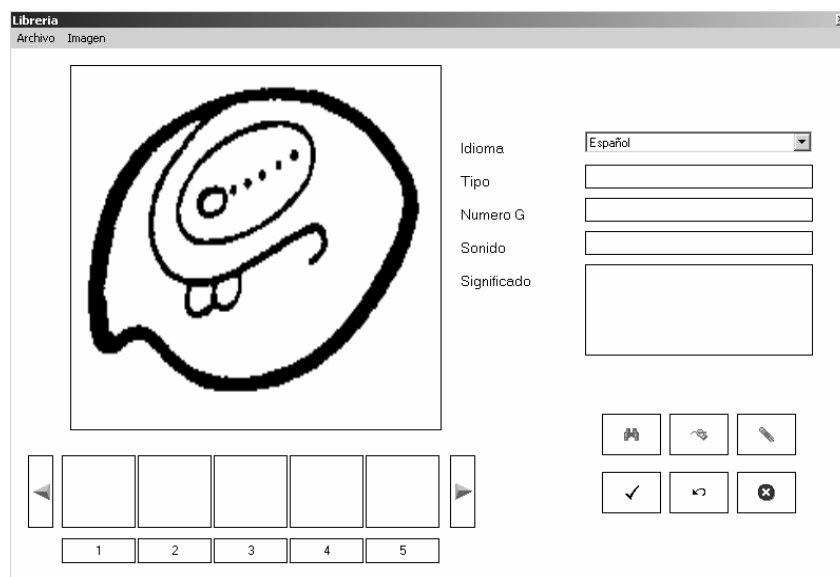
4.3 Añadir un glifo a la librería

Para añadir una imagen a la librería primero debe dar los pasos para clasificar una imagen, que en resumen son. Nueva imagen, Convertir a contornos, y Clasificar, luego de revisar que efectivamente no se encuentra en la librería dar clic en el botón añadir a la librería. Esto cerrara esta pantalla y mostrara la pantalla de la librería (figura 29) con el nuevo glifo en ella para que se ingrese la información del glifo, la información que puede almacenarse es:

- El tipo de glifo, los tipos son afijo, principal y rostro.
- Numero-g los números-g en resumen son:
 - 1-327 = Glifos afijos identificados
 - 328-499 = Glifos afijos recientemente identificados
 - 500-856 = Glifos principales identificados
 - 857-999 = Glifos principales recientemente identificados
 - 1000-1087 = Glifos de rostro identificados
 - 1088-1299 = Glifos de rostro recientemente identificados
 - Glifos no identificados = ?.
 - Glifos inidentificables = 999.
 - Glifos inferidos {numero-g inferido}.
- Sonido: si se dispone del sonido del glifo
- Significado: El significado del glifo.

En caso que de momento no sepa los datos solicitados, puede ponerse alguna información para poder reconocer el glifo para cuando tenga la información del mismo.

Figura 29. Nuevo glifo en la librería.



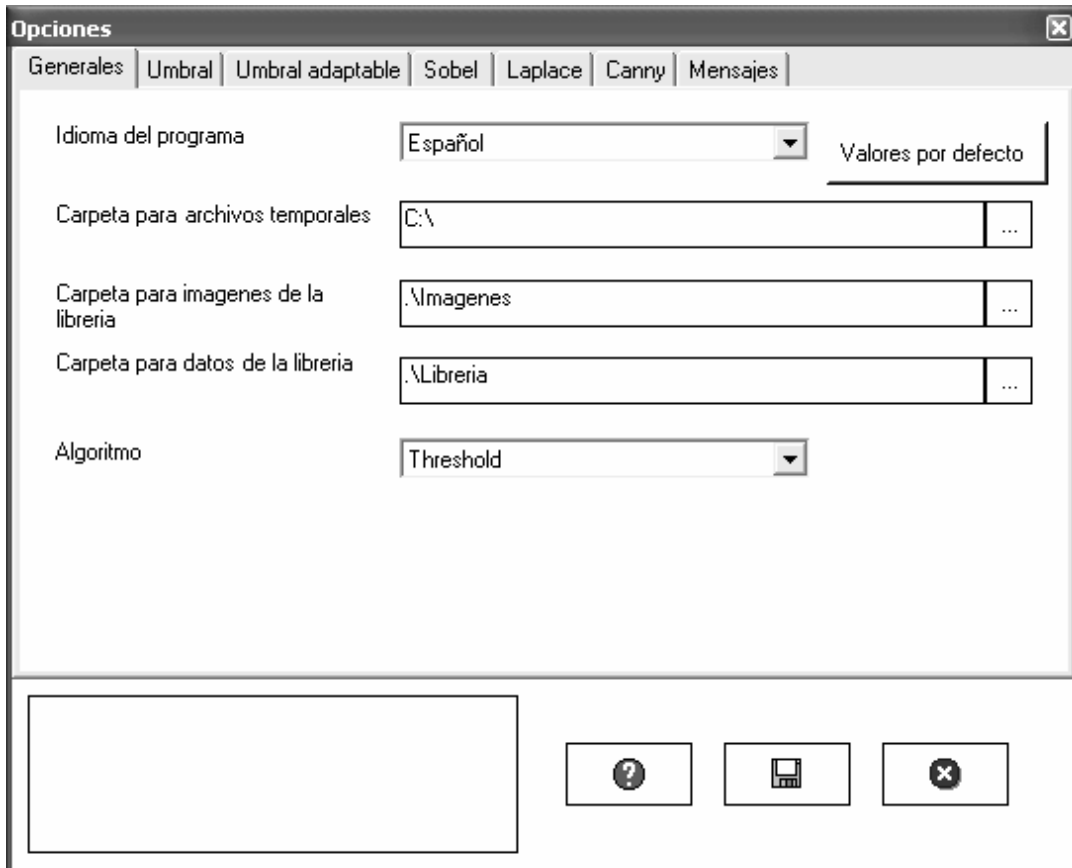
4.4 Opciones del programa

Para entrar a las opciones del programa solo hay que presionar el botón opciones de la pantalla principal a continuación se dará una breve explicación de las opciones principales, para las opciones de cada algoritmo puede consultarse la ayuda. Como puede observarse en la figura 4.8 la pantalla cuenta con un conjunto de pestañas cada una con un botón de valores por defecto, y un botón para guardar los cambios y otro de salir, el botón de valores por defecto pone de nuevo los valores iniciales en la pestaña activa, el botón de guardar salva los cambios y cierra la ventana y el de salir cerrara la ventana perdiendo cualquier cambio que se haya realizado.

4.4.1 Opciones generales

En la pestaña generales la primera opción del programa, es la del idioma, actualmente solo se puede elegir entre el Español y el Ingles, un cambio en esta opción, inmediatamente cambiara las etiquetas del programa y el idioma por defecto de la librería. El siguiente campo es la carpeta para archivos temporales tenga en mente que tiene que tener derechos de escritura y lectura en esta carpeta, el siguiente es la carpeta donde el programa guarda y busca los archivos de imagen de los glifos y la siguiente es la carpeta donde se guardan y leen los datos de la librería, en condiciones normales los valores por defecto no deben ser modificados especialmente el de los últimos dos.

Figura 30. Opciones del programa



La siguiente opción permite elegir el algoritmo de binarización o detección de bordes, el algoritmo mas eficiente es el de Canny y este es el valor por defecto, por lo que se recomienda usar este algoritmo.

4.4.2 Umbral

En este algoritmo el valor limite es usado para la toma de decisiones, valor máximo es usado con algunos de los distintos tipos de umbral, existen varios tipos de umbral a continuación una explicación de cómo funcionan cada uno de ellos.

CV_THRESH_BINARY: Si $\text{origen}(x,y) > \text{valor limite}$, $\text{imagen}(x,y)=\text{valor máximo}$, si no $\text{imagen}(x,y)=0$.

CV_THRESH_BINARY_INV: Si $\text{origen}(x,y) > \text{valor limite}$, $\text{imagen}(x,y)=0$, si no $\text{imagen}(x,y)=\text{valor máximo}$.

CV_THRESH_TRUNC: Si $\text{origen}(x,y) > \text{valor limite}$, $\text{imagen}(x,y)=\text{valor limite}$, si no $\text{imagen}(x,y) = \text{origen}(x,y)$.

CV_THRESH_TOZERO: Si $\text{origen}(x,y) > \text{valor limite}$, $\text{imagen}(x,y) = \text{origen}(x,y)$, si no $\text{imagen}(x,y)=0$.

CV_THRESH_TOZERO_INV: Si $\text{origen}(x,y) > \text{valor limite}$, $\text{imagen}(x,y) = 0$, si no $\text{imagen}(x,y) = \text{origen}(x,y)$.

4.4.3 Umbral adaptable

En este algoritmo los tipos de límite funcionan de la siguiente manera.

BINARY. Si $\text{origen}(x,y) > T(x,y)$, $\text{imagen}(x,y)=\text{valor máximo}$, si no $\text{imagen}(x,y)=0$.

BINARY_INV Si $\text{origen}(x,y) > T(x,y)$, $\text{imagen}(x,y)=0$, si no $\text{imagen}(x,y)=\text{valor máximo}$.

Donde $T(x,y)$ es calculado para cada píxel, dependiendo del método, los métodos son:

MEAN: $T(x,y)$ es la media de un vecindario de 3x3 píxeles menos el valor de parámetro.

GAUSSIAN: es una suma pesada (gaussiano) de un vecindario de 3x3 píxeles menos el valor de parámetro.

4.4.4 Sobel

En este algoritmo los siguientes parámetros son utilizados:

Dx: es el orden de la derivada en x

Dy: es el orden de la derivada en y,

Tamaño de apertura: tamaño de la mascara para el calculo de la derivada, los valores permitidos son 1, 3, 5 y 7.

4.4.5 Laplace

Para este algoritmo el parámetro tamaño de apertura es el tamaño de la mascara que se usara para calcular el laplaciano, los valores permitidos son 1, 3, 5 y 7.

4.4.5 Canny

En este algoritmo los parámetros son los siguientes:

Limite inferior: Es usado para conectar bordes

Limite superior: Es usado para encontrar segmentos iniciales de bordes fuertes.

Tamaño de apertura: el tamaño de la mascara para el operador sobel, los valores permitidos son el 3 y 5.

4.4.6 Mensajes

En esta pantalla se muestra todos los mensajes del programa en los dos idiomas disponibles, así que es posible cambiarlos simplemente con editar los mensajes y grabar los cambios, estos mensajes son mostrados en una matriz cuya primera columna es el idioma del mensaje, la segunda es un código que identifica al mensaje y la tercera es el mensaje que se muestra en el programa, por lo que es en esta columna la que deben realizarse los cambios, estos mensajes pueden ordenarse por cualquiera de las tres columnas.

4.5 Requerimientos del sistema

Mínimo

- Procesador Intel Pentium a 200 MHZ o equivalente
- 64 MB de RAM
- Sistema Operativo Windows 98, Windows NT, Windows 2000, Windows XP
- Microsoft .NET Framework 1.0

- 10 MB de espacio en disco
- Navegador Web (Internet Explorer 4.0 o Netscape 4.0)

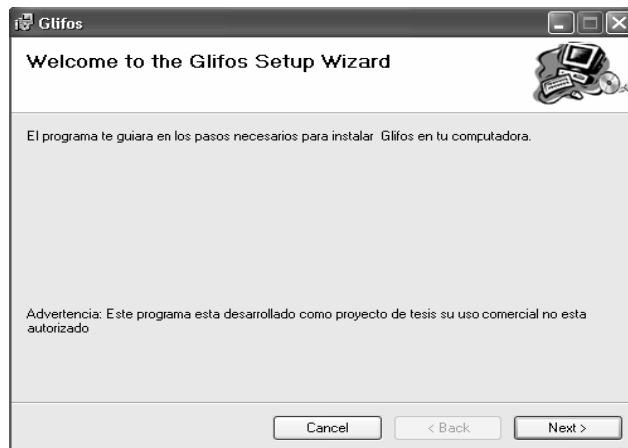
Recomendado

- Procesador Intel a 500 MHZ o superior
- 128 MB de RAM
- Sistema Operativo Windows XP o superior
- Microsoft .NET Framework 1.1
- 100 MB de espacio en disco
- Internet Explorer 6 o superior

4.6 Instalación

La instalación del programa es bastante simple, después de asegurarse que se cumplen los requisitos mínimos del sistema, verifique que cuenta con los archivos necesarios para la instalación estos son, Glifos.msi, setup.ini y setup.exe, se puede encontrar estos archivos en la siguiente dirección de Internet <http://www.geocities.com/oomcgt/>. Se ejecuta el archivo instalador setup.exe y se sigue las instrucciones que se van mostrando en la pantalla, el instalador se encarga de crear las carpetas, archivos, accesos directos, etc.

Figura 31. El programa instalador



CONCLUSIONES

1. Con el programa en actual estado, por lo menos es posible crear un catalogo de grifos mayas y guardar información de los mismos en por lo menos dos idiomas, y buscar tanto por el numero G. descripción, o por imagen, aunque es cierto que todavía no se puede trabajar con imágenes reales de estelas o pirámides.
2. Se demostró que al utilizar los momentos de orden 3 de una fusión de dos variables creada a partir de los contornos extraídos de una imagen de in glifo aislado, es posible crear un programa reconocedor de patrones.
3. Una cantidad considerable de glifos Mayas posee un contorno externo, el cual es, básicamente, un cuadro con los bordes redondeados, por lo que entre glifos con estas características, el clasificador tiende a dar puntajes altos a todos los glifos que tengan un contorno externo parecido. Da como resultado que, si el glifo ya se encuentra en la librería, este no sea, necesariamente el de puntaje más alto.
4. Cuando se trabaja con archivos jpg, a menos que esté grabado con una calidad muy alta (entre 95% 100%) se agrega ruido a la imagen y los algoritmos de extracción de contornos dan resultados muy distintos a los que dan cuando se usa un formato bmp o un jpg con calidad alta, por lo que el clasificador, también, evalúa este ruido, dando como resultado una pobre clasificación del glifo.

RECOMENDACIONES

1. Como se puede observar el proyecto ayuda a la clasificación de glifos individuales, el siguiente paso es la capacidad de separar y clasificar, ya sea manual o automáticamente, cada uno de los glifos que componen un texto, y luego implementar una gramática que permita al programa sugerir una o mas posibles traducciones al texto.
2. Otra área de trabajo es implementar otros algoritmos de extracción de contornos y comparar resultados, esto con el fin de mejorar la calidad del clasificador, o incluso hacer un análisis previo de la imagen y según las características de la misma, que el programa automáticamente sugiriera la mejor opción u opciones para cada tipo de imagen,
3. El programa entra en la categoría de un reconocedor estadístico de patrones, pero existen otras opciones para el desarrollo de un programa que aplique reconocimiento de patrones a glifos mayas, tales como redes neuronales o sistemas expertos.

BIBLIOGRAFÍA

Fundación para el avance de los estudios mesoamericanos FAMSI
<http://www.famsi.org/spanish/mayawriting/>

Inga E. Calvin, Maya Hieroglyphics Study Guide, 2004

Nancy McNelly, How to Write Your Name In Mayan Glyphs,
<http://www.halfmoon.org/names.html>

James B. Porter, MAYA GLYPHS, www.mayaglyphs.net/panel.html

OSTG, Inc. ("Open Source Technology Group"), Opencv,
<http://www.sourceforge.net/projects/opencvlibrary>.

Carnegie Mellon University, The computer Vision Homepage,
<http://www-2.cs.cmu.edu/~cil/vision.html>

Serge Belongie, Matching with Shape Contexts,
http://www.cs.berkeley.edu/projects/vision/shape/sc_digits.html

Intel Corporation, Open Source Computer Vision Library Reference Manual,
order number A77028-004

Raf Alvarado, The Mayan Epigraphic Database Project (MED),
<http://www.iath.virginia.edu/med/home.html>

FRIGUI HICHEM, Pattern Recognition, <http://prlab.ee.memphis.edu/frigui/>

APÉNDICES

OpenCV

OpenCV es un conjunto de funciones en C y algunas clases de C++ que implementan algunos algoritmos de procesamiento de imágenes y de visión por computadora. Es de uso gratuito tanto para uso comercial o no comercial.

Instalación de OpenCV

La instalación de OpenCV para Windows es bastante simple, Se bajan el archivo instalador de <http://www.sourceforge.net/projects/opencvlibrary>. Este archivo instala OpenCV, registra los filtros de DirectShow y realiza otros procedimientos post instalación. Esta instalación incluye archivos de ejemplo y ayuda.

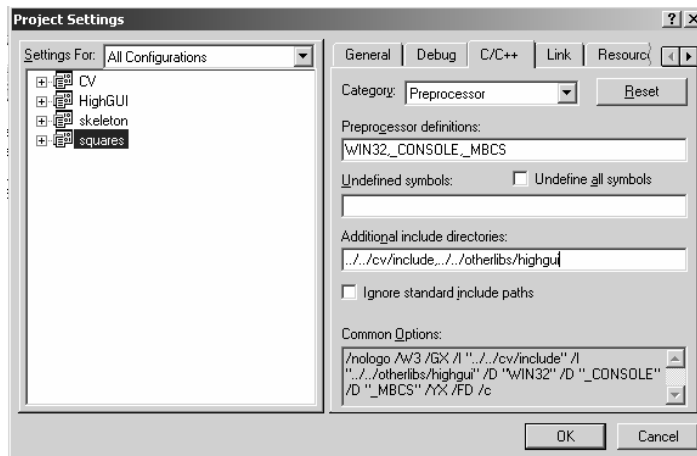
Para utilizar OpenCV en un programa se necesita algún compilador de C o C++, algunos por ejemplo:

- Microsoft Visual C++, 6.0 o mas alto. Esta es la opción mas sencilla, ya, que la mayoría de los ejemplos incluidos en la documentación están escritos en el.
- Intel Compiler 6.0 o mas alto
- Borland C++ 5.5, este computador es gratuito, o mas alto
- GNU C/C++ 2.95.3 o mas alto
- CH, interprete de C

A continuación se explican los pasos para usar OpenCV en un proyecto de Visual C++ se supone algo de experiencia en este compilador, los pasos para los demás compiladores o interpretes se encuentran en la documentación de OpenCV que están disponibles en la misma dirección que el instalador.

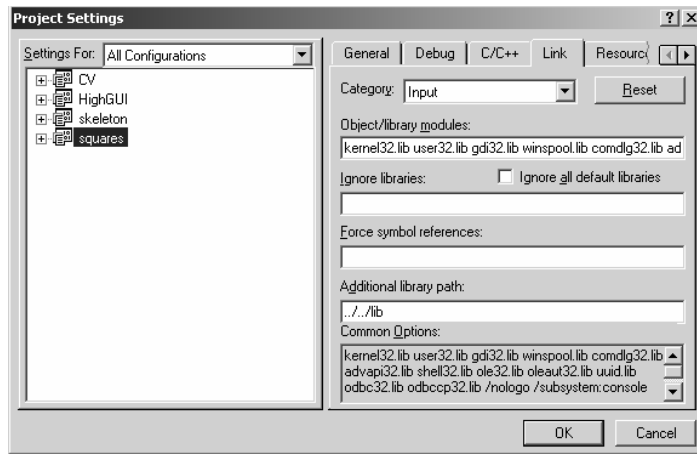
- Crear un proyecto nuevo del tipo **Win32 Application** o **Win32 console application**
- Añadir un archivo al proyecto
Añadir al archivo las cabeceras de OpenCV
#include "cv.h"
#include "highgui.h"
Otra opción es copiar un archivo existente al folder del proyecto y añadirlo al proyecto.
- Definir las propiedades del proyecto
Activar el dialogo de propiedades del proyecto seleccionando del menú **Project->Settings...**
Seleccionar del proyecto en el panel derecho y colocar las opciones comunes a las configuraciones **Release y Debug**
Seleccionar **Settings For:->All Configurations**
Elegir la pestaña de C/C++ en la categoría de **Preprocessor**
-> **Additional Include Directories** añadir separados por coma la dirección relativa (al archivo .dsp) o absoluta a **<opencv>\cv\include, <opencv>\otherlibs\highgui** y opcionalmente, **<opencv>\cvaux\include**.

Figura 32. Categoría Preprocessor



Escoger en la pestaña **Link** en la categoría **Input** y añadir las direcciones para todas las librerías necesarias

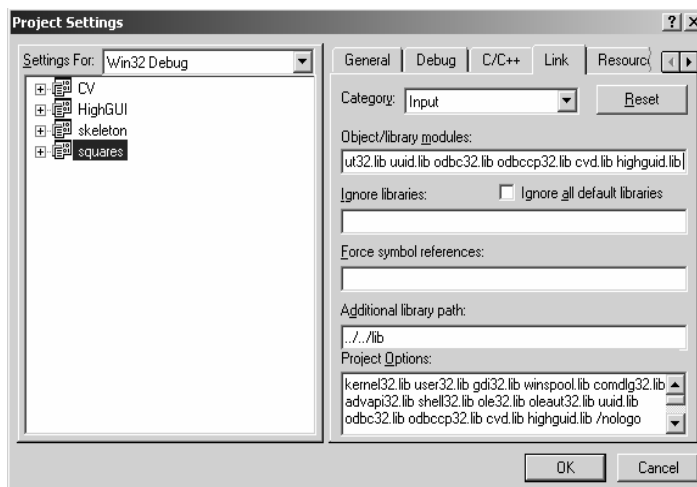
Figura 33. Categoría Input



Definir las propiedades para la configuración de **Debug**

- Seleccionar **Settings For:->Win32 Debug**
- En la pestaña **Link** en la categoría **Object/library modules** añadir separado por espacios **cvd.lib, highguid.lib, cvauxd.lib** (opcional)

Figura 34. Propiedades para Win32 Debug

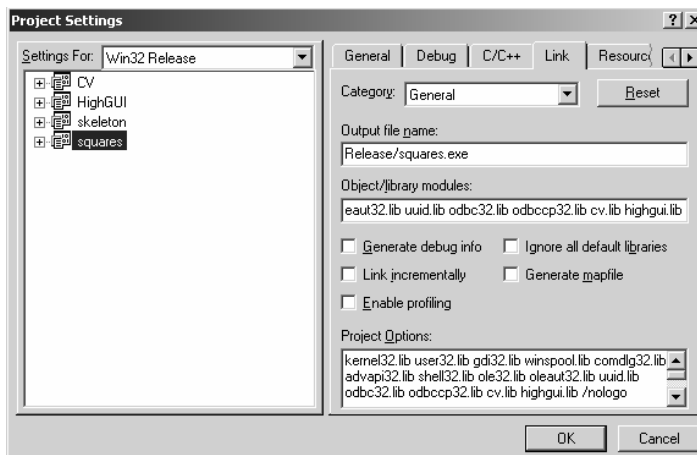


Definir las propiedades para la configuración de **Release**

Seleccionar **Settings For:->Win32 Debug**

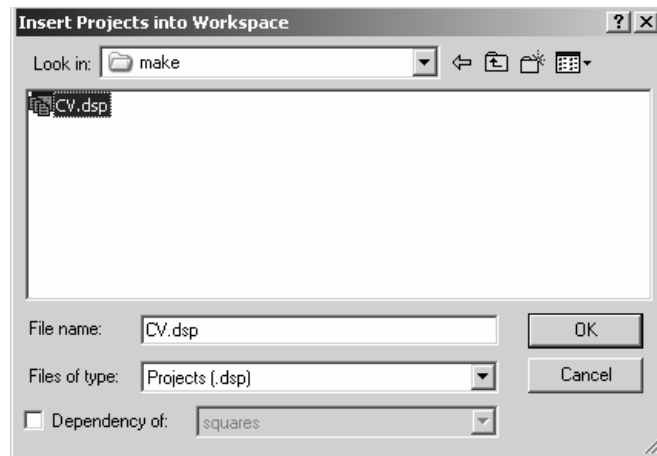
En la pestaña **Link** en la categoría **General** añadir separado por espacios **cvd.lib, highguid.lib, cvauxd.lib** (opcional)

Figura 35. Propiedades para Win32 Release



- Añadir las dependencias en el espacio de trabajo
Escoger del menú **Project -> Insert project into workspace**
Escoger **<opencv>\cv\make\cv.dsp**.

Figura 36. Insertar un proyecto.



Hacer lo mismo para `<opencv>\cvaux\make\cvaux.dsp`,
`<opencv>\otherlibs\highgui\highgui.dsp`.

Definir las dependencias

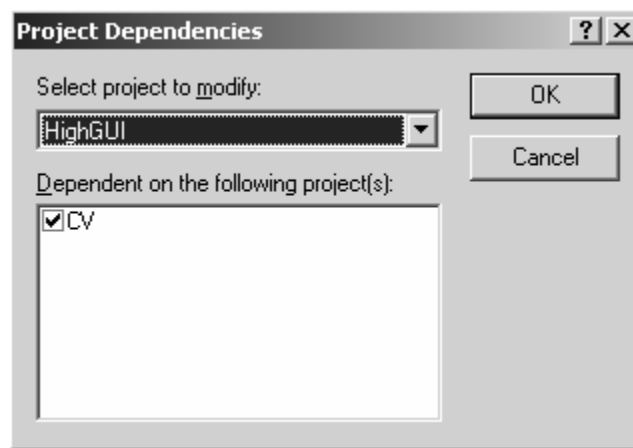
Escoger del menú **Project -> Dependencies...**

Para **cvaux** escoger **cv** (opcional)

Para **highgui** escoger **cv**

Para el proyecto escoger a todos **cv, cvaux, highgui**.

Figura 39. Definir dependencias



Ahora ya se tiene un proyecto con acceso a las funciones de OpenCv.