



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

RECONOCIMIENTO DE SIMBOLOS EN IMAGENES
DIGITALES DE LAS CULTURAS MESOAMERICANAS

P R O Y E C T O

QUE PARA OBTENER EL TITULO DE
LICENCIADA EN CIENCIAS DE LA
C O M P U T A C I O N
P R E S E N T A:
YAZMIN OCADIZ MARTINEZ



DIRECTOR DEL PROYECTO: DR. ERNESTO BRIBIESCA CORREA

ASESOR DEL PROYECTO: DR. HERMILO SANCHEZ CRUZ



2004 FACULTAD DE CIENCIAS
SECCION ESCOLAR



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

DRA. MARÍA DE LOURDES ESTEVA PERALTA
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"Reconocimiento de símbolos en imágenes digitales de las culturas mesoamericanas"

realizado por Ocaziz Martínez Yazmín

con número de cuenta 09412935-5 , quien cubrió los créditos de la carrera de:
Lic. en Ciencias de la Computación.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis Propietario	Dr. Ernesto Bribiesca Correa
Propietario	Dr. Hermilo Sánchez Cruz
Propietario	Dr. Yann Frauel
Suplente	Mat. Ana Luisa Sólis González Cosío
Suplente	M. en C. Miguel Miranda Miranda

E. Bribiesca

Hermilo Sánchez Cruz
Frauel

Mat. Ana Luisa Sólis González Cosío

M. en C. Miguel Miranda Miranda

Consejo Departamental de

Matemáticas



Mat. Ana Luisa Sólis González Cosío

FACULTAD DE CIENCIAS
CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

Reconocimiento de símbolos en imágenes digitales
de las culturas mesoamericanas

A mis papás con amor.

Al hombre que amo, Erick Celso Zavaleta González.

Agradecimientos

A mi mejor amigo, Jesús, por darme la oportunidad de vivir, por recordarme a cada instante que la vida vale la pena y por la enorme ayuda que siempre me ha brindado. Gracias Dios.

A mis papás, Francisca Lilia Martínez Martínez y Miguel Ocádiz García, por el gran amor que me tienen, por su confianza y por apoyarme en la realización de mis sueños.

A mi hermana, Jessica Lilia Ocádiz Martínez, por ser hermana, amiga, compañera, confidente y cómplice; por cuidarme, enseñarme, ayudarme y apoyarme en todo momento; y por todo el cariño que me tiene¹.

Al Dr. Octavio Quezada por invitarme a participar en el proyecto del cual se derivó este trabajo.

A Conacyt por el apoyo brindado al proyecto, del cual se me proporcionó una ayuda económica.

Al Dr Ernesto Bribiesca por su apoyo y confianza en la realización de este trabajo.

A mis sinodales, Dr. Hermilo Sánchez, Dr. Yann Frauel, Mat. Ana Luisa Sólis y M. en C. Miguel Miranda; por el tiempo empleado en la revisión y corrección de este trabajo.

A un gran amigo, Oscar Fernando Niño Romero, por ser mi ángel de la guarda, por su cariño, por su confianza, por aplaudir mis logros y llamarme la atención en mis errores, por aconsejarme, por las buenas pláticas, por las travesuras, por los planes compartidos y sobre todo por creer en mí.

A Miguel Carrillo Barajas por su amistad, por su confianza, por sus enseñanzas y por ser mi sexto sinodal; gracias por ayudarme con la revisión de mi trabajo.

A mis amigos y compañeros durante mi estancia en la universidad: Iván de Jesús Arana Hernández, por su amistad, confianza, cariño y por escucharme; José Israel Campero Domínguez, por cuidarme y preocuparse por mí; Rosa Elba García Velasco, por todos los momentos compartidos; Erick Ruiz Rojas, por su cariño y por

¹No imagino mi vida sin ti hermanita.

confiar en mí; Marisol Flores Castro y Cesar Flores Castro, por estar al pendiente de mí; Humberto Ferrer Camacho, por sus travesuras que sin ellas me hubiera aburrido en clase; Selene Marisol Martínez Ramírez por los momentos divertidos; Luis Rey Rutiaga Robles, Yrasen Torres Morales, Yasmine Macedo Reza, María del Carmen Reyna Ramos González y Hugo González Medina, por su amistad y cariño. Gracias a todos por hacer más alegre y divertido mi paso por la facultad.

A Egar Arturo García Cárdenas, de quién aprendí que todo es posible a base de dedicación y esfuerzo; por su ayuda en los momentos difíciles y por el gran amor que me tuvo.

A la banda, Araceli Liliana Reyes Cabello², Karina León Barragán, Citlali Calderón de Anda, Edgar Bermúdez Contreras, Canek Pelaez Valdés, Gustavo de la Cruz Martínez, Rafael Pastrana Añorve, Oscar Ruiz, Federico Juárez Almaraz, Ismael Hernández³, por su amistad, cariño, confianza, por su ayuda, por los momentos de diversión y por darme la oportunidad de tener muchos hermanos.

A todas aquellas personas que intervinieron en mi vida personal y en mi formación académica.

Finalmente, pero no menos importante, sino todo lo contrario, a alguien muy especial, Erick Celso Zavaleta González, por ser él mismo, por la seguridad y confianza que me inspira, por su amistad, por su enorme cariño, por su nobleza y por todos los momentos que hemos compartido.

²Gracias por prestarme a la "gorda" para calcular algunos de los resultados mostrados en este trabajo.

³Gracias por ayudarme con los cálculos de los resultados mostrados.

Índice general

1. Introducción	1
1.1. Introducción a la descripción y comparación de formas	1
1.2. Origen del proyecto	2
1.3. Objetivo	4
1.4. Aportación de la tesis	4
1.5. Organización de la tesis	5
2. Segmentación de Imágenes	9
2.1. Introducción	9
2.2. Localización de discontinuidades	10
2.2.1. Localización de puntos	11
2.2.2. Localización de líneas	12
2.2.3. Localización de bordes	13
2.3. Unión de bordes y localización de límites	17
2.3.1. Análisis local	18
2.3.2. Análisis global	19
2.4. Aplicación de umbrales	21
2.4.1. Umbral	21
2.4.2. Iluminación	22
2.4.3. Umbral global	23
2.5. Segmentación por regiones	25
2.5.1. Consideraciones básicas	25
2.5.2. Crecimiento de regiones agregando píxeles	26
2.5.3. División y unión de regiones	28
2.6. Morfología matemática	29
2.6.1. Erosión	29
2.6.2. Dilatación	31
2.6.3. Opening	32
2.6.4. Closing	32

3. Medidas de Similitud de Formas 2D	35
3.1. Introducción	35
3.2. Invariantes	36
3.2.1. Traslación	37
3.2.2. Rotación	37
3.2.3. Escala	39
3.3. Números de forma	39
3.3.1. Cadenas de Freeman	40
3.3.2. Descripción de los Números de forma	40
3.3.3. Comparación de formas	43
3.4. Correlación máxima	45
3.5. Transformación de un objeto a otro	47
3.5.1. Gráficas	47
3.5.2. Proceso de transformación	48
3.5.3. Algoritmo húngaro	49
3.5.4. Medida de similitud	55
3.6. Compacidad de forma	56
3.6.1. Perímetro	56
3.6.2. Perímetro de contacto	56
3.6.3. Compacidad clásica	57
3.6.4. Compacidad discreta	57
3.6.5. Compacidad discreta máxima y mínima	58
3.6.6. Compacidad discreta normalizada	59
3.6.7. Comparación de formas usando compacidad	59
4. Aplicación y Resultados	61
4.1. Introducción	61
4.2. Segmentación de imágenes	62
4.3. Aplicando invariantes	78
4.4. Comparación de formas	78
Conclusiones	87
A. Definiciones	89
A.1. Imagen digital	89
A.2. Pixel	90
A.3. Nivel de gris	90
A.4. Pixelización	90

B. Filtros	93
B.1. Filtro	93
B.2. Modificación del histograma	93
B.3. Filtros lineales	94
B.4. Filtro media	94
B.5. Filtro Gaussiano	95

Capítulo 1

Introducción

1.1. Introducción a la descripción y comparación de formas

Un ser humano analiza las formas mediante su sistema visual que está diseñado para realizar percepción organizada. La mente estructura información consistente de características tales como color, forma, tamaño, distancia, posición, y textura, para representar escenas visuales. Estas características se perciben como propiedades de los objetos en la escena, no sólo se perciben líneas abstractas o superficies; no se perciben líneas aisladas, se perciben objetos. Todas las partes de cada objeto son vistas juntas en un todo; no como elementos separados, independientes y libres. Todos los objetos son percibidos con relación a otros objetos que se ubican cerca, lejos, atrás, contiguos, o adelante.

De la misma manera, el ser humano a simple vista puede comparar o clasificar formas. Primero realiza la comparación de las formas completas mediante una asociación de similitudes, identificando así a las formas exactamente iguales; posteriormente, si no son iguales, realiza la comparación por segmentos de las formas determinando qué tan parecidas son. La asociación depende de cada individuo porque es mediante un criterio personal y no siempre es el mismo.

Ahora, ¿cómo puede una computadora describir y comparar formas?. Esta es una pregunta que las áreas de Reconocimiento de Patrones y Visión por Computadora [1] se han encargado de resolver.

Desde hace varios años se ha tenido interés por reconocer y clasificar formas. Las áreas de Reconocimiento de Patrones, Procesamiento Digital y Visión por Computadora se han ocupado de este problema desde la década de 1950, por eso, actualmente existen muchas teorías sobre la descripción y el reconocimiento de formas; cada

teoría es un intento por explicar algún aspecto específico del problema.

Algunas de las teorías ya desarrolladas usan el contorno de la forma para describirla, pero no toman en cuenta el color y la textura. Un cerebro humano, como ya se dijo, toma en cuenta todas y cada una de las características de los objetos a comparar, incluyendo el color y la textura, y si dichos objetos no coinciden en estas características entonces determina que no son iguales. Realizar una comparación automatizada¹ depende de qué interés se tiene para comparar a los objetos, para determinar qué característica de ellos se toma en cuenta. Básicamente, esto es lo que hacen los métodos ya desarrollados para comparar formas: consideran alguna característica de los objetos que sea de interés específico para la comparación.

Otras características que se toman en cuenta, primordialmente de formas planas, son el área y el perímetro; propiedades que son descriptores básicos. También se encuentra la utilización de ejes máximos y mínimos, ejes principales, la densidad de los objetos, los ejes medios, los vértices, la cantidad de huecos, el grosor de las formas que se calcula mediante la compacidad, los ejes de simetría, las regiones, y la convexidad entre otros.

La mayoría de los métodos de reconocimiento de formas pretenden tener una gran habilidad para describirlas con la mayor precisión posible. Similarmente, los métodos de comparación de formas requieren de eficiencia en los análisis de similitud. Aunque la similitud también depende del criterio de cada persona. Por eso, en la descripción y en la comparación de objetos se toman en cuenta características de suma importancia: la orientación; el tamaño; y la pixelización. Si no se consideran estas características, los resultados de la descripción o comparación pueden ser incorrectos. Algunas técnicas que miden similitud lo hacen en base a la descripción de los objetos, y otras, primero sobreponen (correlación) a las formas para establecer equivalencia entre ellas.

Todos los métodos dedicados al análisis y comparación de formas comprueban que las formas más simples son las de más fácil proceso.

1.2. Origen del proyecto

Antes de definir el objetivo principal de este trabajo se dará una breve explicación del motivo para su realización.

El origen de este trabajo parte de un estudio iconográfico que se realiza actualmente sobre las culturas prehispánicas de Mesoamérica. Para tener conocimiento del área geográfica de Mesoamérica se dará una breve y esquemática exposición de su geografía e historia [2].

¹Usando una computadora.

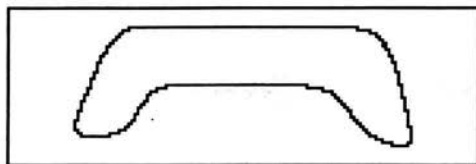


Figura 1.1: Símbolo que aparece en los monumentos de las culturas mesoamericanas.

Se llama actualmente Mesoamérica a la zona geográfica que comprende: al norte, las partes meridionales de los actuales estados mexicanos de Sinaloa, Durango, Zacatecas, San Luis Potosí y Tamaulipas; y al este, al oeste y al sur, las restantes zonas del México actual, los países de Belice, Guatemala, El Salvador, y las regiones occidentales de Honduras, Nicaragua y Costa Rica.

En dicha zona hizo su aparición alrededor del año 1500 a. C., un desarrollo civilizador que se prolongó en el tiempo hasta el primer cuarto del siglo XVI de nuestra era y hasta el contacto con la cultura Occidental.

En la actualidad se han encontrado numerosos datos arqueológicos textuales y lingüísticos que han permitido proporcionar información sobre el origen y ocaso de esas culturas: pirámides, construcción de vasijas, herramientas, conocimientos calendáricos; y también, piedras talladas, principios de escritura y signos plasmados. El estudio de estos últimos constituyen el objetivo del estudio iconográfico.

En la investigación iconográfica se ha detectado la peculiar aparición de un signo en particular, en diferentes objetos pertenecientes a estas culturas, por tal motivo, se deduce que dicho signo o símbolo es la representación de algo importante y es por eso su frecuente aparición. Visualmente se ha detectado esta repetitiva aparición en un mismo objeto o monumento, así como en muchos otros. Hasta este momento, la verificación visual se ha realizado sobre una pequeña muestra de los muchos monumentos existentes, pero interesa demostrarlo en un conjunto amplio. El símbolo en cuestión es mostrado en la figura 1.1.

A partir del análisis realizado en los monumentos, se ha detectado que el símbolo mostrado en la figura 1.1 no es el único que se repite, se han encontrado otros los cuales serían sus variantes. Los otros símbolos detectados se muestran en la figura 1.2. Todos estos símbolos, incluyendo el de la figura 1.1, son llamados "iconos uno".

Debido a que realizar una exhaustiva búsqueda manual de los símbolos en todos los monumentos resulta un proceso muy lento, las personas interesadas en resolver este problema se han acercado a gente especializada que puede solucionarlo mediante

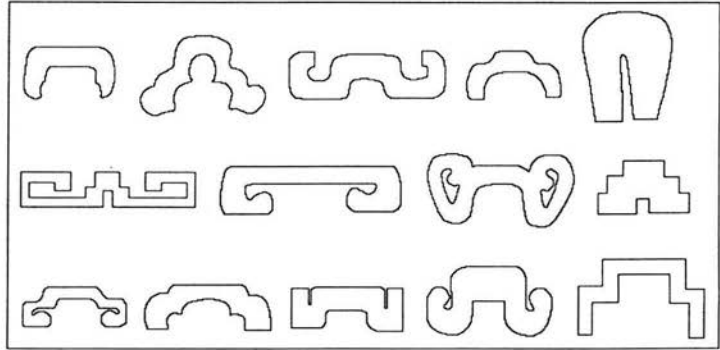


Figura 1.2: Otros símbolos que aparecen constantemente en los monumentos de las culturas mesoamericanas.

la aplicación de conocimientos de procesamiento digital de imágenes, porque con el uso de la computadora se puede automatizar el proceso, construyéndose una herramienta de aplicación en la identificación subsecuente de iconos semejantes.

1.3. Objetivo

Por las razones expuestas en la sección anterior el objetivo principal del desarrollo de esta tesis es ayudar a demostrar que los “iconos uno”, símbolos mostrados en las figuras 1.1 y 1.2, aparecen en la mayoría de los monumentos existentes de las culturas mesoamericanas. Como herramienta se usará la computadora y se aplicarán conocimientos sobre el procesamiento digital de imágenes y el reconocimiento de patrones, áreas que forman parte de las Ciencias de la Computación.

Primero se digitalizará a las imágenes de los diferentes monumentos, éstas son obtenidas a través de fotografías. Posteriormente, se realizará una segmentación de imágenes. Y finalmente se aplicarán algunos algoritmos que permiten detectar la similitud entre dos símbolos.

1.4. Aportación de la tesis

Este trabajo es una muestra de la gran diversidad de aplicaciones que tiene el Procesamiento Digital y el Reconocimiento de Patrones.

Esta tesis es un desarrollo de la aplicación de búsqueda y comparación de formas. Como resultado se puede apreciar que los métodos desarrollados en las áreas de Reconocimiento de patrones y Procesamiento de Imágenes tienen un campo muy amplio de aplicación, y en este caso el campo de aplicación es la historia.

La aportación principal de la tesis es el reconocimiento de los símbolos "iconos uno" en las imágenes de las culturas mesoamericanas, obteniéndose una herramienta de aplicación en la localización de símbolos.

Además de ver una aplicación concreta y del mundo real sobre el Reconocimiento de Patrones, en esta tesis se puede observar una recopilación de algunos algoritmos de segmentación y comparación de imágenes, la cual puede servir como referencia básica para conocer estos temas.

1.5. Organización de la tesis

La presente tesis se divide en tres partes: Segmentación de imágenes, Comparación de formas y Aplicación real de estos conceptos. A cada una de estas partes se dedica un capítulo.

En la primera parte que corresponde al capítulo 2, se detalla la segmentación de imágenes la cual es el primer paso del procesamiento digital y del reconocimiento de patrones. Sin una segmentación sería prácticamente imposible el análisis de las imágenes. La segmentación de imágenes se define como la extracción de los objetos o formas que pertenecen a la imagen, los cuales son lo que la definen. Algunos de los métodos de segmentación son mostrados aquí. El método más usado y muy fácil de implementar es la aplicación de umbrales (sección 2.4), pero hay que tener en cuenta que este método no resulta muy eficiente cuando las intensidades de niveles de gris de los píxeles varían en un rango pequeño; en relación a este tema, se verá en la sección 2.4.2 el problema de la iluminación en las imágenes.

Los puntos, las líneas y los bordes son los elementos básicos que se pueden encontrar en una imagen. Antes de definir la aplicación de umbrales, se exponen técnicas para la localización de éstos usando como herramienta a las plantillas [10]. Para este caso, también se toman en cuenta los niveles de gris, porque a través de las plantillas se determina la existencia de alguno de estos elementos haciendo referencia a los niveles de gris de los píxeles determinados por la plantilla.

Otro método que puede ser efectivo pero es muy lento en su procesamiento es la segmentación por regiones (sección 2.5) descrita por Rafael González [10]. Este método se puede realizar mediante dos procesos distintos e independientes. El primero se basa en la división de la imagen para generar regiones y posteriormente se unen adecuadamente para obtener a los objetos. El segundo obtiene las regiones a

partir de una semilla generadora; la efectividad de este proceso radica en la elección correcta de dicha semilla. Para ambos procesos, no sólo se toman en cuenta los niveles de gris, si no también propiedades que deben cumplirse para que un pixel forme parte de alguna región.

Para finalizar esta primera parte se describen algunas operaciones morfológicas (ver Procesamiento de Imágenes Binarias en [7]): erosión, dilatación, opening y closing; éstas ayudan a la segmentación dado que engrosan o adelgazan los objetos permitiendo así la eliminación de partículas no deseadas o el relleno de huecos. La aplicación de estas operaciones puede ser un paso previo o posterior a la segmentación.

En la segunda parte (capítulo 3) se analiza el reconocimiento de patrones. Algunas técnicas de comparación de formas son mostradas en este capítulo, particularmente se describen las técnicas para formas planas, es decir, en dos dimensiones. Las formas a comparar son los objetos obtenidos de una segmentación previa.

La comparación de formas puede hacerse con un simple emparejamiento de ellas o con métodos más complejos que requieran de sus características. El método de comparación mediante el emparejamiento es conocido como correlación máxima [6] (sección 3.4), éste encuentra la sobreposición de dos formas la cual cubre la mayor parte de su área. La condición para llevar a cabo la comparación de formas usando correlación máxima es que ambas formas deben tener la misma área.

Una de las características de las formas que se toma en cuenta para la comparación es el comportamiento de su contorno, es decir, qué tipo de líneas lo componen: líneas rectas, líneas que forman esquinas hacia afuera o líneas que forman esquinas hacia adentro. Cada tipo de línea se representa con un código numérico que permite dar la descripción del contorno mediante su recorrido. Como se puede apreciar, las líneas curvas no pertenecen al conjunto de representación debido a que el contorno de las formas es discreto². Con el código numérico se tiene una descripción de la forma, y con él se pueden comparar formas. A este método de comparación se le llama números de forma [5] (sección 3.3) y proporciona una medida de similitud a través del árbol de similaridad generado por los órdenes de los números de forma de cada forma a comparar.

Otro método de comparación, relativamente reciente, es la transformación de un objeto en otro (sección 3.5), dicho método lo mencionan Bribiesca y Wilson en [6]. En este método, una de las formas a comparar es la que precisamente se transforma en la otra, para ello se requiere que ambas formas tengan la misma área, en este caso, deben tener la misma cantidad de pixeles. Primero se realiza una sobreposición de las formas para encontrar los pixeles no comunes de ambas formas. Al conjunto

²Contorno formado únicamente de líneas rectas.

de píxeles de la primer forma que no son comunes con los píxeles de la segunda forma son los píxeles positivos, en caso contrario son los píxeles negativos, es decir, son los píxeles de la segunda forma que no son comunes con los píxeles de la primer forma. Ya localizados los píxeles positivos y negativos se procede a trasladar los positivos al lugar de los negativos obteniendo así la transformación. El movimiento de píxeles genera una distancia de traslado que indica un costo de transformación y mediante este costo se puede determinar el grado de similitud, a menor costo mayor similitud. Para tener la opción correcta de un mejor costo, para este caso el costo mínimo, se requiere encontrar un apareo óptimo entre los píxeles positivos y negativos en donde la suma de sus distancias sea la menor. El algoritmo que sirve para encontrar el apareo óptimo es una variante del algoritmo descrito por James R. Evans y Edward Minieka [8] llamado algoritmo húngaro³, desarrollado por H. Kuhn [14].

Otro método usado para la comparación de formas es el uso de la compacidad [4] (sección 3.6). El uso de esta propiedad no proporciona un porcentaje de similitud, sino que determina que tan compacta es la forma, dicho de otra manera, la compacidad puede clasificar a las formas de acuerdo a la característica de ser gruesas o delgadas. La definición de compacidad es: la cantidad de lados de los píxeles que se tocan entre sí.

Como se puede notar, la mayoría de las técnicas de comparación de formas usan la sobreposición. En la sección donde se analiza la correlación máxima se usa la sobreposición que cubre la mayor área. Para las otras técnicas también se puede usar este tipo de sobreposición, pero de acuerdo con estudios previos y pruebas realizadas se ha comprobado que la mejor posición que deben tener las formas para una sobreposición es hacer que sus centros de masa y sus ejes principales coincidan. Además de que éstos deben coincidir, las formas deben tener la misma cantidad de píxeles (misma área), teniendo así, que las formas deben ser invariantes en tamaño, posición y orientación (sección 3.2).

En la tercera y última parte (capítulo 4) se describe detalladamente el proceso de búsqueda de las formas deseadas (“íconos uno”) inmersas en imágenes de las culturas mesoamericanas. Aquí se pueden ver ejemplificados los procesos de segmentación de imágenes y comparación de formas, es decir, se ve claramente el desarrollo de este trabajo. El método empleado para la comparación de formas es: la transformación de un objeto a otro; complementando estos resultados con el uso de compacidades.

Finalmente se agregan dos anexos (apéndices A y B); en el primero se muestran algunas definiciones de términos importantes, los cuales hay que tomar en cuenta

³El algoritmo húngaro original encuentra un apareo óptimo en donde la suma de las distancias es la mayor.

debido a que son esenciales en esta área. Y en el apéndice B se dan algunas técnicas del filtrado de imágenes que se pueden tomar en consideración en caso de ser necesarias, por ejemplo, cuando una imagen presenta partículas de ruido. Básicamente, el filtrado de imágenes ayuda al análisis y puede ser un paso previo a la segmentación.

Capítulo 2

Segmentación de Imágenes

2.1. Introducción

La segmentación es el proceso que se lleva a cabo para dividir una imagen en las partes u objetos que la forman. Dicho proceso es de gran importancia para el análisis de imágenes, porque analizar a la imagen completa resulta muy difícil, en cambio, si primero se lleva a cabo una segmentación entonces los objetos extraídos de la imagen pueden ser más fáciles de identificar, o bien, pueden ser analizados con procesos subsiguientes, tales como la descripción y el reconocimiento.

La segmentación se puede realizar manualmente, pero resultaría un proceso muy lento. Es por ello que se usan algoritmos para automatizar este proceso. Los algoritmos de segmentación generalmente se basan en las propiedades de los niveles de gris de la imagen, como son: discontinuidad y similaridad. En el caso de la discontinuidad, la imagen se divide considerando los cambios repentinos en el nivel de gris de los píxeles, localizando así, puntos aislados, líneas y bordes; localizar estos elementos es el principal interés en esta propiedad. Para la otra propiedad, similaridad, la aproximación a la división de la imagen es mediante umbrales [7], la localización de regiones crecientes, regiones partidas y regiones unidas.

Las formas más simples que pueden aparecer en una imagen son los puntos, las líneas y los bordes. Estos, por su estructura simple son muy fáciles de extraer. En la sección 2.2 se expone a detalle el proceso para la localización de puntos, líneas y bordes; y la utilización de plantillas como herramienta para su búsqueda. Con frecuencia, los bordes localizados están incompletos o distorsionados, debido a que las imágenes presentan ruido. En la sección 2.3 se proporcionan técnicas que realizan el ligado de bordes incompletos y la localización de los mismos. Este proceso de ligado utiliza gráficas para su propósito.

En la sección 2.4 se define el método de segmentación mediante umbrales, también se explican detalladamente sus variaciones y el problema que se presenta cuando una imagen está muy iluminada. Hay que tener presente que la aplicación de umbrales es un procedimiento muy importante en la segmentación de imágenes [12], y es el más usado debido a que es confiable y técnicamente simple.

En la sección 2.5 se expone sobre el método de segmentación mediante regiones. Este método puede ser implementado con dos técnicas distintas e independientes. En la primera se generan regiones mediante la agregación de píxeles a una semilla generadora. En la segunda se divide a la imagen en regiones obteniendo así las formas a extraer. Este procedimiento también requiere de la unión de regiones porque con la simple división no produce las formas exactas (sólo serían formas rectangulares). El problema con esta técnica es que es un proceso lento.

En la parte final de este capítulo, sección 2.6, se describen las operaciones que pertenecen a la morfología matemática. La erosión y dilatación son las operaciones básicas y las otras que se detallan en esta sección, opening y closing, son derivaciones de ellas; además de estas operaciones hay otras que no se mencionan aquí. Las operaciones morfológicas, principalmente ayudan a eliminar el ruido de una imagen, a definir los contornos de los objetos, o a rellenar huecos si así se desea. La intención de esta sección es presentar métodos aplicables a las imágenes binarias para lograr una mejor segmentación de las mismas.

2.2. Localización de discontinuidades

Una manera de realizar la segmentación de una imagen es a través de la localización de discontinuidades. Este método permite encontrar puntos, líneas y bordes en una imagen. En ésta sección se presentan algunos métodos que utilizan *plantillas* para encontrar estas propiedades.

Una **plantilla** (también llamada *máscara*, *ventana* o *filtro*) es una pequeña imagen predefinida. Dicho de otra manera, es un arreglo de dos dimensiones, y para cada localidad del arreglo hay valores preestablecidos (valores de *nivel de gris*¹). Una localidad del arreglo pretende cubrir un *pixel* (ver apéndice) en la imagen. La plantilla sirve para detectar alguna propiedad en las imágenes mediante un desplazamiento de ella sobre toda la imagen. A lo largo de ésta sección se utilizará una plantilla de 3×3 , tal como se muestra en la figura 2.1.

Los $w_i, i = 1, 2, \dots, 9$, mostrados en la figura 2.1 son los coeficientes de cada localidad en la plantilla. Sean $x_i, i = 1, 2, \dots, 9$, los valores de nivel de gris de los píxeles en la imagen que se encuentran bajo cada localidad de la plantilla. Ambos,

¹Ver apéndice.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 2.1: Plantilla de 3×3 .

los coeficientes y los niveles de gris correspondientes pueden ser expresados como vectores columna:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_9 \end{bmatrix} \quad (2.1)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} \quad (2.2)$$

La manera de relacionar ambos vectores es mediante el producto interno dado en la siguiente ecuación:

$$w'x = w_1x_1 + w_2x_2 + \dots + w_9x_9 \quad (2.3)$$

donde w' es w transpuesto. Es así como se pueden detectar puntos aislados u otras propiedades de las imágenes, tal y como se verá más adelante. Observe que la notación usada para $w'x$ es general y aplicable a una plantilla de tamaño arbitrario. Por ejemplo, para una plantilla de $n \times n$, se trabajará con vectores de dimensión n^2 .

2.2.1. Localización de puntos

Los objetos más simples que se pueden encontrar en una imagen son los puntos. Éstos se encuentran fácilmente usando una plantilla. La plantilla básica usada para la detección de puntos aislados es mostrada en la figura 2.2. La búsqueda se realiza mediante un desplazamiento de la plantilla de pixel en pixel por toda la imagen haciendo coincidir el centro de la plantilla con cada pixel en la imagen.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 2.2: Plantilla para detectar puntos aislados.

Para saber si el pixel que se encuentra bajo el centro de la plantilla es un punto aislado se aplica la ecuación 2.3 usando los coeficientes dados por la figura 2.2, es decir, se usa la siguiente ecuación:

$$w'x = -x_1 - x_2 - x_3 - x_4 + 8x_5 - x_6 - x_7 - x_8 - x_9. \quad (2.4)$$

Si la ubicación de la plantilla es sobre un área de la imagen en donde los niveles de gris son constantes (iguales), el resultado de la aplicación de la ecuación anterior debería ser 0. Dicho de otra manera, si el pixel ubicado bajo el centro de la plantilla es un punto aislado, a saber x_5 , entonces el resultado debería ser diferente de 0.

En la práctica, no siempre resulta que alrededor de un punto los niveles de gris son constantes, frecuentemente se encuentran con valores de nivel de gris diferentes, pero tal diferencia no es muy amplia debido a que delimitan un punto. Entonces al usar la ecuación 2.4 para evaluar esta área produce un valor diferente de 0. Pero, si no hay un punto y además los niveles de gris son diferentes, el resultado de la evaluación también es diferente de 0. Para saber si es un punto hay que evaluar el resultado mediante un umbral U . Si

$$|w'x| > U \quad (2.5)$$

entonces en el área bajo la plantilla hay un punto. Donde U es un valor *umbral* (ver sección 2.4) establecido.

2.2.2. Localización de líneas

La localización de puntos es bastante sencilla y simple. Ahora, se presenta un procedimiento similar para localizar líneas en una imagen. El método para localizar líneas usa las plantillas mostradas en la figura 2.3. La plantilla mostrada en la figura 2.3.a sirve para detectar líneas horizontales, la plantilla mostrada en 2.3.b para líneas verticales, y las plantillas mostradas en las figuras 2.3.c y 2.3.d sirven para detectar líneas diagonales, una orientada a 45° y la otra a -45° respectivamente.

-1	-1	-1
2	2	2
-1	-1	-1

a)

-1	2	-1
-1	2	-1
-1	2	-1

b)

-1	-1	2
-1	2	-1
2	-1	-1

c)

2	-1	-1
-1	2	-1
-1	-1	2

d)

Figura 2.3: Plantillas para detectar líneas. a) Plantilla que detecta líneas horizontales. b) Plantilla que detecta líneas verticales. c) Plantilla que detecta líneas diagonales orientadas a 45 grados. d) Plantilla que detecta líneas diagonales orientadas a -45 grados.

Se pueden definir otras plantillas que localicen líneas con orientaciones diferentes a 0° , 45° , -45° y 90° . Dependiendo de la orientación de las líneas a localizar se determinan las localidades en la plantilla que permitirán localizarlas; para estas localidades el coeficiente será 2, en las demás localidades deberá ser -1.

La aplicación del procedimiento para detectar líneas es similar a la localización de puntos aislados. Cada una de las plantillas mostradas en la figura 2.3 son desplazadas por toda la imagen y en cada posición se evalúa la ecuación 2.3 con los coeficientes de cada una de las plantillas. De la misma manera en que se hizo para la localización de puntos, se determina si en el área sobre la que se ubica la plantilla hay una línea o no, tomando en cuenta el resultado de la evaluación de la ecuación 2.3 con los coeficientes de las plantillas mostradas en la figura 2.3. Habrá una línea si se cumple con la condición dada en la ecuación 2.5. Si se cumple con esta condición y dependiendo de la plantilla usada se puede determinar la orientación de la línea que se ha detectado.

2.2.3. Localización de bordes

La localización de puntos y líneas son temas básicos y fundamentales en segmentación, pero en muchas aplicaciones de interés práctico no ocurren frecuentemente. En cambio, la localización de bordes [12] es una aproximación más común para detectar discontinuidades significativas en niveles de gris.

Se define un *borde* como el límite entre dos regiones con propiedades de distinto nivel de gris. Para la localización de bordes se asume que las regiones en cuestión son suficientemente homogéneas. Por lo tanto, la transición entre dos regiones puede ser determinada sólo en las discontinuidades básicas de nivel de gris.

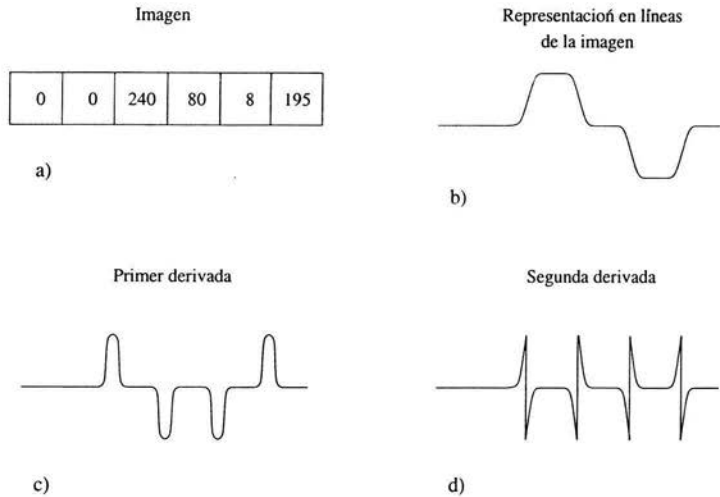


Figura 2.4: Representación de una imagen en líneas y su primera y segunda derivadas. a) Parte de un renglón de una imagen. b) La representación en líneas de la imagen en 2.4.a. c) Primer derivada de la imagen en su representación en líneas. d) Segunda derivada de la imagen en su representación en líneas.

El método para la localización de bordes es mediante el cálculo de derivadas locales. Para poder aplicar las derivadas a una imagen primero hay que representar a dicha imagen como líneas, una por cada renglón. Cada pixel se representa con un segmento de línea horizontal. Las líneas de los pixeles vecinos se unen con una diagonal si sus niveles de gris son diferentes, y con una línea horizontal si son iguales; estas uniones se hacen de manera suave. La pendiente de las diagonales depende de la transición del nivel de gris entre los pixeles: es positiva si de un nivel de gris oscuro se va a uno claro; y es negativa en caso contrario, cuando la transición va de claro a oscuro. Cada diagonal representa un borde. En la figura 2.4.a se muestra una parte de un renglón de una imagen, y en la figura 2.4.b se muestra su respectiva representación en líneas.

La primer derivada aplicada a la imagen en su representación de líneas es 0 en toda región con nivel de gris constante (línea horizontal), y es un valor constante, diferente de 0, para cada transición de nivel de gris (diagonal). La segunda derivada es 0 en las zonas con nivel de gris constante, positiva en la parte de la transición

asociada con el lado oscuro del borde y negativa en la parte de la transición asociada con el lado claro. De esto se deduce que la magnitud de la primer derivada puede ser usada para detectar la presencia de un borde, y el signo de la segunda derivada puede ser usado para detectar de qué lado (claro u oscuro) del borde se encuentra un pixel.

Hasta este momento la representación de imágenes mediante líneas es limitada porque sólo es a través de líneas horizontales y diagonales, pero se puede aplicar un argumento similar a un borde de cualquier orientación. Simplemente se define una representación de líneas perpendicular a la dirección del borde en cualquier punto dado y se interpreta el resultado como se ha descrito.

A continuación se dará una descripción de la obtención de la primer derivada en cualquier punto mediante gradientes, descrita por Rafael González [10], También se dará la descripción de la segunda derivada dada por Laplace.

El gradiente de una imagen $f(x, y)$ en el punto (x, y) es definido como un vector de dos dimensiones:

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.6)$$

Para detectar un borde, es importante la magnitud de éste vector, simplemente llamado *gradiente* y denotado por $G[f(x, y)]$, donde

$$G[f(x, y)] = [G_x^2 + G_y^2]^{1/2} \quad (2.7)$$

Esta cantidad es igual al valor máximo de incremento de $f(x, y)$ por unidad de distancia en dirección de \mathbf{G} .

Es común una aproximación del gradiente por valores absolutos:

$$G[f(x, y)] \approx |G_x| + |G_y| \quad (2.8)$$

debido a que es una aproximación fácil de manejar e implementar.

La *dirección* del vector gradiente es también de suma importancia porque si $\alpha(x, y)$ es el representante del ángulo en dirección de \mathbf{G} en el lugar (x, y) entonces se tiene que

$$\alpha(x, y) = \tan^{-1}(G_y/G_x) \quad (2.9)$$

donde el ángulo es medido con respecto al eje coordenado x .

En la ecuación 2.6 se puede observar que el cálculo del gradiente es mediante la obtención de las derivadas parciales, $\partial f/\partial x$ y $\partial f/\partial y$, para cada posición de

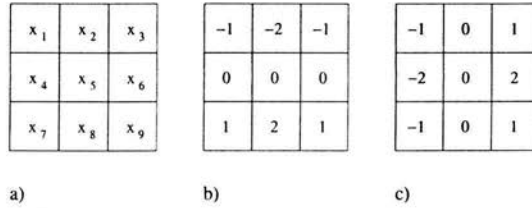


Figura 2.5: a) Una imagen con un área de 3×3 . b) Plantilla para calcular G_x . c) Plantilla para calcular G_y . Ambas plantillas son conocidas como operadores de Sobel.

los pixeles. Esto se puede hacer de diferentes maneras. Una aproximación puede realizarse tomando un área de 3×3 sobre un pixel ubicado en un punto (x, y) . Los componentes del vector gradiente en la dirección de x y y se definen como:

$$G_x = (x_7 + 2x_8 + x_9) - (x_1 + 2x_2 + x_3) \quad (2.10)$$

$$G_y = (x_3 + 2x_6 + x_9) - (x_1 + 2x_4 + x_7) \quad (2.11)$$

El hecho de usar un área de 3×3 para el cálculo del gradiente proporciona mayor alisamiento que usar una área de 2×2 , esto permite que las derivadas sean menos sensibles al *ruido*². También se produce alisamiento adicional haciendo que los pixeles cercanos al punto se multipliquen por 2.

Para calcular G_x y G_y , de acuerdo con las ecuaciones 2.10 y 2.11, se usan plantillas conocidas como *operadores de Sobel* (Rafael C. González [10]); estas plantillas se muestran en la figura 2.5. Los operadores de Sobel proporcionan respuestas dadas por la aplicación combinada con las ecuaciones 2.7 o 2.8 obteniendo el gradiente en cualquier punto (x, y) . Aplicando estas plantillas a una imagen cualquiera, se obtiene el gradiente en todos sus puntos. Al resultado de este proceso se le llama *imagen gradiente*.

Hay que hacer notar que las ecuaciones 2.10 y 2.11 son casos particulares de la ecuación 2.3. Entonces si x es un vector que contiene los valores de pixel mostrados en la figura 2.5.a, se tiene que

$$G_x = w'_1 x \quad (2.12)$$

$$G_y = w'_2 x \quad (2.13)$$

²A las distorsiones que a veces aparecen en las imágenes, tales como iluminación no uniforme, se les conoce como ruido. Vea el apéndice

0	1	0
1	-4	1
0	1	0

Figura 2.6: Plantilla usada para calcular la derivada de Laplace.

donde w_1 y w_2 son los vectores que contienen a los coeficientes de las plantillas mostradas en la figura 2.5.b y 2.5.c respectivamente.

La definición de la segunda derivada, dada por Laplace (ver [7]) es:

$$L[f(x, y)] = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2 \quad (2.14)$$

Usando la referencia de la figura 2.5.a, esta derivada, L , se define con respecto al punto (x, y) con nivel de gris x_5 como:

$$L[f(x, y)] = x_2 + x_4 + x_6 + x_8 - 4x_5 \quad (2.15)$$

Con esta definición (ecuación 2.15) se puede usar la plantilla mostrada en la figura 2.6 para calcular la derivada de Laplace. Usando la misma notación que se ha manejado, la definición de la derivada de Laplace es la siguiente:

$$L[f(x, y)] = w'x \quad (2.16)$$

donde w es el vector formado de los coeficientes de la plantilla mostrada en la figura 2.6, y x es el vector de los coeficientes de una imagen.

A pesar de que la derivada de Laplace permite detectar la transición de intensidad de niveles de gris, es poco usada para localizar bordes. Esto se debe a que es muy sensible al ruido.

2.3. Unión de bordes y localización de límites

En la sección anterior se describieron los métodos para localizar puntos, líneas y bordes. Estos métodos sitúan los píxeles que se encuentran entre los objetos y el fondo de la imagen, pero en muchas ocasiones no delimitan bien a los objetos, debido a que generalmente las imágenes presentan distorsiones. Para solucionar este problema, lo ideal es usar métodos que permitan unir bordes que no estén bien formados. En esta sección se verán algunas técnicas para el ligado de bordes en los casos en que sea necesario.

2.3.1. Análisis local

La unión de bordes se realiza sobre una imagen en la cual ya se detectaron los bordes. La localización de los bordes es a través de los procedimientos ya descritos en las secciones anteriores. Una manera simple de realizar la unión es mediante el ligado de pixeles sobre un borde ya detectado, para ello se analizan sus características en pequeñas áreas de la imagen gradiente. Todos los pixeles que cumplen con las mismas propiedades son ligados, formando así un límite.

Hay dos propiedades principales para establecer el ligado entre pixeles:

- La magnitud proporcionada por el operador gradiente usada para detectar un pixel sobre un borde.
- La dirección del gradiente.

La propiedad 1 se obtiene con el cálculo de $G[f(x, y)]$, como se definió en la ecuación 2.7 o 2.8. Entonces un pixel con coordenadas (x_1, y_1) en una área predefinida con centro en el punto (x, y) , es similar en magnitud al pixel ubicado en (x, y) si

$$|G[f(x, y)] - G[f(x_1, y_1)]| \leq U \quad (2.17)$$

donde U es un umbral.

Para la segunda propiedad, la dirección del vector gradiente está dada por la ecuación 2.9. Entonces de manera análoga, un pixel con coordenadas (x_1, y_1) dentro del área predefinida con centro (x, y) , tiene un ángulo similar al pixel ubicado en el punto (x, y) si

$$|\alpha(x, y) - \alpha(x_1, y_1)| < A \quad (2.18)$$

donde A es un ángulo umbral. Hay que notar que la dirección del borde en el punto (x, y) es perpendicular a la dirección del vector gradiente en ese punto. Sin embargo, para el propósito de comparación de las direcciones, la ecuación 2.18 produce un resultado equivalente.

Aplicando estos conceptos, se ligan los pixeles (x_1, y_1) y (x, y) , donde (x_1, y_1) es un pixel ubicado en el área predefinida con centro (x, y) , si se cumplen las ecuaciones 2.17 y 2.18. Este proceso se repite para cada punto de la imagen gradiente. A cada conjunto de pixeles ligados al pixel ubicado en el centro del área en cuestión, se le asigna un nivel de gris diferente.

2.3.2. Análisis global

Para las imágenes que presentan un alto grado de distorsión, los métodos antes descritos no son muy eficientes para detectar límites [10]. En ésta sección se presenta una aproximación global basada en la representación de imágenes mediante gráficas y la búsqueda de caminos de bajo costo que corresponden a bordes significativos. Con esta representación se tiene una aproximación rigurosa para imágenes que presentan distorsiones. Como se verá, este método es más complicado que los ya expuestos y por lo tanto su procesamiento requiere más tiempo de ejecución.

Antes de proceder, se darán algunas definiciones que se necesitan para describir el método de detección de bordes.

Definición 2.1 Una gráfica $G=(N,A)$ está formada por un conjunto finito no vacío de nodos, N , y un conjunto A , formado de parejas constituidas por distintos elementos de N . Cada pareja $(a_i, a_j) \in A$ es llamada arista.

Definición 2.2 Una gráfica $G=(N,A)$ es dirigida si cada arista $(a_i, a_j) \in A$ es dirigida. Una arista $(a_i, a_j) \in A$ es dirigida si (a_i, a_j) es un par ordenado; a_i es el nodo origen y a_j es el nodo destino.

Definición 2.3 Una gráfica pesada $G=(N,A)$ es aquella en la que cada $(a_i, a_j) \in A$ tiene asociado un valor numérico llamado peso y denotado por $w(a_i, a_j)$.

Definición 2.4 Un camino del nodo a_1 al nodo a_n es una secuencia de nodos a_1, a_2, \dots, a_n con $a_i \in N$, $i = 1, 2, \dots, n$, tal que $(a_i, a_{i+1}) \in A$.

Definición 2.5 El peso, w , de un camino a_1, a_2, \dots, a_n es dado por:

$$w = \sum_{i=1}^{n-1} w(a_i, a_{i+1}) \quad (2.19)$$

La representación de una imagen como una gráfica se construye tomando en cuenta a los pixeles; por cada conjunto de esquinas de pixel hay un nodo, los conjuntos son formados por esquinas que son adyacentes. Entre cada par de nodos hay dos aristas dirigidas con diferente sentido, si existe un sólo lado de pixel entre los conjuntos de esquinas (figura 2.7). A cada arista se le asigna un peso, el cual es calculado mediante la siguiente ecuación:

$$w(e_1, e_2) = I - (i(p_i) - i(p_j)), \quad (2.20)$$

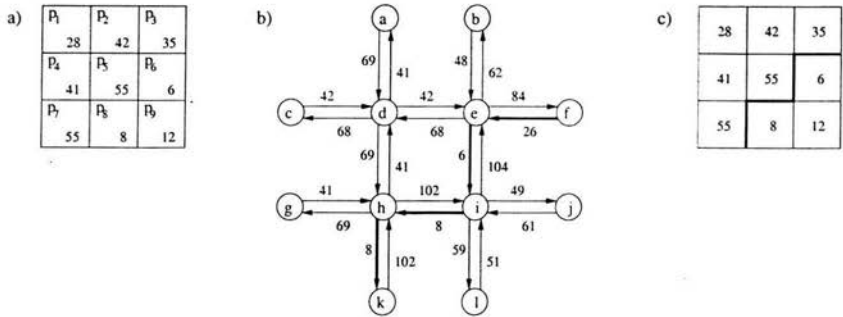


Figura 2.7: a) Imagen de 3×3 . b) Gráfica dirigida y pesada que representa a la imagen de la figura 2.7.a. c) Imagen de la figura 2.7.a con un borde marcado por una línea gruesa.

donde I es el valor de nivel de gris más grande de todos los pixeles de la imagen, mientras que $i(p_i)$ e $i(p_j)$ son los valores de nivel de gris de los pixeles p_i y p_j respectivamente, cuyo lado de pixel comparten.

Para aclarar los conceptos mencionados se verá un ejemplo. En la figura 2.7 se muestra una imagen representada por pixeles y su valor de nivel de gris para cada pixel (figura 2.7.a), también se muestra la gráfica correspondiente (figura 2.7.b). A cada nodo se le asignó arbitrariamente una etiqueta. Note que las aristas que aparecen entre cada par de nodos tienen diferente peso, eso se debe a que se calcula $w(e_1, e_2)$ y $w(e_2, e_1)$ para cada una de ellas. Por ejemplo, el peso de la arista dirigida (d, e) se obtiene con el cálculo de la ecuación 2.20, $w(d, e) = 55 - (55 - 42) = 42$, dado que $I = 55$, $i(p_5) = 55$ e $i(p_2) = 42$; en cambio, para la arista dirigida (e, d) se tiene que $w(e, d) = 55 - (42 - 55) = 68$.

Ya que se tiene la imagen en forma de gráfica, se requiere encontrar bordes. Para ello se localiza el camino que tenga el menor peso. Dicho de otra manera, el camino en la gráfica cuyo peso es el menor determina un borde en la imagen. La longitud del camino se determina con el tamaño de la imagen (o sub-imagen) o mediante un criterio personal. En la figura 2.7.b está marcado con flechas más gruesas el camino con el peso mínimo, correspondiente a los nodos f, e, i, h, k . Una vez que ya se detectó el camino con el peso mínimo, los pixeles que corresponden a los nodos del camino forman un borde en la imagen (figura 2.7.c).

En general, el problema de encontrar el camino con el peso mínimo no es trivial desde un punto de vista computacional. Algunos algoritmos sacrifican efectividad a

cambio de tiempo de ejecución, o viceversa.

2.4. Aplicación de umbrales

Los umbrales son una de las aproximaciones más importantes para el proceso de la segmentación de imágenes, además son muy usados. En esta sección, se verán diferentes técnicas para la aplicación de umbrales, y las ventajas y limitaciones que existen para estos métodos.

2.4.1. Umbral

Un umbral es un valor, condición o característica que divide a un grupo en dos o más partes. Generalmente, los umbrales que se usan para la segmentación de imágenes son los de valor numérico debido a que se desea separar los píxeles (y el nivel de gris de un pixel es un valor numérico entre 0 y 255) que forman objetos de los que no, pero también hay casos en donde se usan los otros tipos de umbrales (no numéricos) o combinación de ellos.

Una de las razones principales por las cuales se usan umbrales es porque a través de la representación de una imagen en un histograma es muy fácil de distinguir objetos. El histograma se realiza en base a la frecuencia de los niveles de gris y dado que un objeto en una imagen tiene el mismo nivel de gris para cada uno de los píxeles que lo forman o varían en un rango pequeño, entonces las frecuencias para estos valores serán amplias y notorias.

Para distinguir un objeto en una imagen usando un umbral, se genera una imagen nueva cambiando el valor de nivel de gris a cada uno de los píxeles de la imagen original. Si la imagen a la cual se le quiere extraer el objeto es $f(x, y)$ entonces la imagen que se obtiene de aplicarle un umbral U es:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > U \\ 0 & \text{si } f(x, y) \leq U \end{cases} \quad (2.21)$$

Teniendo así que los píxeles con valor de nivel de gris igual a 1 para la imagen $g(x, y)$ corresponden al objeto, mientras que los píxeles con nivel de gris igual a 0 forman parte de lo que sería el fondo de la imagen. Los valores de 0 y 1 son los que normalmente se usan para definir a las imágenes que se obtienen de la aplicación de umbrales, las cuales se les conoce como *imágenes binarias*, pero estos valores (0 y 1) pueden variar según las necesidades, por ejemplo: 1 y 10.

En una imagen no sólo se puede encontrar un objeto usando un umbral, se pueden encontrar varios. Para ello, se obtienen imágenes definidas en forma similar

a la que se definió en la ecuación 2.21. Por ejemplo, para una imagen con dos objetos se tiene que los píxeles que cumplen con $U_1 < f(x, y) \leq U_2$ forman a un objeto, los píxeles que cumplen con $f(x, y) > U_2$ forman al segundo objeto, y los píxeles que cumplen con $f(x, y) \leq U_1$ son parte del fondo de la imagen; donde U_1 y U_2 son dos diferentes umbrales. A este tipo de umbrales se les conoce como multivaluados y generalmente son menos confiables que los umbrales simples (como el que se usa en la definición de la ecuación 2.21) porque hay una gran dificultad en establecer múltiples umbrales que efectivamente correspondan a regiones aisladas que son de interés, especialmente cuando el número de regiones sobresalientes en el histograma es grande.

Existen tres tipos de umbrales: global, local y dinámico. El umbral global es aquel que solamente depende de $f(x, y)$. Un umbral es local si depende de $f(x, y)$ y además depende de una condición $p(x, y)$ (por ejemplo, una función que evalúa el promedio del nivel de gris de una vecindad centrada en el pixel ubicado en el punto (x, y)). El umbral dinámico es el que depende de la ubicación de cada uno de los píxeles, es decir, de las coordenadas (x, y) .

2.4.2. Iluminación

Un punto muy importante que hay que considerar en una imagen es la iluminación, Rafael González [10] hace referencia a este tema. En ocasiones, debido a la iluminación no uniforme o escasa que presentan las imágenes resulta difícil realizar el proceso de segmentación mediante un umbral simple. Para ilustrar este problema considere la imagen $f(x, y)$ mostrada en la figura 2.8.a. El histograma de esta imagen, mostrado en la figura 2.8.b, es claramente formado por dos montañas que representan a dos regiones de la imagen que deberían ser fácilmente segmentadas debido a la presencia de un valle formado por la unión de las dos montañas. Suponga que a dicha imagen se le agrega cierta iluminación, es decir, se multiplica la función $f(x, y)$ por la función de iluminación $i(x, y)$ mostrada en la figura 2.8.c para producir la imagen $g(x, y)$ mostrada en la figura 2.8.d. El histograma de esta imagen es mostrado en la figura 2.8.e. Observe que el valle original fue eliminado y realizar una segmentación utilizando un umbral simple es un proceso imposible.

La solución al problema de la iluminación no es sencilla debido a que a partir de la imagen $g(x, y)$ no es fácil extraer las funciones $f(x, y)$ e $i(x, y)$ dado que no se conoce esta última. En caso de conocerla, la operación a realizar es una simple división $g(x, y)/i(x, y)$, pero generalmente la función $i(x, y)$ es desconocida.

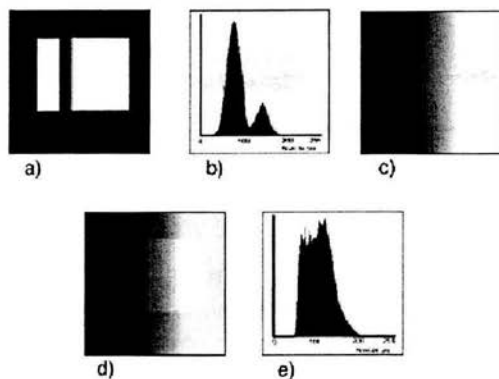


Figura 2.8: Problema de iluminación. a) Imagen. b) Histograma de 2.8.a. c) Función de iluminación. d) Imagen obtenida de la multiplicación de 2.8.a con 2.8.c. e) Histograma de 2.8.d.

2.4.3. Umbral global

Una técnica frecuentemente usada para la segmentación de imágenes consiste en dividir la escala de grises en conjuntos y usar umbrales para determinar regiones o para obtener puntos pertenecientes al contorno del objeto.

Suponga que los niveles de gris en una imagen dada, $f(x, y)$, tiene el histograma mostrado en la figura 2.9.a. Del histograma se concluye que hay un número grande de pixeles oscuros. Los pixeles restantes son distribuidos equitativamente en la parte restante de la escala de grises. El comportamiento de este histograma es característico de imágenes que contienen objetos grises sobrepuestos en un fondo oscuro. Para detectar el límite entre los objetos y el fondo, se divide el histograma en dos bandas separadas por un umbral U (ver figura 2.9.b). El problema ahora es, encontrar un U adecuado de tal manera que la banda B_1 contenga a los niveles de gris que corresponden a pixeles del fondo, y B_2 contenga los niveles de gris de los objetos. Para encontrar los contornos en $f(x, y)$, González [10] define un procedimiento que permite detectar límites en ambas direcciones, horizontal y vertical; el cual hace uso de imágenes intermedias. Dicho procedimiento se describe a continuación:

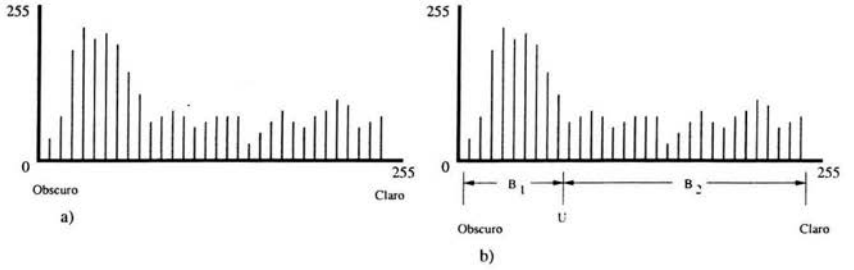


Figura 2.9: a) Histograma de una imagen. b) Histograma mostrado en 2.9.a particionado por dos bandas para determinar el umbral.

Paso 1. Para cada renglón en $f(x, y)$ (es decir, $x = 0, 1, \dots, n - 1$), crear un renglón correspondiente en una imagen intermedia $g_1(x, y)$, usando la siguiente relación para $y = 1, 2, \dots, n - 1$:

$$g_1(x, y) = \begin{cases} L_E & \text{Si los niveles de } f(x, y) \text{ y } f(x, y - 1) \text{ están} \\ & \text{en diferentes bandas de la escala de grises.} \\ L_B & \text{En otro caso.} \end{cases} \quad (2.22)$$

donde L_E y L_B son valores para bordes específicos y niveles del fondo, respectivamente.

Paso 2. Para cada columna en $f(x, y)$ (es decir, $y = 0, 1, \dots, n - 1$), crear un columna correspondiente en una imagen intermedia $g_2(x, y)$ usando la siguiente relación para $x = 1, 2, \dots, n - 1$:

$$g_2(x, y) = \begin{cases} L_E & \text{Si los niveles de } f(x, y) \text{ y } f(x - 1, y) \text{ están} \\ & \text{en diferentes bandas de la escala de grises.} \\ L_B & \text{En otro caso.} \end{cases} \quad (2.23)$$

La imagen deseada, consiste de los puntos límite entre los diferentes objetos y el fondo, es obtenida usando la siguiente relación para $x, y = 0, 1, \dots, n - 1$:

$$g(x, y) = \begin{cases} L_E & \text{Si } L_E \text{ es igual a } g_1(x, y) \text{ o a } g_2(x, y). \\ L_B & \text{En otro caso.} \end{cases} \quad (2.24)$$

Este procedimiento se puede generalizar para manejar más bandas pero hay que modificar las ecuaciones 2.22 y 2.23 que están definidas para el uso de una sola banda. Para ello, habría que agregar más valores con diferentes niveles de gris, dependiendo de la banda en la cual el cambio toma lugar.

Con el uso de más bandas en la aplicación de umbrales, aumenta la eficiencia de extracción de contornos. El problema sigue siendo la ubicación de los valores umbrales. Una forma de obtener estos valores es buscándolos por prueba y error. Este método es exitoso si el número de imágenes a ser procesadas es pequeño. Situaciones en donde el proceso es automático, el trabajo resulta poco confiable porque los valores umbrales no son los mejores debido a que se dan de modo invariante.

2.5. Segmentación por regiones

El objetivo de la segmentación es extraer objetos de una imagen. Los algoritmos ya mencionados en secciones anteriores realizan el proceso mediante la localización de contornos ayudándose de las discontinuidades de niveles de gris, y de la aplicación de umbrales. Una forma diferente de segmentar una imagen es por medio de la ubicación de regiones. Las imágenes se pueden ver como un conjunto de regiones o una región en si. En esta sección se hablará del proceso de segmentación basado en la localización de regiones pero antes se darán algunas consideraciones que se deben tomar en cuenta.

2.5.1. Consideraciones básicas

Considere una imagen como una región, R . La segmentación de la imagen es el proceso de dividir R en n subregiones, R_1, R_2, \dots, R_n , entonces se tiene que

1. $\bigcup_{i=1}^n R_i = R$
2. Cada R_i , con $i = 1, 2, \dots, n$, es una región conexa
3. $R_i \cap R_j = \emptyset$, para las i y j tales que $i \neq j$
4. $P(R_i) = Verdadero$, para $i = 1, 2, \dots, n$
5. $P(R_i \cup R_j) = Falso$ para $i \neq j$

donde $P(R_i)$ es un predicado lógico definido para los elementos del conjunto R_i .

La condición 1 dice que la segmentación debe ser completa, es decir, cada pixel de la imagen debe ser parte de una región. La condición 2 indica que las regiones son conexas. La condición 3 indica que las regiones no deben intersectarse. La condición 4 se refiere a que los pixeles que forman una región segmentada cumplen con una propiedad. Un simple ejemplo sería que $P(R_i) = Verdadero$ si todos los pixeles de R_i tienen el mismo valor de nivel de gris. Finalmente, la condición 5 indica que un predicado P no lo cumplen dos regiones diferentes al mismo tiempo.

2.5.2. Crecimiento de regiones agregando pixeles

La segmentación de imágenes mediante el crecimiento de regiones es un procedimiento que genera regiones agrupando pixeles y hace más grande una región agregándole pixeles. Inicialmente se toma una región llamada *semilla* formada de un pequeño conjunto de pixeles. Se hace crecer a la región agregando los pixeles vecinos, de los pixeles que forman parte de la semilla, los cuales tienen las mismas propiedades, por ejemplo, nivel de gris, color o textura. Para ejemplificar detalladamente este concepto se dará un ejemplo. La figura 2.10.a muestra una imagen representada por pixeles y su valor de nivel de gris para cada pixel. Los dos pixeles marcados con un círculo serán las semillas que generarán dos regiones, R_1 y R_2 . La propiedad P se utilizará para hacer crecer a las regiones que formarán cada una de las semillas, la cual se define como: la diferencia absoluta de los valores de nivel de gris entre la semilla y los demás pixeles tiene que ser menor que un valor umbral, en este caso $U = 10$. Si un pixel cumple simultáneamente con esa propiedad para ambas semillas se asigna a la región R_1 . La figura 2.10.b muestra el resultado de la obtención de las regiones donde los pixeles de la región R_1 son representados por una x y los pixeles de la región R_2 son representados por una y . Observe que si se toman otras semillas pero cada una de ellas forma parte de cada una de las regiones obtenidas el resultado debe ser el mismo. Y si se toma otro valor umbral, por ejemplo $U = 25$ se obtiene una sola región, como se muestra en la figura 2.10.c.

De lo anterior se puede ver que existen dos problemas: la selección de las semillas que propiamente representan regiones de interés, y la selección de propiedades satisfactorias para agregar pixeles a las regiones durante el proceso de crecimiento. Elegir una o más semillas depende del problema. Cuando no se conoce la forma en que debe ser la segmentación se usa cualquier pixel o conjunto de pixeles como semillas pero es preferible tomar en cuenta alguna consideración para tener un mejor resultado. En el ejemplo dado, note que los valores de las semillas son los

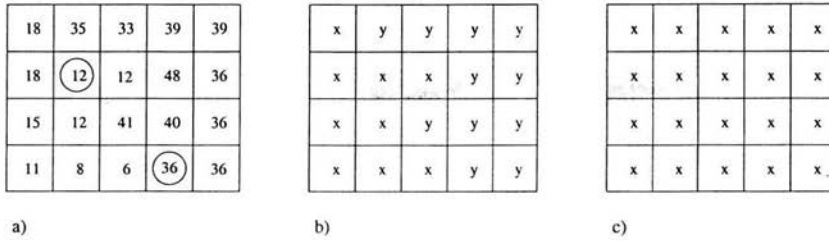


Figura 2.10: Aplicación del algoritmo de crecimiento de regiones. a) Imagen con dos pixeles marcados por círculos que son usados como semillas para el proceso de segmentación mediante el crecimiento de regiones. b) Imagen obtenida del proceso de segmentación por regiones crecientes aplicado a la imagen en 2.10.a, usando un valor umbral igual a 10. c) Imagen obtenida de la aplicación del proceso de segmentación por regiones crecientes a la imagen en 2.10.a, usando un valor umbral igual a 25.

más frecuentes en la imagen. La selección de una propiedad para agregar pixeles a una región sin saber como debe ser la segmentación es más compleja, lo ideal para escoger una propiedad conveniente es saber qué características similares deben cumplir los pixeles que formarán una región.

También hay que considerar que al realizar el proceso de crecimiento de regiones se pueden obtener regiones que no sean de interés, por ejemplo, localizar una región que sea parte del fondo de la imagen u obtener objetos que no se quieren analizar. Esto también es un problema de selección de semillas y propiedades que hacen crecer a las regiones.

Otro problema muy importante en el proceso de crecimiento de regiones es saber hasta que momento se hace crecer una región. Básicamente, el proceso termina cuando no hay más pixeles que satisfagan las propiedades para incluirlos en alguna región. Otros factores que se pueden tomar en cuenta para incrementar el poder del algoritmo de crecimiento de regiones son: considerar el tamaño, la semejanza entre un posible pixel que se va a agregar y uno que forma parte de la región creciente (por ejemplo, el valor del nivel de gris entre el pixel a agregar y el promedio de los valores de nivel de gris de los pixeles que forman a la región), y la forma de una región dada para empezar a crecer.

2.5.3. División y unión de regiones

Un procedimiento que se realiza de forma contraria al procedimiento de la sección anterior es tomar a la imagen completa como una región e ir dividiendo para obtener regiones más pequeñas y unir las de tal manera que satisfacen las condiciones dadas en la sección 2.5.1. El algoritmo que divide y une regiones trabaja iterativamente y se describe a continuación:

Sea R la región formada por toda la imagen y P un predicado que deben satisfacer las regiones. Generalmente una imagen tiene forma de rectángulo, y la segmentación de R consiste en dividirla sucesivamente en pequeños rectángulos y estos a su vez en rectángulos más pequeños de tal forma que, para cualquier región R_i , $P(R_i) = Verdadero$. Dicho de otra manera, si $P(R) = Falso$, se divide la imagen en cuadrantes. Si P es *Falso* para cualquier cuadrante, se subdivide en subcuadrantes, y así sucesivamente.

Hay que observar que este proceso de división tiene una conveniente representación en forma de *quadtree*³. La raíz del árbol corresponde a la imagen completa y cada nodo corresponde a una subdivisión. El árbol no es necesariamente balanceado.

Al final de la división de la imagen en cuadrantes, es posible que haya regiones adyacentes con las mismas propiedades. Para ello se usa la unión de regiones, esto es, se unen dos regiones adyacentes solamente si cumplen con las condiciones dadas en la sección 2.5.1., es decir, si R_i y R_j son adyacentes y además $P(R_i \cup R_j) = Verdadero$ entonces se unen las regiones.

El algoritmo de segmentación mediante la división y unión de regiones desarrollado por H. Samet [16] se describe mediante los siguientes pasos:

1. Cualquier región R_i tal que $P(R_i) = Falso$, dividirla en cuatro cuadrantes.
2. Unir cualesquiera dos regiones R_i y R_j , que sean adyacentes con $P(R_i \cup R_j) = Verdadero$.
3. Terminar cuando ya no sea posible dividir o unir.

Existen diferentes variantes sobre el algoritmo dado. Por ejemplo, se puede iniciar el algoritmo con una imagen que ya esté dividida en regiones. Las regiones pueden ser del mismo o diferente tamaño. Si las regiones son de diferente tamaño, durante el proceso del algoritmo también se pueden obtener subdivisiones de diferentes tamaños. Un posible inconveniente para este algoritmo (ya sea con las regiones del mismo tamaño o no), es que las divisiones siempre dan como resultado cuatro cuadrantes lo cual puede ocasionar que sea un proceso lento.

³árbol en el cual cada nodo tiene exactamente cuatro descendientes.

2.6. Morfología matemática

En esta sección se describen algunas de las operaciones lógicas que pueden ayudar a la segmentación de imágenes. A la aplicación de estas operaciones se le conoce como *morfología matemática* [7]. Aunque las operaciones básicas son sencillas, ellas y sus variantes pueden ser combinadas para producir efectos más complejos.

En general, la morfología matemática consiste en la aplicación de una operación lógica entre una imagen binaria y un *elemento estructural*. El elemento estructural es una imagen binaria predefinida que puede ser de cualquier tamaño. La aplicación de la operación se realiza tomando en cuenta las posiciones de cada uno de los píxeles de la imagen y las posiciones de los píxeles del elemento estructural. Este proceso da como resultado otra imagen binaria que se construye a partir de las posiciones resultantes de la aplicación de la operación. El efecto creado depende del tamaño y el contenido del elemento estructural y del tipo de operación lógica.

Como notación, se usará a B para el conjunto de posiciones de los píxeles que forman a la imagen binaria, es decir, B es el conjunto de las coordenadas que determinan la localización de un 1 dentro de la imagen. Se denotará con S al conjunto de posiciones de los píxeles que forman al elemento estructural.

Las operaciones morfológicas básicas son dos: *erosión* y *dilatación*. La erosión adelgaza a la imagen y la dilatación la hace más gruesa. A continuación, se definen la erosión, la dilatación y dos de las operaciones derivadas de ellas.

2.6.1. Erosión

La erosión es el proceso que elimina todos los píxeles que forman parte del contorno de un objeto, obteniendo un objeto con un área más pequeña pero con la misma forma. Si el objeto es circular, su diámetro es decrementado en dos píxeles con cada erosión. Si el ancho del objeto consta de tres píxeles en algún punto, al aplicar la erosión se obtiene un objeto con regiones desconexas, o lo que es lo mismo, se obtienen dos o más objetos. Los objetos que tienen una o más regiones con uno o dos píxeles de ancho se eliminan. La erosión es usada para eliminar de una imagen segmentada los objetos que son pequeños y de poco interés.

La erosión es definida por:

$$E = B \ominus S = \{(x, y) | \forall (x_2, y_2) \in S, \exists (x_1, y_1) \in B : (x, y) = (x_1, y_1) - (x_2, y_2)\} \quad (2.25)$$

donde E es la imagen resultante de erosionar a B con S ; (x, y) , (x_1, y_1) y (x_2, y_2) son las posiciones de cada pixel de E , B y S respectivamente.

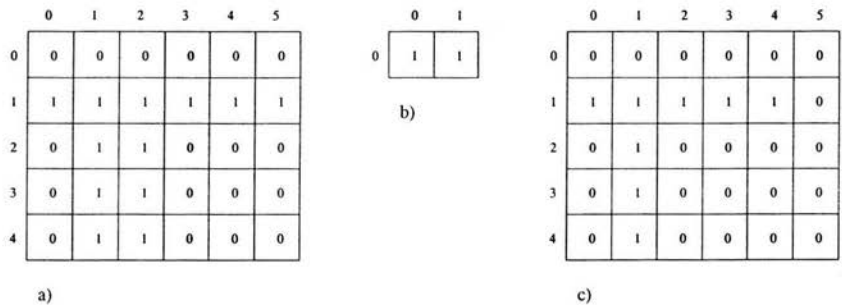


Figura 2.11: Aplicación de la operación morfológica, erosión. a) Una imagen binaria. b) El elemento estructural. c) Imagen resultante de la aplicación de la erosión entre 2.11.a y 2.11.b.

La ecuación 2.25 quiere decir que las posiciones de los píxeles que forman a E son las que corresponden a la diferencia entre la posición de los píxeles que forman a B y cada una de las posiciones de los píxeles que forman a S . Para aclarar esta definición se verá un ejemplo. En la figura 2.11 se muestra una imagen binaria (figura 2.11.a), el elemento estructural (figura 2.11.b) y la imagen resultante de erosionarlas (figura 2.11.c). Los elementos del conjunto B son $\{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (2,1), (2,2), (3,1), (3,2), (4,1), (4,2)\}$ y los del conjunto S son $\{(0,0), (0,1)\}$. La siguiente lista muestra el cálculo para obtener cada uno de los elementos del conjunto E , los cuales están subrayados.

$$\begin{array}{ll}
 (1,0) - (0,0) = \underline{(1,0)} & (1,0) - (0,1) = (1,-1) \\
 (1,1) - (0,0) = \underline{(1,1)} & (1,1) - (0,1) = \underline{(1,0)} \\
 (1,2) - (0,0) = \underline{(1,2)} & (1,2) - (0,1) = \underline{(1,1)} \\
 (1,3) - (0,0) = \underline{(1,3)} & (1,3) - (0,1) = \underline{(1,2)} \\
 (1,4) - (0,0) = \underline{(1,4)} & (1,4) - (0,1) = \underline{(1,3)} \\
 (1,5) - (0,0) = \underline{(1,5)} & (1,5) - (0,1) = \underline{(1,4)} \\
 \\ \\
 (2,1) - (0,0) = \underline{(2,1)} & (2,1) - (0,1) = (2,0) \\
 (2,2) - (0,0) = \underline{(2,2)} & (2,2) - (0,1) = \underline{(2,1)} \\
 (3,1) - (0,0) = \underline{(3,1)} & (3,1) - (0,1) = \underline{(3,0)} \\
 (3,2) - (0,0) = \underline{(3,2)} & (3,2) - (0,1) = \underline{(3,1)} \\
 (4,1) - (0,0) = \underline{(4,1)} & (4,1) - (0,1) = \underline{(4,0)} \\
 (4,2) - (0,0) = \underline{(4,2)} & (4,2) - (0,1) = \underline{(4,1)}
 \end{array}$$

Note que las coordenadas subrayadas son las que aparecen en ambas columnas (una columna por cada elemento de S). Y note que las posiciones obtenidas que están fuera del rango determinado por el tamaño de la imagen son ignoradas, por ejemplo: $(1,-1)$.

2.6.2. Dilatación

La dilatación es el proceso que incorpora pixeles alrededor de todo el contorno del objeto, dándole un área mayor. Si el objeto es circular, su diámetro se incrementa en dos pixeles con cada dilatación. Si dos objetos son separados por menos de tres pixeles en cualquier punto, se obtiene un objeto conexo (se unen los objetos) en ese punto. La dilatación es usada para el llenado de hoyos en objetos segmentados.

La dilatación es definida por:

$$D = B \oplus S = \{(x, y) | \exists(x_1, y_1) \in B, \exists(x_2, y_2) \in S : (x, y) = (x_1, y_1) + (x_2, y_2)\} \quad (2.26)$$

donde D es la imagen resultante de dilatar a B con S ; (x, y) , (x_1, y_1) y (x_2, y_2) son las posiciones de cada pixel de D , B y S respectivamente. Las posiciones de los pixeles que forman a D se obtienen con la suma de las posiciones de los pixeles que forman a B y a S .

Para ilustrar esta definición, en la figura 2.12 se muestra un ejemplo. En la figura 2.12.a se muestra la imagen binaria, en la figura 2.12.b se muestra el elemento estructural y en la figura 2.12.c se muestra la imagen resultante de la aplicación de la dilatación entre 2.12.a y 2.12.b. La obtención de los pixeles que forman a D se muestran a continuación.

$(0,1) + (0,0) = (0,1)$	$(0,1) + (0,1) = (0,2)$	$(0,1) + (1,0) = (1,1)$
$(1,1) + (0,0) = (1,1)$	$(1,1) + (0,1) = (1,2)$	$(1,1) + (1,0) = (2,1)$
$(2,1) + (0,0) = (2,1)$	$(2,1) + (0,1) = (2,2)$	$(2,1) + (1,0) = (3,1)$
$(3,1) + (0,0) = (3,1)$	$(3,1) + (0,1) = (3,2)$	$(3,1) + (1,0) = (4,1)$
$(3,2) + (0,0) = (3,2)$	$(3,2) + (0,1) = (3,3)$	$(3,2) + (1,0) = (4,2)$
$(4,0) + (0,0) = (4,0)$	$(4,0) + (0,1) = (4,1)$	$(4,0) + (1,0) = (5,0)$

Los elementos de B son $\{(0,1),(1,1),(2,1),(3,1),(3,2),(4,1)\}$, y los elementos de S son $\{(0,0),(0,1),(1,0)\}$. Todas las coordenadas resultantes de las sumas corresponden a las posiciones de los pixeles de D . Las coordenadas fuera del rango del tamaño de la imagen son ignoradas.

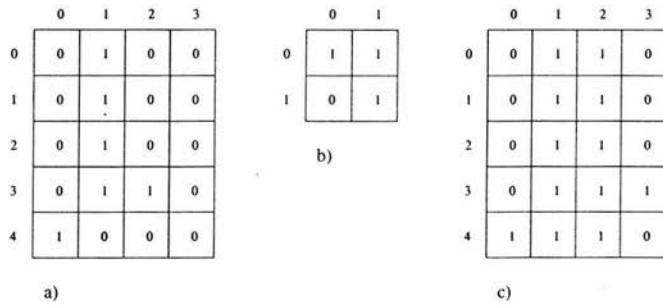


Figura 2.12: Aplicación de la dilatación. a) Imagen Binaria. b) Elemento estructural. c) Imagen obtenida de la aplicación de la dilatación entre 2.12.a y 2.12.b.

2.6.3. Opening

Una combinación entre erosión y dilatación, es decir, aplicar erosión y después dilatación a una imagen binaria da como resultado una operación conocida como *opening*. Esta operación produce el efecto de eliminar pequeños y delgados objetos, descomponer objetos en puntos delgados, y generalmente alisa el contorno de objetos grandes sin modificar significativamente su tamaño.

La operación opening se define como:

$$B \circ S = (B \ominus S) \oplus S \quad (2.27)$$

2.6.4. Closing

El proceso que realiza dilatación seguido de erosión es llamado *closing*. Esta operación tiene el efecto de rellenar pequeños y delgados hoyos, unir objetos cercanos, y generalmente alisa el contorno de objetos sin cambios significativos en su tamaño.

La operación closing es definida por:

$$B \bullet S = (B \oplus S) \ominus S \quad (2.28)$$

Frecuentemente, en la segmentación de imágenes mediante umbrales aparecen porciones de ruido, dando como resultado contornos totalmente desiguales, los objetos presentan hoyos falsos, y en el fondo de la imagen aparecen pequeños objetos de ruido. Sucesivas aplicaciones de las operaciones de opening o closing pueden eliminar el ruido en algunas situaciones. Además, aplicar varias iteraciones de erosión

seguidas por la misma cantidad de iteraciones de dilatación produce el efecto deseado.

Capítulo 3

Medidas de Similitud de Formas 2D

3.1. Introducción

Uno de los temas más estudiados en el área de Reconocimiento de Patrones son las medidas de similitud, tanto para formas en dos dimensiones (2D), formas planas, como para formas en tres dimensiones (3D), formas con volumen. Desde hace varios años se ha intentado encontrar una técnica eficiente que permita medir la similitud entre formas; estos años de intentos han llevado al desarrollo e implementación de una gran cantidad de métodos (ver [1], [4], [5], [6] y [9]).

En el capítulo anterior se explicaron varios procedimientos para obtener los objetos (formas) que son parte de una imagen. En este capítulo se presentan algunas técnicas que permiten medir la similitud entre esos objetos, en particular, objetos en dos dimensiones.

Antes de comparar a las formas, ellas deben cumplir con algunas condiciones en cuanto a tamaño, posición y orientación, con el objeto de una mayor precisión en la comparación. En la sección 3.2 se toca el tema de invariantes, traslación, para ubicar el centro de masa; rotación usando los ejes principales; y escala, para hacer constante el tamaño. También se expone sobre la importancia del uso de los invariantes en la comparación de formas.

En la sección 3.3 se expone uno de los primeros métodos que se desarrollaron, los números de forma. Con los números de forma se sabe cómo son las formas porque describen contornos mediante un código numérico. La comparación de formas usando los números de forma es a través de la generación de su árbol de similaridad. En la sección 3.4 se toca el tema de correlación máxima. La comparación con este

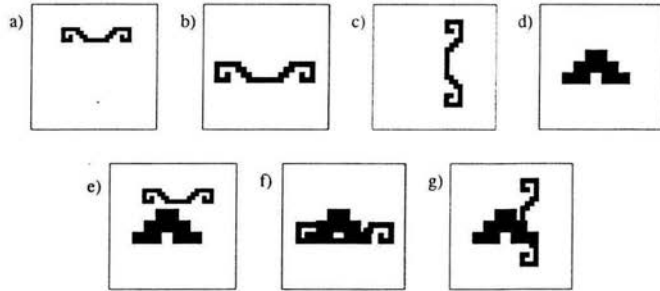


Figura 3.1: En las figuras 3.1.a, 3.1.b y 3.1.c se muestran la misma forma pero con diferente tamaño, posición y orientación. En la figura 3.1.d se muestra la otra imagen con la que se comparan. En las figuras 3.1.e, 3.1.f y 3.1.g se muestra la comparación mediante una sobreposición de imágenes.

método es por medio de un emparejamiento o sobreposición de formas en el que se maximiza dicho emparejamiento. En la sección 3.5 se presenta el método de transformación de un objeto a otro. Este método usa conceptos importantes sobre teoría de gráficas y la finalidad de él es convertir una forma en la otra a través de un desplazamiento de píxeles. Aquí se podrá ver el parecido entre formas dado que el método proporciona un costo de transformación: a más pequeño el costo mayor similitud. Y finalmente en la sección 3.6 se expone el tema de la compacidad de forma. En esta sección se describen los diferentes tipos de compacidad y se analiza la comparación de formas que usa compacidad; de este análisis se concluye que la compacidad evalúa qué tan gruesas o delgadas son las formas. El uso de la compacidad puede ser un proceso previo a métodos que proporcionan medidas de similitud.

3.2. Invariantes

Tomar dos formas y compararlas tal y como se presentan no siempre es la mejor opción, debido a que la posición de dichas formas puede ubicarse en cualquier lugar de la imagen y puede no ser adecuada. Por ejemplo, si se quiere comparar mediante un emparejamiento, lo que se hace es sobreponer una imagen en la otra pero esto no es la mejor medida de similitud. En la figura 3.1 se muestra tres veces la misma forma pero en diferente posición dentro de la imagen y cómo sería su comparación

con otra.

Para solucionar el problema se aplican *invariantes*. Un invariante, en este caso, es una operación cuya aplicación conserva la forma del objeto. Los invariantes que se expondrán aquí son: traslación, rotación y escala. Dichas operaciones permitirán manipular las imágenes sin perder su forma, ayudando a realizar posteriormente una mejor comparación de las mismas.

3.2.1. Traslación

Tomando como referencia el plano en 2 dimensiones las ecuaciones para trasladar un objeto ubicado en el plano son:

$$x' = x + x_0 \quad (3.1)$$

$$y' = y + y_0 \quad (3.2)$$

De manera análoga se puede considerar a la imagen como un espacio en 2 dimensiones, y para trasladar la forma se utilizan las mismas ecuaciones. Donde (x', y') son las coordenadas del nuevo punto, (x, y) son las del punto a trasladar, y (x_0, y_0) es la distancia por trasladar (un valor constante).

La traslación se realiza para obtener invariancia bajo traslación, la cual se obtiene con el *centro de masa*.

El **centro de masa**, en términos no formales, es la posición en la imagen donde se concentra más información, es decir, es el centro de la forma y no el centro de toda la imagen. Las ecuaciones para encontrar el centro de masa de una imagen binaria son las siguientes:

$$x_{cm} = \frac{\sum(x_i * n_i)}{N} \quad (3.3)$$

$$y_{cm} = \frac{\sum(y_i * n_i)}{N} \quad (3.4)$$

donde (x_{cm}, y_{cm}) son las coordenadas del centro de masa, (x_i, y_i) son las coordenadas de cada punto en la imagen, n_i es el nivel de gris en cada punto, y N es el número total de píxeles con nivel de gris igual a 1.

3.2.2. Rotación

La invariancia en rotación se obtiene con el uso de los *ejes principales*. Estos permiten determinar cuál es la parte más ancha o delgada de la forma. Obtener esa

información sirve para poder hacer mejor la comparación entre las formas porque un apareo es más adecuado si la orientación de las formas es la misma.

Los **ejes principales** se encuentran a una distancia angular de un sistema rectangular de coordenadas (x, y) . La distancia se determina con las siguientes ecuaciones:

$$\varphi_1 = \arctan \frac{I_{12}}{\lambda_1 - I_{22}} \quad (3.5)$$

$$\varphi_2 = \arctan \frac{I_{12}}{\lambda_2 - I_{22}} \quad (3.6)$$

donde φ_1 y φ_2 son los ángulos y cumplen con:

$$|\varphi_1 - \varphi_2| = \frac{\pi}{2} \quad (3.7)$$

λ_1 y λ_2 son las raíces de la ecuación:

$$\lambda^2 - \lambda(I_{11} + I_{22}) + I_{11}I_{22} - I_{12}^2 = 0 \quad (3.8)$$

mientras que I_{11} , I_{22} , I_{12} son los componentes del tensor de inercia I (ver [3]),

$$I = \begin{vmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{vmatrix} \quad (3.9)$$

cuyo valor se calcula mediante:

$$I_{11} = \sum_{k=1}^m n_k (y_k)^2 \quad (3.10)$$

$$I_{22} = \sum_{k=1}^m n_k (x_k)^2 \quad (3.11)$$

$$I_{12} = I_{21} = - \sum_{k=1}^m n_k (x_k)(y_k) \quad (3.12)$$

donde n_k es el nivel de gris en cada punto de la imagen determinado por la coordenada (x_k, y_k) .

Para que los ejes principales de dos formas coincidan hay que aplicar la función de rotación a una de ellas. Esto se hace mediante el siguiente cálculo:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.13)$$

El valor del ángulo θ se obtiene de la distancia angular entre los ejes principales de ambas formas.

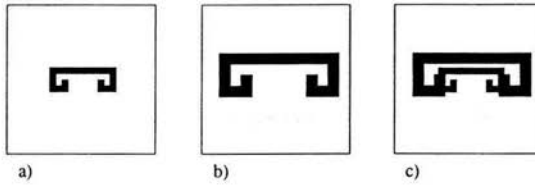


Figura 3.2: a) y b) dos imágenes con la misma forma. c) Comparación de 3.2.a y 3.2.b.

3.2.3. Escala

Es posible que la comparación visual de dos imágenes que contienen la misma forma indique que son iguales y que una comparación mediante emparejamiento indique lo contrario. La razón de este resultado puede atribuirse no sólo a diferencias de posición u orientación, sino a diferencias de tamaño (vea figura 3.2).

Como se puede apreciar en la figura 3.2, el emparejamiento no es exacto y es porque las imágenes tienen diferente tamaño a pesar de que tienen la misma forma.

El tamaño es uno de los principales motivos por el cual conviene tener invariancia bajo escala. Esto se obtiene cambiando el número de píxeles de manera que ambas imágenes tengan la misma cantidad.

3.3. Números de forma

Uno de los métodos de comparación de formas estudiados en el área de Reconocimiento de Patrones y proporcionado por E. Bribiesca y A. Guzmán [5] es el método de los *números de forma*. Este método representa el contorno de las formas mediante un código numérico; y realiza la comparación de las mismas mediante los códigos respectivos.

Si dos imágenes diferentes a simple vista tienen formas diferentes, entonces los códigos de estas imágenes, proporcionados por el método de los números de forma, también deben ser diferentes. Sin embargo, si las imágenes fueran rotadas resultaría que sus códigos no permanecen iguales; para evitar esto, se establecerán condiciones especiales.

Como antecedente a los números de forma están las *cadenas de Freeman* [9]. Las cadenas al igual que los números de forma describen la forma de la imagen. Primero se definirán las cadenas de Freeman y posteriormente los números de forma.

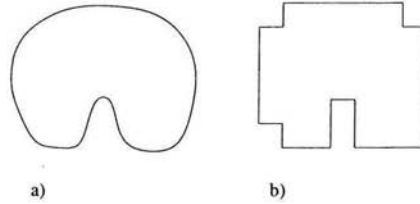


Figura 3.3: a) Una forma. b) El contorno pixelizado de la imagen mostrada en 3.3.a.

Para obtener las cadenas de Freeman y los números de forma se necesita que la imagen esté pixelizada (ver apéndice), es decir, que la imagen esté en forma discreta y no continua. Ver figura 3.3.

3.3.1. Cadenas de Freeman

Las cadenas de Freeman describen la forma de la imagen mediante un recorrido por el contorno. Este recorrido se hace en el sentido que giran las manecillas del reloj, y por cada paso del recorrido se asigna un número de código.

Freeman desarrolló dos tipos de cadenas, de ocho y de cuatro direcciones. Cada una de éstas indican el movimiento que se debe realizar para hacer el recorrido. Los movimientos para la cadena de ocho direcciones son (vea figura 3.4.b): arriba, abajo, izquierda, derecha, diagonal derecha superior, diagonal derecha inferior, diagonal izquierda superior y diagonal izquierda inferior; y para la cadena de cuatro direcciones sólo son arriba, abajo, izquierda y derecha. Para la cadena de ocho direcciones cada movimiento está representado por un número del 0 al 7 (figura 3.4.b), y para la cadena de cuatro direcciones con un número del 1 al 4 (figura 3.4.e).

La figura 3.4 muestra cómo es el recorrido aplicado a la imagen de la figura 3.3.b usando la cadena de ocho direcciones (figura 3.4.a) y la cadena de cuatro direcciones (figura 3.4.d). También se muestran los códigos de los recorridos y sus respectivas direcciones.

3.3.2. Descripción de los Números de forma

Los números de forma al igual que las cadenas de Freeman describen el contorno de formas mediante un código numérico. Para los números de forma el código utilizado consta de los números 1, 2 y 3. La descripción también se hace con un recorrido de contorno en dirección al giro de las manecillas de reloj.

El contorno de una forma pixelizada consta básicamente de líneas continuas y esquinas. Las esquinas se pueden ver de dos tipos: esquinas hacia afuera y esquinas hacia adentro de la imagen. Cada uno de estos elementos es representado por el código de los números de forma. El número 1 representa esquinas convexas, el número 2 representa esquinas (líneas) continuas y el número 3 representa esquinas cóncavas. En la figura 3.5 se puede apreciar el código de los números de forma aplicado a la figura 3.3.b. Este código es la diferencial del código de Freeman, que resulta invariante bajo rotación. Es decir, si la forma es rotada su código, dado por los números de forma, se conserva.

Observe que para una forma se pueden obtener diferentes códigos, un código por cada posición donde se inicie el recorrido del contorno. Por ejemplo, para la figura 3.5 se tienen las siguientes cadenas de código:

1. 122221312222122123321213122213
2. 222213122221221233212131222131
3. 222131222212212332121312221312
4. 221312222122123321213122213122
5. 213122221221233212131222131222
6. 131222212212332121312221312222
7. 312222122123321213122213122221
8. 122221221233212131222131222213
9. 222212212332121312221312222131
10. 222122123321213122213122221312
11. 221221233212131222131222213122
12. 212212332121312221312222131222
13. 122123321213122213122221312222
14. 221233212131222131222213122221
15. 212332121312221312222131222212
16. 123321213122213122221312222122
17. 233212131222131222213122221221
18. 332121312221312222131222212212
19. 321213122213122221312222122123
20. 212131222131222213122221221233
21. 121312221312222131222212212332
22. 213122213122221312222122123321
23. 131222131222213122221221233212
24. 312221312222131222212212332121
25. 122213122221312222122123321213
26. 222131222213122221221233212131

27. 221312222131222212212332121312
28. 213122221312222122123321213122
29. 131222213122221221233212131222
30. 312222131222212212332121312221
31. 122221312222122123321213122213
32. 222213122221221233212131222131
33. 222131222212212332121312221312

Note que las cadena 1 y 31 son iguales, también la 2 y 32, y la 3 y 33. Además, en esta lista están todas las cadenas que representan a la forma.

La razón de que existan diferentes cadenas que representen a una sola forma se debe a que el recorrido del contorno se puede iniciar desde cualquier punto de la forma. En el caso del ejemplo, el inicio del recorrido está señalado con una flecha. Lo mismo sucede con las cadenas de Freeman, el código depende del punto inicial del recorrido.

De todas las cadenas que representan a una forma, una de ellas es la mínima (en orden lexicográfico). La cadena mínima es la que de izquierda a derecha tiene los números de valor más pequeño. En otras palabras, es la cadena más chica en orden lexicográfico con alfabeto $\{1, 2, 3\}$. Para el ejemplo, la cadena mínima es la número 21 (121312221312222131222212212332).

Antes de obtener el código del contorno de una forma mediante los números de forma es necesario establecer la orientación de la forma. Esto se debe a que la pixelización no siempre produce el mismo contorno pixelizado para una forma con diferentes orientaciones (figura 3.6). La manera de resolver este problema es haciendo que las formas siempre tengan la misma orientación al momento de pixelizarlas. Para ello se usa la invariancia en rotación, es decir, los ejes principales de la forma siempre deben tener la misma dirección, y esta dirección debe corresponder a la de los ejes coordenados. Por lo tanto, primero hay que tener invariancia en rotación y después se encuentra el código del contorno dado por los números de forma.

3.3.3. Comparación de formas

En el proceso de pixelización de formas se requiere que el tamaño de la cuadrícula sea el mismo, el cual determina la resolución de las formas, es decir, mientras más pequeña sea la cuadrícula las formas se aprecian mejor. En un árbol de resoluciones a cada resolución de cuadrícula le corresponde un nivel; cada forma puede ser localizada en algún nivel. El árbol de resoluciones se puede utilizar para comparar formas: si son iguales, el camino para llegar a ellas a su posición en el árbol es el mismo; en cambio, si son diferentes sus caminos también lo son.

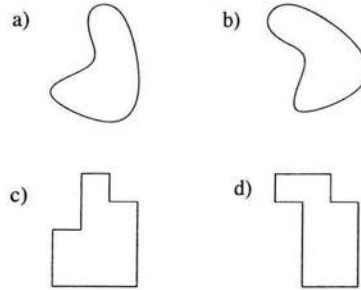


Figura 3.6: a) Una forma. b) La misma forma de 3.6.a pero rotada 45 grados. c) La forma pixelizada de 3.6.a. d) La forma pixelizada de 3.6.b. Note que 3.6.c y 3.6.d no son iguales.

Para comparar dos formas se necesita que sus respectivos números de forma tengan la misma cantidad de dígitos. A este valor se le conoce como *orden del número de forma*. Note que el orden de cualquier número de forma siempre es par debido a que el contorno de una forma es una curva cerrada [5].

Un método para obtener el número de forma de orden n es el siguiente:

1. Hacer invariante en rotación a la forma.
2. Encontrar el rectángulo mínimo que contiene a la forma.
3. Cuadricular el rectángulo de manera conveniente para obtener el número de forma de orden n .

El paso 3 de este método es semejante a la pixelización pero con una diferencia: con este método se tiene que escoger apropiadamente el tamaño de la cuadrícula para que la forma resulte con un número de forma de orden n .

Obtener un número de forma de un orden específico puede no ser sencillo; el código producido, en un primer intento, puede no ser el deseado debido a la profundidad de las concavidades del contorno. En este caso, el proceso para encontrarlo se repite, aplicado con otro tamaño de cuadrícula. El número de forma puede no existir en casos especiales tales como contornos con indentaciones estrechas. A pesar de esto, en la mayoría de los casos, un número de forma puede ser calculado.

El grado de similitud entre dos formas es el orden más grande para el cual los números de forma respectivos son iguales. Esto es, para ambas formas se calcula cada uno de los números de forma de orden i , para $i = 4, 6, 8, 10, \dots$, y si los números

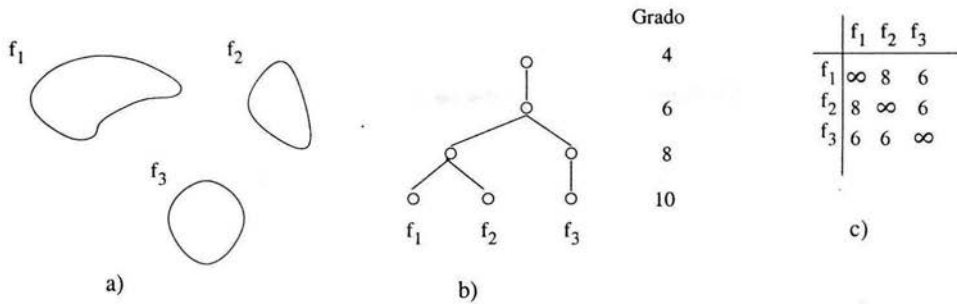


Figura 3.7: Comparación de tres formas usando números de forma. a) Tres formas a comparar. b) Árbol de similitud de las formas. c) Grados de similitud de las formas.

de forma de ambas coinciden hasta $i = k$ pero difieren en $i = k + 2$ entonces su grado de similitud es k . Si las formas a comparar son idénticas sus números de forma de orden n siempre son iguales para toda $n = i + 2$, con $i = 2, 4, 6, 8, 10, \dots$; si las formas son muy similares sus números de forma son iguales hasta un orden n con un valor grande para n , por ejemplo $n = 200$; y si las formas son visiblemente diferentes el orden de sus números de forma serán iguales para valores pequeños de n , por ejemplo $n = 10$.

En la figura 3.7 se muestra un ejemplo de comparación de tres formas (figura 3.7.a), su árbol de similitud (figura 3.7.b) y sus grados de similitud (figura 3.7.c).

3.4. Correlación máxima

La *correlación máxima* es un método de comparación que se basa en maximizar el número de píxeles comunes de las formas a comparar. La manera en que se lleva a cabo es mediante la sobreposición y para ello se requiere que las formas tengan la misma área, es decir, que sean invariantes bajo escala (ver sección 3.2.3).

El método consiste en encontrar todas las posibles posiciones de emparejamiento entre las formas, para ello, se fija la orientación y posición de una de ellas. Posteriormente, la otra forma se sobrepone en la primera, tomando diferentes orientaciones y posiciones, adquiridas con las operaciones de traslación y rotación. Dichas operaciones se aplican con las ecuaciones 3.1, 3.2 y 3.13 respectivamente. En la figura 3.8

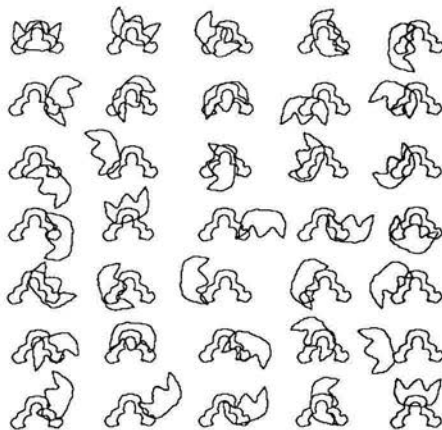


Figura 3.8: Algunas sobreposiciones de dos formas para encontrar la máxima correlación.

se muestra este proceso.

De todos los emparejamientos posibles, uno de ellos es el que tiene el máximo número de píxeles en común, este emparejamiento proporciona la correlación máxima. La manera de encontrar dicho emparejamiento es aplicar el *factor de emparejamiento* a cada uno de ellos:

$$\epsilon = \frac{Z}{Z + X} \quad (3.14)$$

donde X es el área de la intersección de las formas y Z es el área de la unión menos la intersección. Ver figura 3.9.

El valor más pequeño del factor de emparejamiento encontrado es el que determina la posición de las formas con la correlación máxima. Este cálculo determina eficazmente la correlación máxima, que a simple vista o con un conteo manual podría ser incorrecto o imposible de encontrar.

En la búsqueda de la correlación máxima hay que tomar en cuenta algunas consideraciones importantes. Se puede encontrar un mismo valor mínimo para dos o más sobreposiciones diferentes de las formas, esto no afecta si en la comparación lo que importa es el valor del factor de emparejamiento, pero puede afectar para

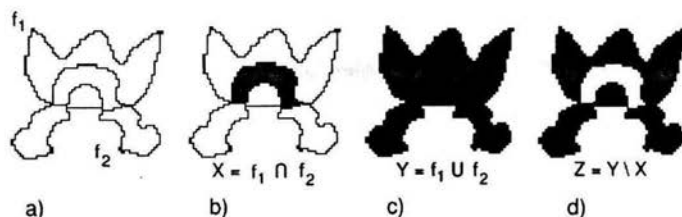


Figura 3.9: Proceso para encontrar la máxima correlación. a) Dos formas, f_1 y f_2 superpuestas. b) $f_1 \cap f_2$. c) $f_1 \cup f_2$. d) $(f_1 \cup f_2) \setminus (f_1 \cap f_2)$.

determinar la superposición única de las formas. También, si se usa correlación máxima para comparar tres o más formas y determinar cuáles dos son las más parecidas, se puede hallar el mismo factor de emparejamiento para dos o más parejas de formas diferentes, obteniéndose un resultado no único. O si se quiere transformar una forma a otra (ver sección 3.5) utilizando correlación máxima como invariante, los resultados pueden ser inapropiados.

El método de comparación mediante la transformación de un objeto a otro usa como invariantes a los ejes principales, el centro de masa y el área, es decir, normaliza a las formas en términos de estos; y posteriormente superpone a las formas. Esta superposición también es una correlación pero no necesariamente es la máxima. Cualquier emparejamiento de formas es una correlación la cual puede no ser máxima.

3.5. Transformación de un objeto a otro

Uno de los métodos desarrollados para comparar dos o más objetos es la *transformación de un objeto a otro*. Con este método se mide el costo de transformación y dependiendo de ese costo se determina que tan parecidas o diferentes son las formas.

La realización de este método requiere del uso de gráficas, por eso en la siguiente sección se proporcionan algunos conceptos de este tema.

3.5.1. Gráficas

En el capítulo 2, en la sección 2.3.2, se proporcionan las definiciones de: gráfica, gráfica dirigida, gráfica pesada, camino, peso de un camino. Tomando esas definiciones, a continuación se agregan otras que complementan el conjunto de definiciones

que se necesitarán para la realización del método que se verá en esta sección.

Definición 3.1 Un apareo M en una gráfica $G=(N,A)$ es un subconjunto de aristas de A tal que, para cada $n_i \in N$ existe a lo más un $n_j \in N$ tal que $(n_i, n_j) \in M$.

Definición 3.2 Un nodo $n_i \in N$ en una gráfica $G=(N,A)$ es saturado si existe $n_j \in N$ tal que $(n_i, n_j) \in M$, donde M es un apareo en G .

Definición 3.3 Un apareo M en una gráfica $G=(N,A)$ es perfecto si todo $n_i \in N$ es saturado.

Definición 3.4 Un apareo perfecto M en una gráfica $G=(N,A)$ es óptimo si G es pesada y $\sum_{(n_i, n_j) \in M} w(n_i, n_j)$ es menor o igual a $\sum_{(n_i, n_j) \in H} w(n_i, n_j)$ para todo H apareo perfecto en G .

Definición 3.5 Un camino n_1, n_2, \dots, n_m con $n_i \in N$, $i = 1, 2, \dots, m$, en una gráfica $G=(N,A)$ es alternante si $(n_i, n_{i+1}) \in M$ y $(n_{i+1}, n_{i+2}) \notin M$ o viceversa, para $i = 1, 3, 5, \dots, m-2$, donde M es un apareo en G .

Definición 3.6 Un camino n_1, n_2, \dots, n_m con $n_i \in N$, $i = 1, 2, \dots, m$, en una gráfica $G=(N,A)$ es extendido si es un camino alternante y $(n_1, n_2) \notin M$ y $(n_{m-1}, n_m) \notin M$, donde M es un apareo en G .

Definición 3.7 Una gráfica $G=(N,A)$ es bipartita si existe una partición del conjunto N en dos conjuntos, S y T , tal que si $(s, t) \in A$ entonces $s \in S$ y $t \in T$ o $s \in T$ y $t \in S$.

Definición 3.8 Una gráfica bipartita $G=(N,A)$ con partición S y T es completa si para toda $s \in S$ y para toda $t \in T$ existe $(s, t) \in A$.

3.5.2. Proceso de transformación

Para realizar el proceso de transformación de un objeto a otro se requiere: que las formas a transformar estén pixelizadas; que tengan la misma área; que su orientación sea la misma y que sus centros de masa coincidan, es decir, que sean invariantes bajo escala, rotación y traslación (ver sección 3.2). El hecho de que cumplan con estas condiciones garantiza obtener una mejor medida de similitud.

El proceso de transformación de un objeto A a un objeto B consiste en:

1. Sobreponer el objeto A con el objeto B .

2. Encontrar los píxeles P_p tales que $P_p = A \setminus B^1$. Estos píxeles son llamados *píxeles positivos*.
3. Encontrar los píxeles P_n tales que $P_n = B \setminus A$. Estos píxeles son llamados *píxeles negativos*.
4. Mover los píxeles positivos al lugar de los píxeles negativos, de tal manera que el total de las distancias recorridas por los píxeles positivos sea la menor.

Hay muchas maneras de mover los píxeles positivos al lugar de los píxeles negativos. Si k es el número de píxeles positivos, entonces $k!$ es el número de las diferentes maneras de mover los P_p al lugar de los P_n . En el paso cuatro se requiere encontrar el lugar adecuado en los píxeles negativos para cada uno de los píxeles positivos, para obtener así que la distancia recorrida por todos los P_p es la menor. Para encontrar esta distancia se considera a una gráfica bipartita completa pesada con conjuntos de bipartición P_p y P_n , donde a cada arista (p_i, n_j) con $p_i \in P_p$ y $n_j \in P_n$, se le asocia un peso que corresponde a la distancia Euclidiana entre los píxeles p_i y n_j . El apareo perfecto óptimo para la gráfica proporciona las distancias a recorrer de cada uno de los P_p y la suma de estas distancias, que para este caso debe ser la mínima.

Las figuras 3.10 y 3.11, ilustran el proceso de transformación y la gráfica bipartita completa pesada con el apareo perfecto óptimo, respectivamente.

El procedimiento para encontrar el apareo perfecto óptimo está dado por el algoritmo húngaro, el cual se verá en la siguiente sección.

3.5.3. Algoritmo húngaro

El algoritmo húngaro [14] descrito por James R. Evans y Edward Minieka [8] es necesario para realizar la transformación de un objeto a otro. Este algoritmo permite encontrar el apareo perfecto óptimo en una gráfica bipartita completa pesada. Dicho algoritmo usa los arreglos $u[]$, $v[]$ y $\pi[]$ que ayudan a construir los apareos, una función predecesora $p()$ para construir caminos extendidos, un apareo inicial M , una matriz $c[][]$ donde serán almacenados los pesos de cada arista de la gráfica bipartita, y dos conjuntos S y T que representan a los conjuntos de bipartición de la gráfica. El tamaño de los arreglos $u[]$, $v[]$ y $\pi[]$ está determinado por la cardinalidad del conjunto S o T .

El algoritmo original encuentra el apareo perfecto óptimo de peso máximo pero a continuación se describe una variante de él que encuentra el apareo perfecto

¹ $A \setminus B$ es la diferencia de conjuntos.

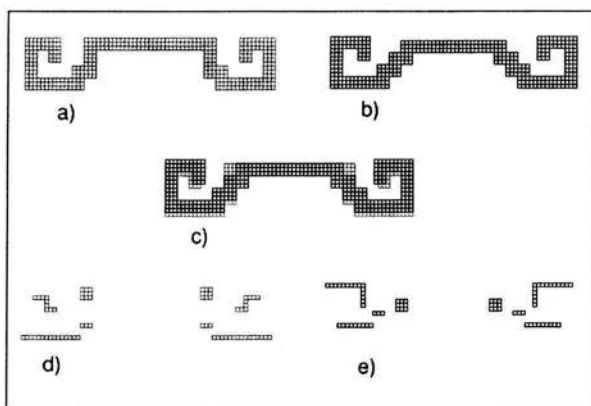


Figura 3.10: Proceso de transformación de un objeto a otro. a) Objeto A. b) Objeto B. c) Sobreposición de los objetos A y B. d) Píxeles positivos, estos píxeles serán los trasladados. e) Píxeles negativos, en los lugares de éstos se ubicarán los píxeles positivos.

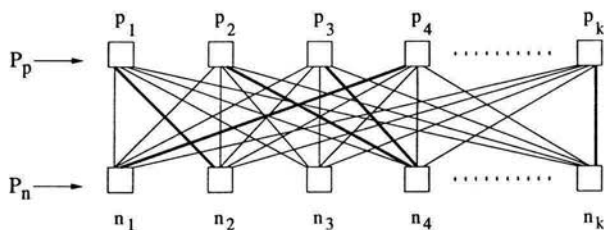


Figura 3.11: Gráfica bipartita completa pesada con conjuntos de bipartición P_p y P_n . Las líneas más gruesas muestran el apareo perfecto óptimo.

óptimo de peso mínimo.

Inicio. Obtener $u[i] = \min\{c[i][j] \mid \forall j \in T\}$ para cada $i \in S$. Hacer $v[j] = 0$, $\pi[j] = \infty$ tal que $j \in T$. Construir una subgráfica que consista de todas las aristas para las cuales $u[i] + v[j] = c[i][j]$, con $i \in S$ y $j \in T$. Para cada $i \in S$, elegir la primer arista (i, j) , $j \in T$, tal que j no esté saturado y colocarla en el apareo inicial M . Todos los nodos no han sido etiquetados ni visitados.

Paso 1. Etiquetar cada nodo $i \in S$ no saturado con $p(i) = 0$.

Paso 2. Seleccionar cualquier nodo $i \in S$ no visitado y etiquetado, o elegir $j \in T$ tal que $\pi[j] = 0$. Si no existen, entonces ir al paso 5.

Paso 3. Si el nodo elegido en el paso 2 es $i \in S$, entonces para cada arista (i, j) , con $j \in T$, tal que $(i, j) \notin M$, etiquetar el nodo j con $p(j) = i$, si $-u[i] + v[j] + c[i][j] < \pi[j]$, reemplazando cualquier etiqueta existente, y reemplazar $\pi[j]$ con $-u[i] + v[j] + c[i][j]$. Si el nodo elegido en el paso 2 es $j \in T$ y si no está saturado, entonces ir al paso 4; si está saturado, entonces hay una arista $(i, j) \in M$, etiquetar el nodo $i \in S$ con $p(i) = j$. En otro caso ($i \in S$ o $j \in T$) regresar al paso 2.

Paso 4. Se ha encontrado un camino extendido que termina en el nodo $j \in T$. Trazar este camino usando la función predecesora $p()$. Aumentar el apareo M añadiendo todas las aristas del camino extendido que no estén en M . Hacer $\pi[j] = \infty \forall j \in T$, borrar todas las etiquetas y regresar al paso 1.

Paso 5. Calcular $\delta_1 = \max\{u[i] \mid i \in S\}$, $\delta_2 = \min\{\pi[j] > 0 \mid j \in T\}$, y $\delta = \min\{\delta_1, \delta_2\}$. Hacer $u[i] = u[i] + \delta$ para cada nodo etiquetado $i \in S$. Hacer $v[j] = v[j] + \delta \forall j \in T$ con $\pi[j] = 0$. Hacer $\pi[j] = \pi[j] - \delta$ para cada nodo etiquetado $j \in T$ con $\pi[j] > 0$. Si $\delta = \delta_2$, entonces ir al paso 2. De otra forma se ha encontrado el apareo óptimo de peso mínimo.

A continuación se dará un ejemplo de la aplicación del algoritmo. La gráfica bipartita usada para este ejemplo y su respectiva representación en matriz (matriz $c[\square][\square]$) son mostradas en la figura 3.12.

Inicio. $u[1] = 12$, $u[2] = 20$, $u[3] = 14$ y $u[4] = 14$.
 $v[a] = 0$, $v[b] = 0$, $v[c] = 0$ y $v[d] = 0$.
 $\pi[a] = \infty$, $\pi[b] = \infty$, $\pi[c] = \infty$ y $\pi[d] = \infty$.
 La subgráfica es mostrada en la figura 3.13.
 $M = \{(1, a), (2, b)\}$

Paso 1. Los nodos 3 y 4 no son saturados, entonces $p(3) = 0$ y $p(4) = 0$.

Paso 2. Los nodos que están en S no visitados y etiquetados son: $\{3, 4\}$.

No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.

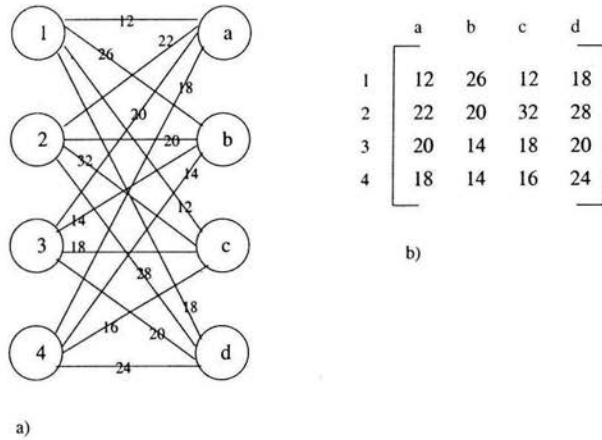


Figura 3.12: a) Gráfica bipartita usada para el ejemplo de la aplicación del algoritmo húngaro. b) Matriz de la gráfica mostrada en 3.12.a.

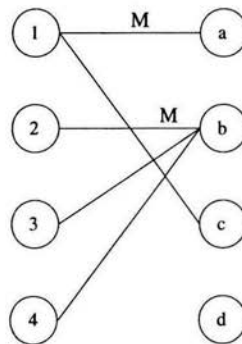


Figura 3.13: Subgráfica obtenida en el paso inicial del ejemplo de aplicación del algoritmo húngaro. Las aristas $(1, a)$ y $(2, b)$ pertenecen al apareo M .

Escoger $i = 3$.

Paso 3. Etiquetar $j \in T$: $p(a) = 3$, $p(b) = 3$, $p(c) = 3$ y $p(d) = 3$.

Los nuevos valores para π son: $\pi[a] = 6$, $\pi[b] = 0$, $\pi[c] = 4$ y $\pi[d] = 6$.

Paso 2. El nodo que está en S no visitado y etiquetado es: $\{4\}$.

El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{b\}$.

Escoger $i = 4$.

Paso 3. Etiquetar $j \in T$: $p(a) = 4$ y $p(c) = 4$.

Los nuevos valores para π son: $\pi[a] = 4$, $\pi[b] = 0$, $\pi[c] = 2$ y $\pi[d] = 6$.

Paso 2. No hay nodos en S no visitados y etiquetados.

El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{b\}$.

Escoger $j = b$.

Paso 3. El nodo b es saturado. Etiquetar $p(2) = b$.

Paso 2. El nodo que está en S no visitado y etiquetado es: $\{2\}$.

No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.

Escoger $i = 2$.

Paso 3. Etiquetar $j \in T$: $p(a) = 2$.

Los nuevos valores para π son: $\pi[a] = 2$, $\pi[b] = 0$, $\pi[c] = 2$ y $\pi[d] = 6$.

Paso 2. No hay nodos en S no visitados y etiquetados.

No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.

Ir al paso 5.

Paso 5. $\delta_1 = \max\{12, 20, 14, 14\} = 20$, $\delta_2 = \min\{2, 2, 6\} = 2$, $\delta = \min\{20, 2\} = 2$.

Los nodos 2, 3, y 4 están etiquetados, entonces

$u[1] = 12$, $u[2] = 22$, $u[3] = 16$ y $u[4] = 16$.

$v[a] = 0$, $v[b] = 2$, $v[c] = 0$ y $v[d] = 0$.

$\pi[a] = 0$, $\pi[b] = 0$, $\pi[c] = 0$ y $\pi[d] = 4$.

$\delta = \delta_2$. Ir al paso 2.

Paso 2. No hay nodos en S no visitados y etiquetados.

Los nodos $j \in T$ no visitados con $\pi[j] = 0$ son: $\{a, c\}$.

Escoger $j = a$.

Paso 3. El nodo a es saturado. Etiquetar $p(1) = a$.

Paso 2. El nodo que está en S no visitado y etiquetado es: $\{1\}$.

El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{c\}$.

Escoger $i = 1$.

Paso 3. Los valores de p y π no cambian.

Paso 2. No hay nodos en S no visitados y etiquetados.

El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{c\}$.

Escoger $j = c$.

Paso 3. El nodo c es no saturado. Ir al paso 4.

Paso 4. Aumentar el apareo M con las aristas del camino extendido dadas por

- la función $p()$, empezando con el nodo encontrado en el paso anterior.
 $p(c) = 4$ y $p(4) = 0$, entonces $M = \{(1, a), (2, b), (4, c)\}$.
 $\pi[a] = \infty$, $\pi[b] = \infty$, $\pi[c] = \infty$ y $\pi[d] = \infty$.
- Paso 1.* El nodo 3 es no saturado, entonces $p(3) = 0$.
- Paso 2.* El nodo que está en S no visitado y etiquetado es: $\{3\}$.
 No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.
 Escoger $i = 3$.
- Paso 3.* Etiquetar $j \in T$: $p(a) = 3$, $p(b) = 3$, $p(c) = 3$ y $p(d) = 3$.
 Los nuevos valores para π son: $\pi[a] = 4$, $\pi[b] = 0$, $\pi[c] = 2$ y $\pi[d] = 4$.
- Paso 2.* No hay nodos en S no visitados y etiquetados.
 El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{b\}$.
 Escoger $j = b$.
- Paso 3.* El nodo b es saturado. Etiquetar $p(2) = b$.
- Paso 2.* El nodo que está en S no visitado y etiquetado es: $\{2\}$.
 No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.
 Escoger $i = 2$.
- Paso 3.* Etiquetar $j \in T$: $p(a) = 2$.
 Los nuevos valores para π son: $\pi[a] = 0$, $\pi[b] = 0$, $\pi[c] = 2$ y $\pi[d] = 4$.
- Paso 2.* No hay nodos en S no visitados y etiquetados.
 El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{a\}$.
 Escoger $j = a$.
- Paso 3.* El nodo a es saturado. Etiquetar $p(1) = a$.
- Paso 2.* El nodo que está en S no visitado y etiquetado es: $\{1\}$.
 No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.
 Escoger $i = 1$.
- Paso 3.* Etiquetar $j \in T$: $p(c) = 1$.
 Los nuevos valores para π son: $\pi[a] = 0$, $\pi[b] = 0$, $\pi[c] = 0$ y $\pi[d] = 4$.
- Paso 2.* No hay nodos en S no visitados y etiquetados.
 El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{c\}$.
 Escoger $j = c$.
- Paso 3.* El nodo c es saturado. Etiquetar $p(4) = c$.
- Paso 2.* El nodo que está en S no visitado y etiquetado es: $\{4\}$.
 No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.
 Escoger $i = 4$.
- Paso 3.* Los valores de p y π no cambian.
- Paso 2.* No hay nodos en S no visitados y etiquetados.
 No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.
 Ir al paso 5.
- Paso 5.* $\delta_1 = \max\{12, 22, 16, 16\} = 22$, $\delta_2 = \min\{4\} = 4$, $\delta = \min\{22, 4\} = 4$.

Los nodos 1, 2, 3, y 4 están etiquetados, entonces

$$u[1] = 16, u[2] = 26, u[3] = 20 \text{ y } u[4] = 20.$$

$$v[a] = 4, v[b] = 6, v[c] = 4 \text{ y } v[d] = 0.$$

$$\pi[a] = 0, \pi[b] = 0, \pi[c] = 0 \text{ y } \pi[d] = 0.$$

$$\delta = \delta_2. \text{ Ir al paso 2.}$$

Paso 2. No hay nodos en S no visitados y etiquetados.

El nodo $j \in T$ no visitado con $\pi[j] = 0$ es: $\{d\}$.

Escoger $j = d$.

Paso 3. El nodo d es no saturado. Ir al paso 4.

Paso 4. Aumentar el apareo M con las aristas del camino extendido dadas por la función $p()$, empezando con el nodo encontrado en el paso anterior.

$$p(d) = 3 \text{ y } p(3) = 0, \text{ entonces } M = \{(1, a), (2, b), (4, c), (3, d)\}.$$

$$\pi[a] = \infty, \pi[b] = \infty, \pi[c] = \infty \text{ y } \pi[d] = \infty.$$

Paso 1. No hay nodos en S no saturados.

Paso 2. No hay nodos en S no visitados y etiquetados.

No hay nodos $j \in T$ no visitados tal que $\pi[j] = 0$.

Ir al paso 5.

Paso 5. $\delta_1 = \max\{16, 26, 20, 20\} = 26, \delta_2 = \min\{\infty, \infty, \infty, \infty\} = \infty,$

$$\delta = \min\{26, \infty\} = 26.$$

$\delta = \delta_1$, entonces el algoritmo termina.

Entonces el apareo perfecto óptimo de la gráfica bipartita mostrada en la figura 3.12.a es dado por el apareo $M = \{(1, a), (2, b), (3, d), (4, c)\}$.

3.5.4. Medida de similitud

La medida de similitud, S , entre dos formas A y B usando el método de transformación de un objeto a otro es proporcionada por el apareo perfecto óptimo:

$$S(A, B) = \sum_{i,j}^k w(p_i, n_j) \quad (3.15)$$

donde $p_i \in P_p, n_j \in P_n, w(p_i, n_j)$ es el peso asignado para la arista (p_i, n_j) , y además la arista (p_i, n_j) pertenece al apareo perfecto óptimo. Recuerde que el peso de cada arista se tomó como la distancia Euclidiana entre los píxeles p_i y n_j .

Con esta medida se tiene que a mayor peso (S más grande) la diferencia entre los objetos a comparar es mayor. También, es claro que si las formas a comparar son iguales, la cantidad de píxeles positivos y negativos es cero, por lo tanto la medida de similitud es cero.

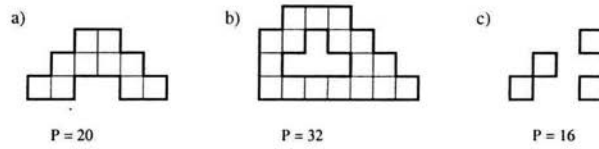


Figura 3.14: a) Forma con perímetro igual a 20. b) Forma que tiene dos segmentos de perímetro; su perímetro total es igual a la suma de ambos, 32. c) Forma que tiene más de dos segmentos de perímetro; su perímetro es igual a 16.

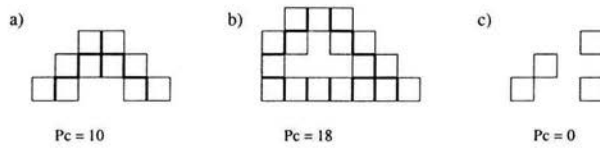


Figura 3.15: a) Forma con perímetro de contacto igual a 10. b) Forma con perímetro de contacto igual a 18. c) Forma sin perímetro de contacto.

3.6. Compacidad de forma

Otro método estudiado en el área de Reconocimiento de Patrones para comparar formas es el que usa *compacidad*. La compacidad es una propiedad importante de las formas, con ella se puede determinar que tan delgada es. Antes de mostrar el método de comparación usando compacidad se darán algunas definiciones.

3.6.1. Perímetro

El *perímetro* P de una forma corresponde a la longitud de su contorno. La longitud es obtenida de la suma de lados de los pixeles que forman el contorno. En la figura 3.14 se muestra un ejemplo. Se muestran tres formas diferentes con sus respectivos perímetros. El perímetro de cada una de ellas está marcado con una línea gruesa.

3.6.2. Perímetro de contacto

El *perímetro de contacto* P_c de una forma corresponde a la suma de longitudes que son comunes a dos pixeles. En la figura 3.15 se puede apreciar el perímetro de

contacto de cada una de las formas mostradas en la figura anterior. El perímetro de contacto está marcado con una línea gruesa.

3.6.3. Compacidad clásica

La compacidad de una forma se representa con C y se obtiene con la siguiente ecuación:

$$C = \frac{P^2}{N} \quad (3.16)$$

donde P es el perímetro de dicha forma y N es el número total de píxeles que tiene la forma, es decir, N es el área.

La compacidad es una propiedad de las formas invariantes bajo transformaciones geométricas tales como: traslación, rotación y escala. La invariancia bajo escala se tiene con la compacidad discreta normalizada, la cual se verá más adelante.

3.6.4. Compacidad discreta

Bribiesca [4] define la compacidad de las formas usando el perímetro de contacto y enuncia un teorema sobre este concepto:

Definición 3.9 *La medida de compacidad discreta C_D para formas corresponde al perímetro de contacto, es decir,*

$$C_D = P_C \quad (3.17)$$

Teorema 3.1 *Para cualquier forma que tenga N píxeles. La siguiente ecuación se satisface:*

$$2P_C + P = TN, \quad (3.18)$$

donde P_C es el perímetro de contacto, P es el perímetro de la forma, y T es el número de lados del píxel. Geométricamente, significa que la suma de dos veces el perímetro de contacto más el perímetro es igual a la suma total de los perímetros de todos los píxeles de la forma.

Con la ecuación 3.18 dada por el teorema se obtiene directamente el perímetro de contacto:

$$P_C = \frac{TN - P}{2} \quad (3.19)$$

y se sabe que el número de lados de un píxel es 4, por lo cual,

$$P_C = \frac{4N - P}{2} \quad (3.20)$$

Por lo tanto, obtener la compacidad discreta de una forma basta con calcular el perímetro de contacto con la ecuación 3.20.

El perímetro de contacto se puede obtener para cualquier tipo de formas, incluyendo formas con hoyos o formas sin perímetro de contacto, por ejemplo, las formas que se muestran en la figuras 3.14.b y 3.14.c respectivamente. Para la forma mostrada en la figura 3.14.a se tiene que, $P = 20$ y $N = 10$, entonces sustituyendo en la ecuación 3.20 se tiene que $P_C = \frac{4 \times 10 - 20}{2} = \frac{20}{2} = 10$, que es justamente el valor que se había mostrado en la figura 3.15.a. Para la forma mostrada en la figura 3.14.b se tiene que, $P = 32$ y $N = 17$, entonces haciendo la sustitución se tiene que $P_C = \frac{4 \times 17 - 32}{2} = \frac{36}{2} = 18$, nuevamente coincide con el valor mostrado en la figura 3.15.b. Finalmente, para la forma mostrada en la figura 3.14.c se tiene que, $P = 16$ y $N = 4$, haciendo la sustitución se tiene que $P_C = \frac{4 \times 4 - 16}{2} = \frac{0}{2} = 0$, lo cual corresponde al valor de la figura 3.15.c y en este caso se dice que la forma no tiene perímetro de contacto.

3.6.5. Compacidad discreta máxima y mínima

La compacidad discreta está determinada a partir del perímetro de contacto, el cual expresa cuántos lados de píxel son comunes a dos píxeles, o dicho de otra manera, cuántos píxeles se tocan entre si. Para cierta cantidad de píxeles existe una forma en la que los píxeles casi no se tocan. Dicha forma es aquella que cumple con ser una línea delgada, es decir, es formada de píxeles alineados consecutivamente. Esta forma es la que tiene la *compacidad mínima*.

La ecuación para calcular la compacidad discreta mínima C_{Dmin} de una forma con N píxeles es:

$$C_{Dmin} = N - 1 \quad (3.21)$$

De manera opuesta, hay una forma en la que la mayoría de los píxeles que contiene se tocan entre si. Esta forma es un cuadrado y es la que tiene *compacidad máxima*.

La ecuación para calcular la compacidad discreta máxima C_{Dmax} de una forma con N píxeles es:

$$C_{Dmax} = 2(N - \sqrt{N}) \quad (3.22)$$

La ecuación 3.22 es obtenida de:

$$C_{Dmax} = \frac{TN - 4\sqrt{N}}{2} \quad (3.23)$$

donde se sustituyó a T por 4 porque se está tomando en cuenta que la forma contiene pixeles y que éstos tienen 4 lados. Esta última ecuación se obtuvo a partir de la ecuación 3.19.

3.6.6. Compacidad discreta normalizada

Para dos formas iguales pero con diferente área, el valor de su compacidad discreta es distinta. Eso se debe a que el cálculo de la compacidad es en términos del número de pixeles que contiene la forma. En este caso la forma con mayor área, tendrá compacidad más grande ya que contiene un número mayor de pixeles y por lo tanto tiene más pixeles que se tocan entre si.

Lo ideal sería que la compacidad discreta fuera invariante bajo escala, es decir, dos formas iguales pero con diferente área tuvieran el mismo valor de compacidad. Para ello, se define la *compacidad discreta normalizada*, la cual permite conservar el valor de compacidad discreta independientemente del área.

La ecuación que obtiene el valor de la compacidad discreta normalizada C_{DN} es la siguiente:

$$C_{DN} = \frac{C_D - C_{Dmin}}{C_{Dmax} - C_{Dmin}} \quad (3.24)$$

Los valores que adquiere la compacidad discreta normalizada varían en el rango $[0, 1]$.

3.6.7. Comparación de formas usando compacidad

Si dos formas son idénticas su compacidad normalizada es la misma y si son muy parecidas, su valor de compacidad no debe diferir mucho. Esto no siempre pasa debido a que por ejemplo, si se toman a dos formas iguales pero una con su contorno distorsionado, sus valores de compacidad pueden diferir mucho ya que el perímetro de la forma con contorno distorsionado es mayor y por lo tanto su compacidad también. Por lo tanto, se pueden encontrar formas totalmente diferentes con el mismo valor de compacidad. Esta es una gran desventaja para la comparación de formas usando compacidad.

Por lo anterior, la compacidad no es una buena medida de similitud para formas. Lo que si se puede concluir es que con la compacidad se pueden clasificar a las formas

en delgadas o gruesas. Además, el uso de ésta puede ser un proceso previo a métodos que proporcionan medidas de similitud.

Capítulo 4

Aplicación y Resultados

4.1. Introducción

En el capítulo 1 se mencionó el objetivo principal de esta tesis, reconocer determinados símbolos en imágenes de las culturas mesoamericanas. En el capítulo 2 se explicó ampliamente el proceso de extracción de objetos de una imagen. Y en el capítulo 3 se mostraron diferentes técnicas de comparación de formas u objetos. Ahora, en este capítulo se conjuntarán los métodos expuestos en los capítulos 2 y 3 para aplicarlos a las imágenes que son de interés tal como se mencionó en el capítulo 1, dando como resultado un proceso de reconocimiento; también se mostrarán los resultados principales y más significativos de la aplicación.

En la sección 4.2 se explica la aplicación del proceso de segmentación a las imágenes. Para ejemplificar este proceso se usan cuatro imágenes tomadas del conjunto de imágenes mesoamericanas recopiladas, y se muestra los resultados de la segmentación realizada sobre ellas.

En la sección 4.3 se muestran las imágenes obtenidas de la segmentación pero ahora de forma invariante, es decir, todas ellas tienen la misma orientación, posición y tamaño.

Y finalmente, en la sección 4.4 se dan los resultados de comparación entre las formas a buscar (“iconos uno”) y las formas segmentadas e invariantes. El método usado para la comparación es la transformación de un objeto a otro. Para complementar estos resultados se analizan y comparan las compatibilidades de las formas segmentadas y los “iconos uno”, y se ve como ayuda esto a la comparación de formas.

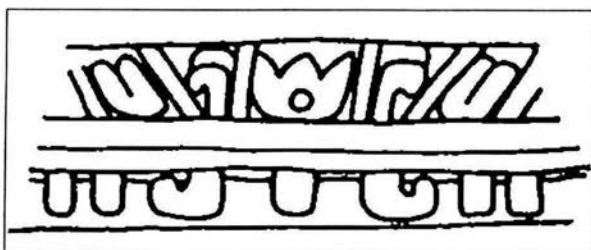


Figura 4.1: Imagen de la cultura Teotihuacana.

4.2. Segmentación de imágenes

Las imágenes mostradas en las figuras 4.1, 4.2, 4.3 y 4.4 son usadas para explicar el procedimiento del reconocimiento de símbolos.

El primer paso del reconocimiento es la realización de la segmentación. Como se puede apreciar en las imágenes mostradas en las figuras 4.1, 4.2, 4.3 y 4.4, las formas que se quiere extraer de ellas, son los dibujos plasmados o las formas que sobresalen en las piedras. Los principales métodos de la aplicación usados para la segmentación son los que requieren de la aplicación de umbrales, además de técnicas complementarias como los algoritmos de localización y unión de bordes para extraer los objetos que están deteriorados o incompletos. También se aplicaron los algoritmos de segmentación por regiones crecientes; estos métodos son más eficientes para imágenes similares a las que se muestran en las figuras 4.1 y 4.3 debido a que las formas a extraer son dibujos en donde el contorno está bien delimitado.

A continuación se detalla el proceso de segmentación para cada una de las imágenes mostradas en las figuras 4.1, 4.2, 4.3 y 4.4.

Para la imagen en la figura 4.1 el método de segmentación usado fue la segmentación por regiones crecientes. En esta imagen es inútil aplicar umbrales debido a que solo contiene dos tonos (blanco y negro), y al aplicarle un umbral se obtiene la misma imagen, lo cual no es conveniente porque se quiere obtener los objetos que la forman. En la figura 4.5 se muestra la imagen de la figura 4.1 con las semillas generadoras de regiones señaladas con una flecha, y en la figura 4.6 se muestra las regiones obtenidas a partir de estas semillas.

Los métodos de segmentación aplicados a la imagen de la figura 4.2 son: aplicación de umbral y localización de bordes; dando como resultado las imágenes mostradas en las figuras 4.7.a y 4.7.b respectivamente. Se puede observar que los objetos en estas imágenes no están bien delimitados, por lo que una segmentación



Figura 4.2: Imagen del Tajín.



Figura 4.3: Imagen de la cultura Mixteca.



Figura 4.4: Imagen de la cultura Olmeca.

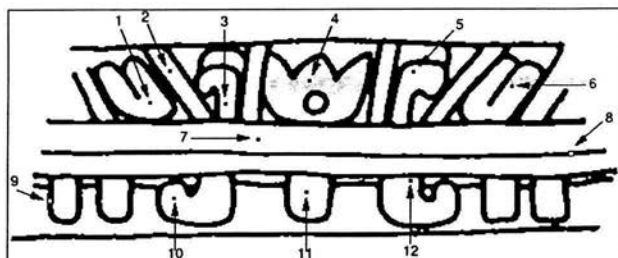


Figura 4.5: Imagen de la figura 4.1 con algunas semillas generadoras de regiones señaladas con una flecha.

mediante regiones es poco eficaz. A partir de las imágenes de la figura 4.7 la segmentación se realizó manualmente.

Para la imagen de la figura 4.3 se realizó el mismo procedimiento que para la imagen de la figura 4.1 pero primero se aplicó un umbral para obtener una imagen en dos tonos (figura 4.8.a) y se le aplicó un filtro para eliminar el exceso de ruido (figura 4.8.b). En la figura 4.9 se muestra la imagen de la figura 4.8.b con algunas semillas generadoras y en las figuras 4.10, 4.11 y 4.12 se muestran las regiones obtenidas de la segmentación a partir de dichas semillas.

La imagen mostrada en la figura 4.13 es el resultado de la segmentación mediante la localización de bordes de la figura 4.4, y las imágenes mostradas en las figuras 4.14 y 4.15 son las imágenes resultantes de la segmentación mediante la aplicación de umbrales para la misma imagen. Se aplicaron dos umbrales diferentes porque así se detectan más objetos. Posteriormente, la segmentación se realizó manualmente por las mismas razones que para la imagen de la figura 4.2, los contornos de la imágenes resultantes de la aplicación de umbrales y localización de bordes no están bien definidos.

Finalmente, se seleccionaron algunos de los objetos extraídos de cada una de las imágenes, siendo los resultados del proceso de segmentación, los cuales se pueden apreciar en las figuras 4.16, 4.17, 4.18 y 4.19.

Es importante notar que el proceso de segmentación puede producir como resultado objetos que no son de interés o que no representan a alguna forma pero debido a que se quiere automatizar el proceso se deben considerar todos los objetos extraídos.

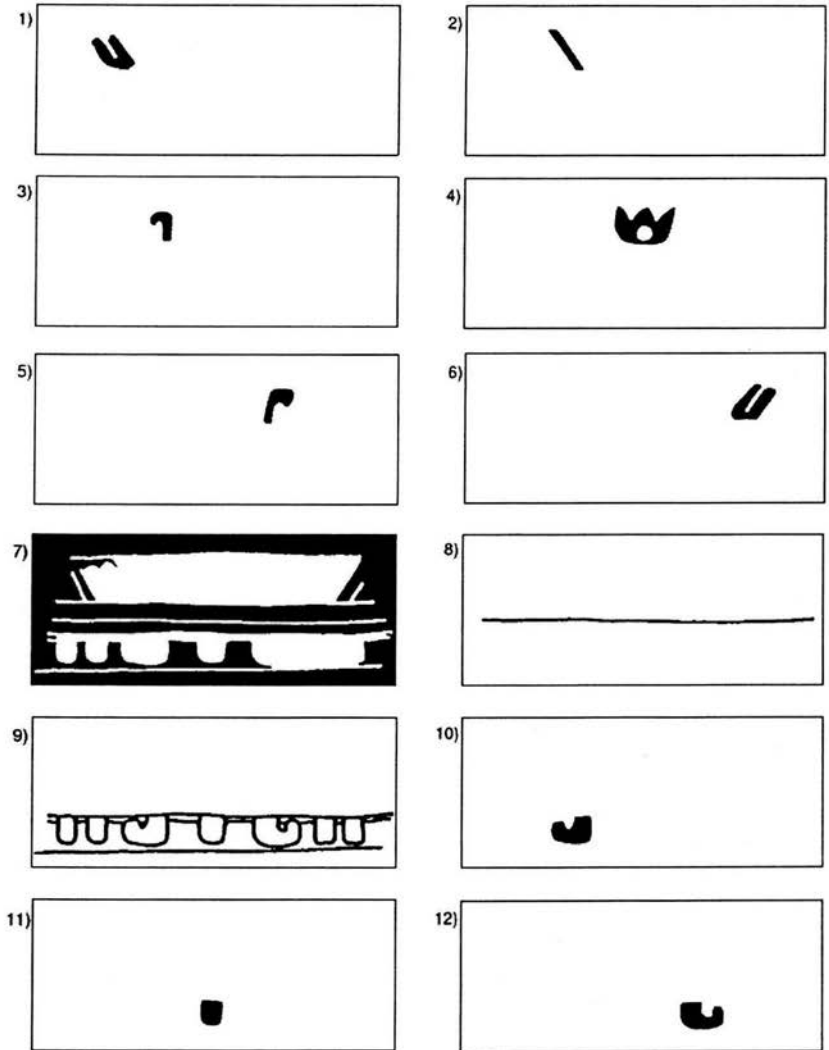


Figura 4.6: Regiones obtenidas a partir de la segmentación mediante regiones crecientes de la imagen mostrada en la figura 4.5.

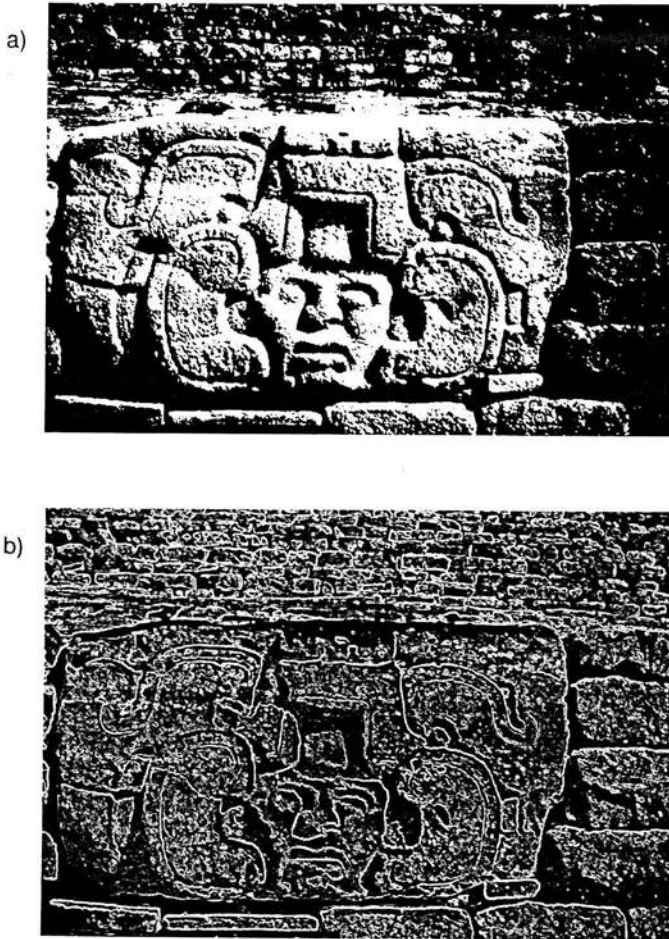


Figura 4.7: a) Imagen resultante de la aplicación de un umbral a la imagen de la figura 4.2. b) Imagen resultante de la localización de bordes aplicado a la imagen de la figura 4.2.

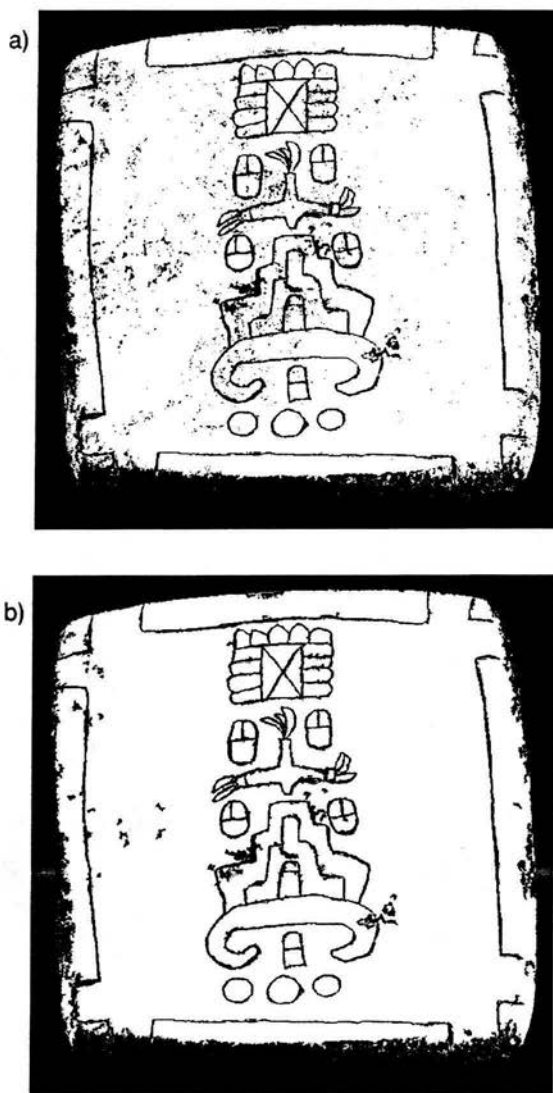


Figura 4.8: a) Imagen resultante de la aplicación de un umbral a la imagen de la figura 4.3. b) Imagen resultante de la aplicación de un filtro a la imagen de la figura 4.8.a.

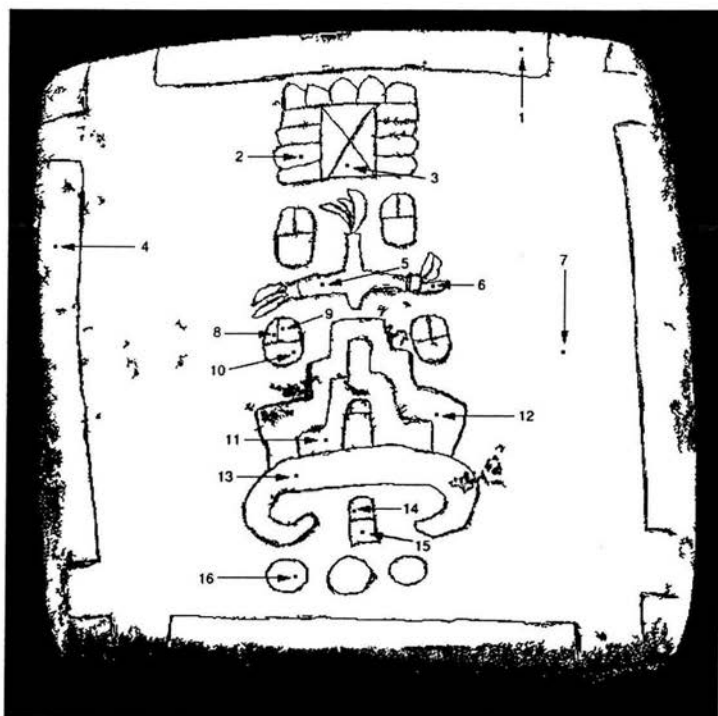


Figura 4.9: Imagen de la figura 4.8.b con algunas semillas generadoras de regiones señaladas con una flecha.

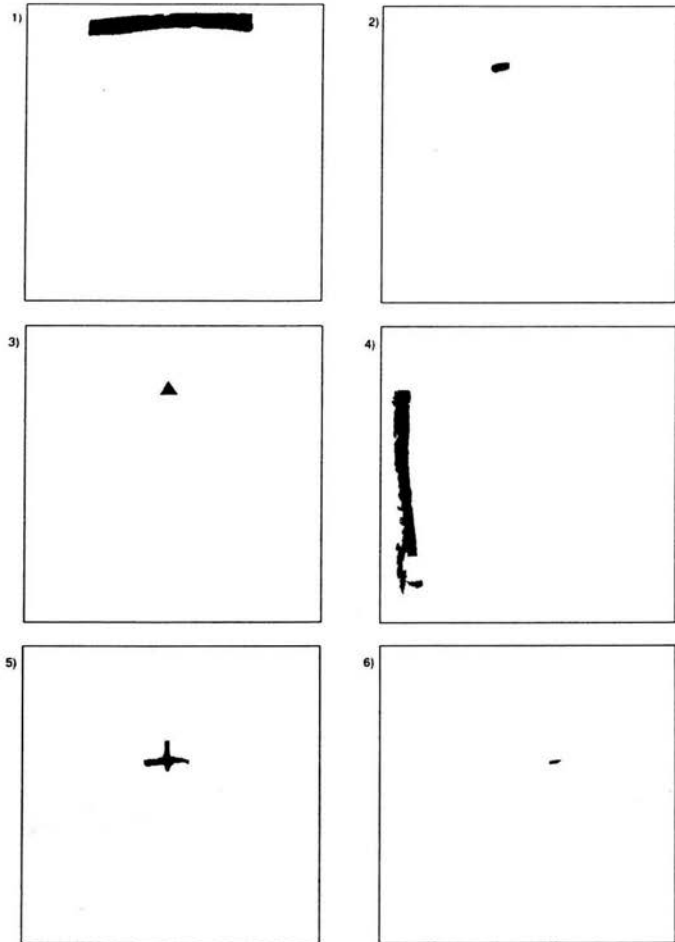


Figura 4.10: Regiones obtenidas a partir de la segmentación mediante regiones crecientes de la imagen mostrada en la figura 4.9.

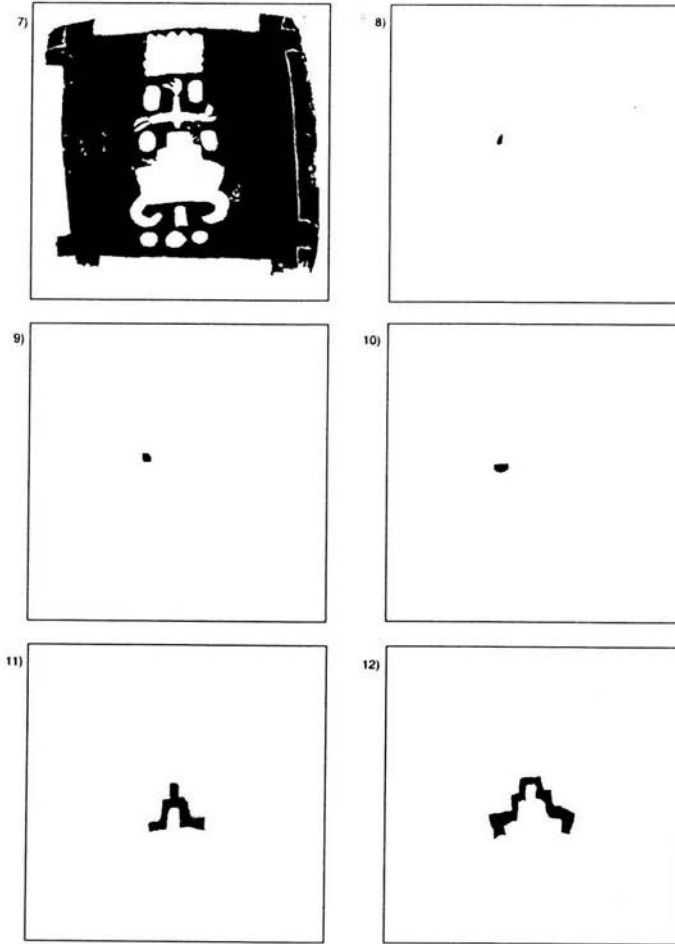


Figura 4.11: Regiones obtenidas a partir de la segmentación mediante regiones crecientes de la imagen mostrada en la figura 4.9.

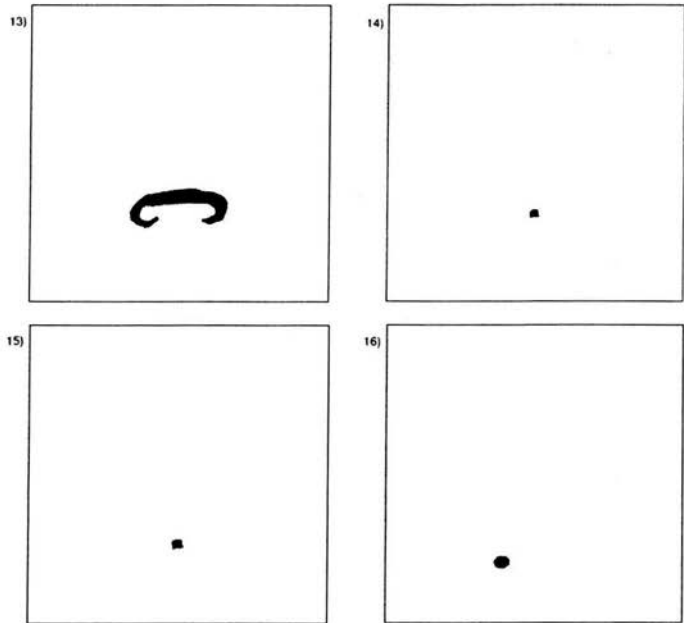


Figura 4.12: Regiones obtenidas a partir de la segmentación mediante regiones crecientes de la imagen mostrada en la figura 4.9.



Figura 4.13: Imagen resultante de la localización de bordes aplicado a la imagen de la figura 4.4.

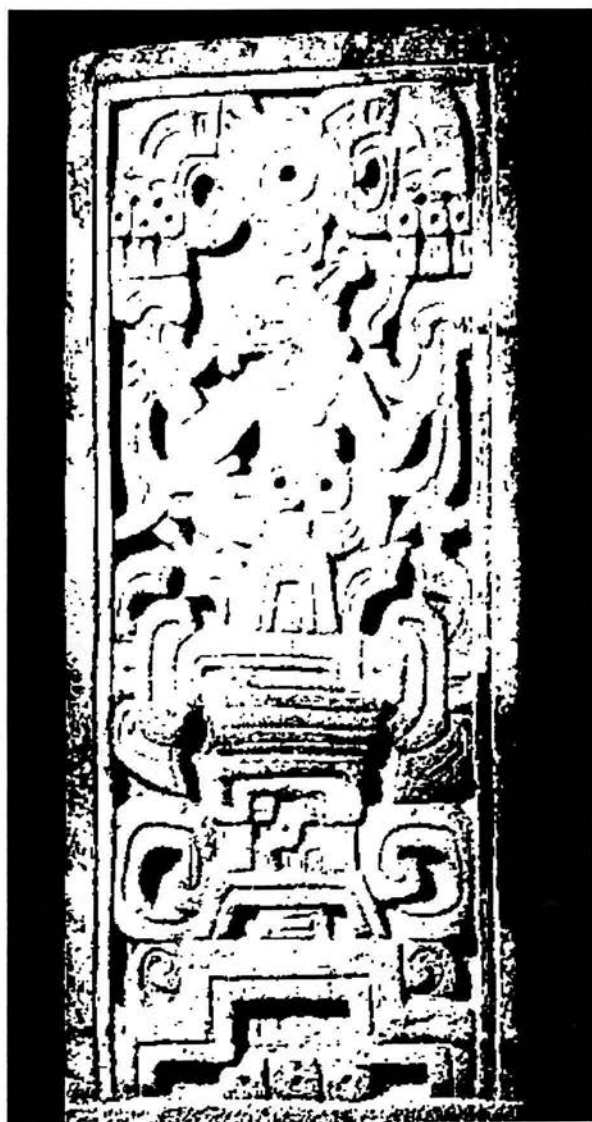


Figura 4.14: Imagen resultante de la aplicación de un umbral a la imagen de la figura 4.4.



Figura 4.15: Imagen resultante de la aplicación de un umbral a la imagen de la figura 4.4.

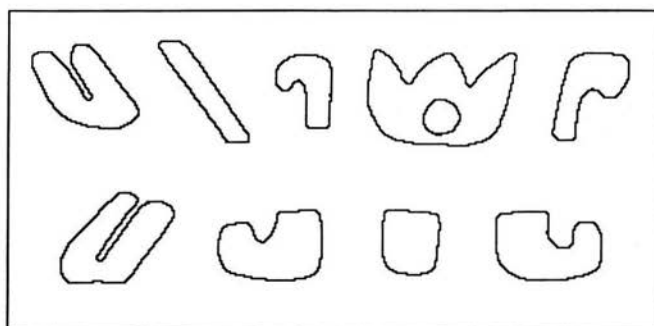


Figura 4.16: Algunos objetos segmentados de la imagen mostrada en la figura 4.1.

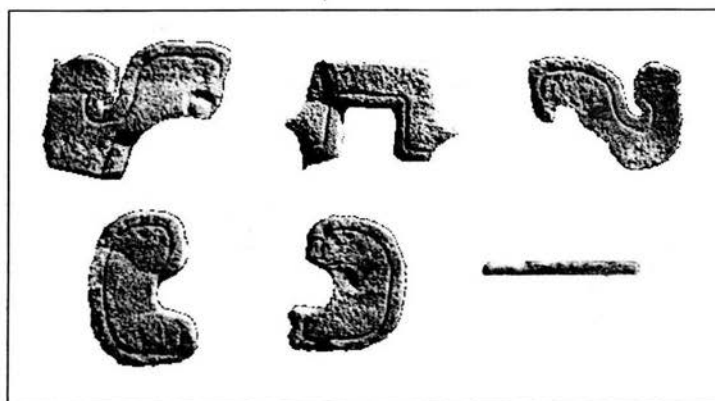


Figura 4.17: Algunos objetos segmentados de la imagen mostrada en la figura 4.2.

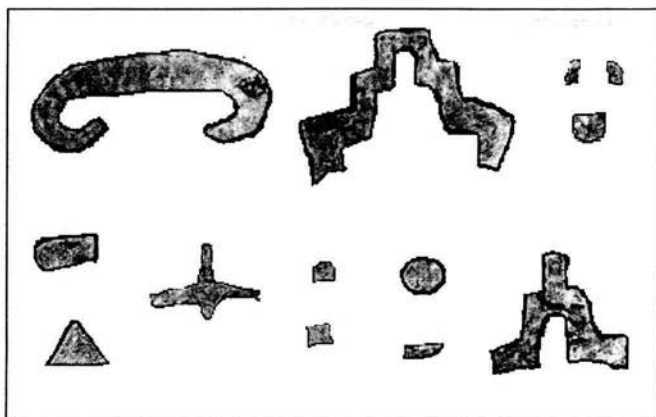


Figura 4.18: Algunos objetos segmentados de la imagen mostrada en la figura 4.3.

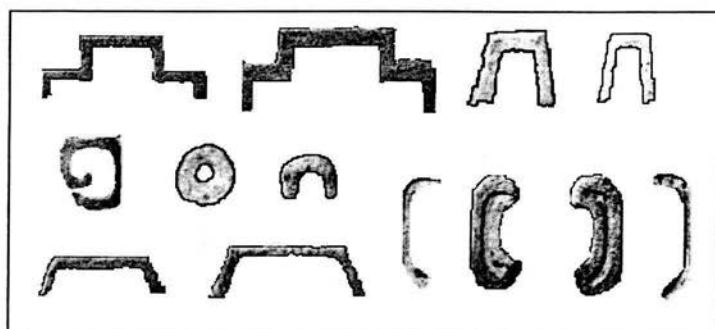


Figura 4.19: Algunos objetos segmentados de la imagen mostrada en la figura 4.4.

4.3. Aplicando invariantes

Por las razones expuestas en la sección 3.2 y por los requerimientos de los diferentes métodos de comparación de formas, se obtuvieron formas invariantes de los objetos mostrados en las figuras 4.16, 4.17, 4.18 y 4.19. La figura 4.20 muestra estas formas invariantes. Todas las formas tienen la misma área: 800 píxeles. Note que las formas tienen la misma orientación, es decir, la ubicación de sus ejes principales coinciden; que a su vez coinciden con la ubicación de los ejes del plano, horizontal y vertical. El centro de masa de cada una de las formas está marcado con un •.

4.4. Comparación de formas

El objetivo principal de este trabajo es reconocer determinados símbolos en las imágenes, por lo cual, cada uno de los símbolos (mostrados en la figura 4.21) se compararon con los objetos de la figura 4.20. Las formas de la figura 4.21 son invariantes en cuanto a escala, traslación y rotación, de la misma manera que los objetos mostrados en la figura 4.20.

Primero se analizaron las compacidades de cada una de las formas, esto es con el objeto de poder clasificar a los objetos segmentados y determinar cuáles se van a comparar con los “iconos uno”. En las tablas 4.1 y 4.2 se muestran las compacidades normalizadas de los objetos segmentados y de los “iconos uno”, respectivamente.

Se sabe que las compacidades sólo clasifican a las formas en delgadas y gruesas, entonces la manera de realizar la clasificación es con el siguiente proceso:

1. Encontrar el valor máximo de las compacidades normalizadas de todos los “iconos uno”.
2. Encontrar el valor mínimo de las compacidades normalizadas de todos los “iconos uno”.
3. Para cada forma segmentada determinar si su valor de compacidad normalizada pertenece al rango dado por los valores encontrados en los puntos 1 y 2.

La forma encontrada en el paso 1 es icono04, y su valor de compacidad normalizada es 0.972554. La forma encontrada en el paso 2 es icono15, y su valor de compacidad normalizada es 0.838223. De los objetos segmentados, seg01 a seg41, los que tienen un valor de compacidad en el rango de 0.838223 a 0.972554 son: seg01, seg02, seg03, seg04, seg05, seg06, seg07, seg09, seg10, seg11, seg12, seg13, seg14, seg15, seg17, seg18, seg23, seg24, seg25, seg29, seg30, seg31, seg32, seg33,

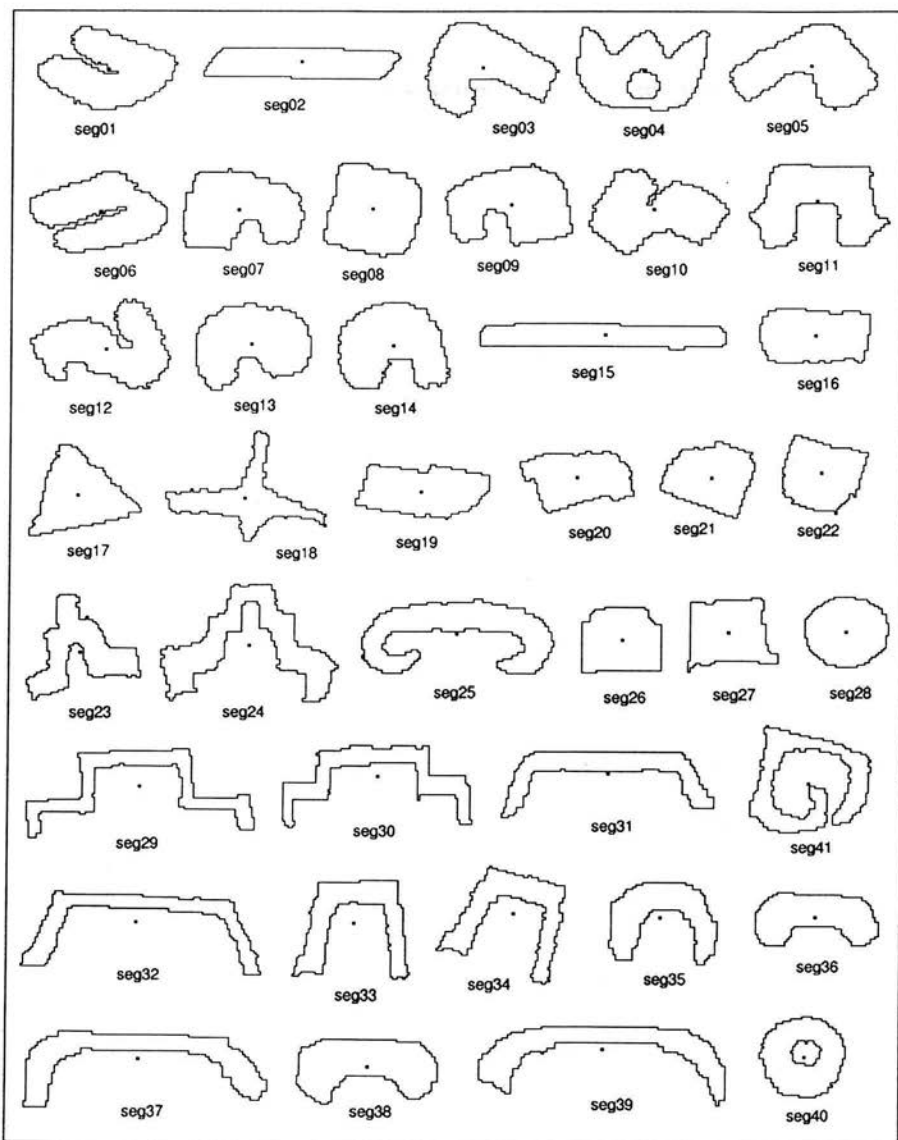


Figura 4.20: Formas invariantes en escala y rotación, obtenidas de las figuras 4.16, 4.17, 4.18 y 4.19. El centro de masa está marcado con un •.

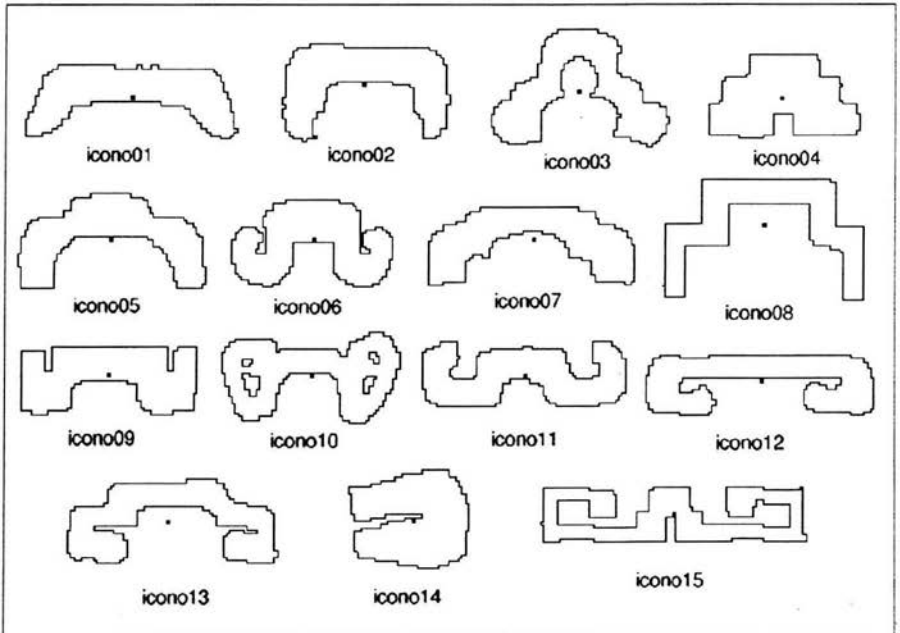


Figura 4.21: “Iconos uno” invariantes en escala y rotación. Su centro de masa está marcado con un • .

Imagen	Compacidad Normalizada
seg01	0.941658
seg02	0.956434
seg03	0.956434
seg04	0.926881
seg05	0.953747
seg06	0.932255
seg07	0.969867
seg08	0.990017
seg09	0.965837
seg10	0.948374
seg11	0.951061
seg12	0.933598
seg13	0.972554
seg14	0.964494
seg15	0.934941
seg16	0.983300
seg17	0.968524
seg18	0.904045
seg19	0.972664
seg20	0.972612
seg21	0.979270
seg22	0.979270
seg23	0.922851
seg24	0.887925
seg25	0.898672
seg26	0.995390
seg27	0.979270
seg28	0.990017
seg29	0.852999
seg30	0.881209
seg31	0.898672
seg32	0.861059
seg33	0.878522
seg34	0.883896
seg35	0.940314
seg36	0.967181
seg37	0.898672
seg38	0.965837
seg39	0.894642
seg40	0.948374
seg41	0.850313

Cuadro 4.1: Compacidades normalizadas de los objetos mostrados en la figura 4.20.

Imagen	Compacidad Normalizada
icono01	0.944344
icono02	0.944344
icono03	0.913448
icono04	0.972554
icono05	0.941658
icono06	0.930911
icono07	0.938971
icono08	0.910762
icono09	0.947031
icono10	0.898672
icono11	0.922851
icono12	0.909418
icono13	0.905388
icono14	0.959121
icono15	0.838223

Cuadro 4.2: Compacidades normalizadas de los “iconos uno”.

seg34, seg35, seg36, seg37, seg38, seg39, seg40, seg41. Se tiene que de los 41 objetos segmentados, 9 no entran en la clasificación. Estos 9 no se compararán con los “iconos uno” debido a que sus compacidades son muy diferentes. Visualmente se puede apreciar que esos 9 objetos no se parecen a los “iconos uno”.

El paso que sigue a la clasificación mediante el uso de compacidades es la comparación de los objetos segmentados con los “iconos uno”; el método usado para esto es: transformación de un objeto a otro, el cual utiliza el algoritmo húngaro. La razón para utilizar este algoritmo es para mostrar su desempeño en la comparación de formas y verificar su eficiencia.

En las tablas 4.4 y 4.5 se muestran los resultados obtenidos de la comparación entre los “iconos uno” y las formas segmentadas. Los valores que están resaltados en negritas indican a cuál icono se parecen más cada una de las formas. Por ejemplo, en la tabla 4.3 el valor resaltado es 2046.96, entonces la figura llamada seg01 se parece más a icono04.

	icono01	icono02	icono03	icono04	icono05	icono06	icono07
seg01	5117.14	3441.52	4646.75	2046.96	4764.82	2328.55	5713.88

Cuadro 4.3: Ejemplo. La forma seg01 se parece más a icono04.

Los datos mostrados en las tablas 4.4 y 4.5 sólo permiten saber a qué “icono uno” se parece más cada una de las formas y no qué tanto se parecen. En la tabla 4.6 se muestran los porcentajes de similitud entre la forma y el “icono uno” al cual se parece más. Es importante notar que el cálculo de los porcentajes es relativo a los quince “iconos uno” dados; si hubiera más (o menos) “iconos uno” los porcentajes podrían cambiar debido a que no son todas las formas existentes con las cuales se pueden comparar.

Para determinar si un objeto segmentado es un “icono uno”, el porcentaje de similitud entre el objeto y el “icono uno” debe ser mayor o igual al 80%. Este porcentaje puede variar dependiendo del problema a solucionar y es tomado así porque es el más conveniente para el problema presentado. El cálculo de porcentaje es mediante la siguiente ecuación:

$$z = 100 - \frac{x}{y/100} \quad (4.1)$$

donde z es el porcentaje resultante, x es el valor mínimo obtenido en las tablas 4.4 y 4.5 para cada una de las formas, es decir, son los valores en negritas, y y es el caso contrario de x , es el valor máximo para cada una de las formas. Por ejemplo, para seg01, $x = 1565.75$ y $y = 8649.64$, por lo tanto $z = 81.90$.

Entonces, los objetos que entran en la clasificación (formas cuyo porcentaje es mayor al 80%) son: seg01, seg06, seg11, seg13, seg14, seg23, seg24, seg25, seg29, seg30, seg31, seg32, seg33, seg34, seg35, seg36, seg37, seg38, seg39. Por lo tanto, estas formas pueden identificarse como “iconos uno”.

Como se puede observar, mediante el proceso expuesto en este capítulo se pueden identificar los símbolos deseados (“iconos uno”) en las imágenes de las culturas mesoamericanas. Por lo tanto, se puede concluir que el proceso expuesto en este trabajo se realizó con éxito.

	icono01	icono02	icono03	icono04	icono05	icono06	icono07
seg01	6389.95	4732.51	4759.65	1583.95	5334.17	2495.26	6922.76
seg02	2645.48	4941.55	6774.95	6698.56	4558.35	5751.89	3353.85
seg03	6128.97	4143.28	4002.60	2125.06	4686.61	2257.63	6762.48
seg04	5117.14	3441.52	4646.75	2046.96	4764.82	2328.55	5713.88
seg05	5293.44	3907.38	3698.36	1666.55	4163.23	1666.75	5805.84
seg06	6160.79	4343.91	4694.03	1546.73	5045.14	2387.43	6724.74
seg07	5919.74	4070.98	4298.47	1915.43	4710.55	1984.27	6538.92
seg09	5864.97	4147.88	4312.86	1734.99	4820.44	1892.89	6416.82
seg10	5431.09	3898.32	4360.43	1586.52	4669.70	2092.93	6079.77
seg11	5091.48	3078.13	3105.39	1662.10	3865.27	1103.26	5428.91
seg12	5518.82	4016.10	4362.20	1897.97	4808.68	2270.62	6074.52
seg13	6126.51	4228.90	4347.35	997.19	4847.71	2132.53	6738.46
seg14	6721.50	4720.23	4373.24	1318.95	5211.37	2508.28	7319.67
seg15	5690.31	8341.31	10269.79	11115.89	7846.42	9872.95	5959.17
seg17	7717.55	5824.02	4695.72	2055.99	6288.05	3520.90	8282.93
seg18	5987.22	5655.28	4716.94	3653.65	5434.43	3895.33	6393.40
seg23	7024.43	4918.17	3598.38	2209.55	5414.68	3215.83	7376.92
seg24	4463.95	3568.35	2002.39	5394.10	2893.93	4064.00	3829.36
seg25	1920.36	2879.66	4822.26	7463.83	2286.09	4849.88	1481.10
seg29	7928.60	8928.85	9372.05	3304.76	8128.28	10870.52	6941.50
seg30	5376.16	6473.35	7743.12	10162.47	5784.45	8537.03	4384.07
seg31	5709.25	7435.40	9632.80	11402.36	6874.28	9775.99	5304.27
seg32	10699.44	11825.11	13406.79	15901.98	11340.29	14225.69	7854.04
seg33	8477.65	6588.86	4917.02	6759.98	7024.38	3377.98	8087.14
seg34	6038.47	4156.92	3248.58	5347.84	4438.69	2388.94	5589.71
seg35	4860.64	2083.42	2360.83	3145.34	2928.45	2317.87	4976.19
seg36	2998.63	1887.31	3633.88	3036.94	2710.67	1963.75	3685.04
seg37	7567.69	9216.50	11127.37	13236.27	8702.60	11562.32	6997.01
seg38	3142.82	1481.41	1418.73	2973.19	2823.20	1800.93	3819.85
seg39	7623.05	9382.49	3787.81	13371.93	8855.70	11704.16	7108.96
seg40	7989.57	5868.91	4802.21	2722.77	6442.16	3839.35	8367.96
seg41	7094.09	5002.61	4651.12	5128.13	6061.70	4738.57	7207.87

Cuadro 4.4: Tabla de resultados (primera parte). Comparación utilizando el método de la transformación de un objeto a otro.

	icono08	icono09	icono10	icono11	icono12	icono13	icono14	icono15
seg01	7526.26	4696.80	4793.84	5768.25	8610.11	7145.48	1565.75	8649.64
seg02	5883.93	3436.84	4099.17	2354.91	2884.61	4058.65	8274.83	2650.97
seg03	6990.47	4594.58	4600.03	5488.21	8372.62	6670.14	2484.89	8483.50
seg04	6353.59	3411.87	3232.54	4355.43	7188.51	5964.76	2590.36	7464.85
seg05	6359.60	3507.99	3774.92	4718.84	7488.21	6012.04	2756.49	7800.92
seg06	7237.98	4468.70	4472.58	5489.53	8375.50	6739.34	1545.26	8414.62
seg07	7065.37	4288.91	4346.76	5301.04	8179.51	6576.40	2004.46	8225.00
seg09	7049.33	4104.50	4140.74	5275.18	8091.41	6578.02	1887.39	8208.82
seg10	6900.03	3875.86	4011.49	4847.17	7695.77	6332.50	2636.51	7712.24
seg11	8228.95	3375.62	3225.47	4514.12	7110.30	5470.29	2722.89	8860.07
seg12	6715.36	3700.33	3683.07	4895.71	7628.28	6288.75	2572.15	7821.02
seg13	7141.86	4474.00	4512.60	5502.62	8364.66	6720.87	1711.70	8425.02
seg14	7491.98	5117.09	5074.00	6139.61	8973.59	7218.50	1343.08	9005.51
seg15	7797.90	7372.27	7775.97	6161.76	4572.18	6706.83	12711.39	3383.61
seg17	8389.62	6070.96	6064.53	7099.06	9937.75	8290.89	2046.41	9901.80
seg18	7083.91	4815.67	5227.51	5291.26	7753.92	6573.95	4200.91	7711.41
seg23	7213.34	5495.02	5502.55	6386.38	9025.12	7277.43	3689.30	12257.01
seg24	2978.92	4131.20	4557.55	4991.79	5591.71	3671.65	6697.36	1050.72
seg25	3197.95	2933.91	3395.00	2917.50	2444.47	1592.30	2648.76	4593.17
seg29	2683.37	9407.20	9391.53	8832.79	6461.38	6411.89	13601.13	7464.21
seg30	2939.63	6857.88	6876.17	6387.40	4092.17	4090.51	19431.54	5565.08
seg31	6341.12	7395.10	7608.46	6378.11	3240.92	5736.09	16781.07	4702.44
seg32	9480.46	12332.82	12305.79	11207.46	3710.75	9377.85	19148.25	8956.66
seg33	6615.10	7462.13	6967.05	8655.45	9650.02	7989.05	6611.51	17986.44
seg34	4220.69	5095.58	4901.93	6416.91	7233.07	5248.79	5806.39	12995.77
seg35	4501.75	3704.69	3602.73	4844.02	6650.76	4535.12	4117.61	11041.04
seg36	5055.21	1460.18	2079.91	2554.12	5293.18	4160.94	4603.12	8939.40
seg37	7517.70	9275.57	9473.92	8175.97	2548.97	6885.18	14618.16	6083.24
seg38	5211.24	1643.66	2024.83	2609.53	5437.20	4286.93	4523.55	8031.89
seg39	7748.47	9365.99	9613.60	8288.11	2633.71	7016.19	14772.66	6025.97
seg40	7959.51	6315.38	6085.64	7359.73	10058.95	8139.13	2252.15	10235.25
seg41	6226.78	5707.41	4975.38	6632.24	8412.04	6743.12	5303.16	8850.48

Cuadro 4.5: Tabla de resultados (segunda parte). Comparación utilizando el método de la transformación de un objeto a otro.

Forma	“Icono uno”	Porcentaje de Similitud
seg01	icono14	81.90 %
seg02	icono11	71.54 %
seg03	icono04	74.95 %
seg04	icono04	72.58 %
seg05	icono04	78.64 %
seg06	icono14	81.64 %
seg07	icono04	76.71 %
seg09	icono04	78.86 %
seg10	icono04	79.43 %
seg11	icono06	87.55 %
seg12	icono04	75.73 %
seg13	icono04	88.16 %
seg14	icono04	85.35 %
seg15	icono15	73.38 %
seg17	icono14	79.41 %
seg18	icono04	52.88 %
seg23	icono04	81.97 %
seg24	icono03	80.84 %
seg25	icono07	80.16 %
seg29	icono08	80.27 %
seg30	icono08	84.87 %
seg31	icono12	80.69 %
seg32	icono12	80.62 %
seg33	icono06	81.21 %
seg34	icono06	81.62 %
seg35	icono02	81.13 %
seg36	icono09	83.67 %
seg37	icono12	82.56 %
seg38	icono02	81.55 %
seg39	icono12	82.17 %
seg40	icono14	77.99 %
seg41	icono03	47.45 %

Cuadro 4.6: Porcentajes de similitud.

Conclusiones

En la búsqueda y reconocimiento de patrones se utilizan dos técnicas: segmentación de imágenes y comparación de formas. Para aplicar cada una de estas técnicas se han desarrollado diferentes métodos, cada uno de ellos guiados por alguna característica de las imágenes. Todos estos métodos tienen una amplia aplicación para resolver diversos problemas.

Algunos de los métodos existentes para la segmentación de imágenes fueron descritos aquí. Indudablemente existen más técnicas para lograr el segmentado pero generalmente se usa la aplicación de umbrales. Los métodos existentes no logran un segmentado al 100% y con frecuencia se utilizan varios para complementarse y lograr un resultado efectivo.

En el caso de la comparación de formas cada método tiene una técnica diferente debido a que su desarrollo se basa en alguna característica en particular de la forma, por ejemplo, los números de forma toman en cuenta el contorno, la compacidad el perímetro y el área, la correlación máxima los píxeles comunes, y la transformación de un objeto a otro la distancia entre píxeles. Estos métodos en general son eficientes y la utilización de cada uno de ellos depende del problema a solucionar. Los métodos descritos en este trabajo no son todos los que existen.

Los métodos de segmentación usados para resolver el problema planteado en este trabajo son: aplicación de umbrales complementado con la unión de bordes; y segmentación por regiones. El primero se usó por su eficiencia y se complementó con la unión de bordes porque en algunas imágenes los niveles de gris no definían bien a los objetos; y el segundo sólo se usó en casos en donde los contornos de los objetos estaban bien definidos. El uso de este último no fue excesivo debido a que es un método lento en su procesamiento. El método utilizado para la comparación de formas fue la transformación de un objeto a otro complementado con la compacidad. El uso de la compacidad permite clasificar a las formas en delgadas y gruesas y, si todos los símbolos que se desea buscar tienen cierto grado de grosor entonces las formas a comparar deben tener el mismo grado; por eso primero se hace una clasificación de las formas a comparar, para evitar comparaciones innecesarias con el

otro método usado. La razón de usar el método de la transformación de un objeto a otro es para mostrar la eficiencia y el comportamiento de dicho método resolviendo problemas de este tipo.

La realización de la segmentación no fue una tarea sencilla porque aún no se ha encontrado un método que realice este trabajo con un 100 % de efectividad, además de la diversidad de imágenes que se tiene, las cuales pueden contener características especiales, tales como, ruido o una gran variedad de niveles de gris; en donde hasta el momento sólo el ojo humano puede detectar formas.

El proceso propuesto aquí para buscar "iconos uno" puede ser utilizado para buscar cualquier otro tipo de símbolos. Este proceso puede verse como un procedimiento general para la búsqueda de cualquier símbolo.

Todos los algoritmos utilizados para el desarrollo de este trabajo fueron implementados con el lenguaje de programación de alto nivel: java.

El único problema que presenta el proceso para la búsqueda de símbolos mostrado en esta tesis es que requiere de varias horas de ejecución¹. Este requerimiento de tiempo de ejecución está relacionado con el método utilizado para la comparación de formas: transformación de un objeto a otro aplicando el algoritmo húngaro para encontrar apareos perfectos óptimos. La complejidad del algoritmo húngaro es de $O(n^3)$ [8], lo cual implica un consumo considerable de tiempo de ejecución para gráficas bipartitas de 1000 o más nodos (vértices).

A pesar del requerir bastante tiempo de ejecución el proceso propuesto en esta tesis se aplicó exitosamente para el caso de la búsqueda y el reconocimiento de los símbolos "iconos uno" en las imágenes de las culturas mesoamericanas, de acuerdo con lo que se reporta en el capítulo 4, es decir, en la imágenes mostradas para realizar el proceso se detectaron "iconos uno".

¹Para las pruebas se empleó una PC Pentium II con 64 MB de memoria RAM.

Apéndice A

Definiciones

A.1. Imagen digital

La representación computacional de una imagen es a través de una división finita de puntos de la misma. La división se hace formando una cuadrícula, así que la distancia horizontal y vertical entre cada punto es la misma.

La forma de representar la cuadrícula de una imagen bidimensional en una computadora es mediante una matriz; cada elemento de la matriz representa un pixel. La localización de un pixel en la matriz es por medio del número de renglón y columna y se representa con (i, j) , donde i es el renglón y j es la columna. Esta notación permite trasladar la imagen a un plano con ejes coordenados (x, y) (el eje x es el horizontal y el eje y es el vertical).

La localización de un pixel en la matriz es por medio de sus coordenadas, las cuales se obtienen con un desplazamiento de pixel en pixel, hacia abajo para llegar al renglón deseado, y hacia la derecha para llegar a la columna deseada. La localización de un punto en el plano es con un desplazamiento hacia arriba o abajo y hacia la derecha o izquierda. Situándose en cualquier posición en la matriz se pueden realizar desplazamientos hacia arriba o abajo y hacia la derecha o izquierda igual que en el plano, por lo tanto la matriz se puede ver como un plano.

Para obtener la posición de determinado pixel en el plano se utilizan las siguientes ecuaciones:

$$x = j - n \tag{A.1}$$

$$y = -(i - m) \tag{A.2}$$

donde (i, j) es la coordenada del pixel ubicado en la matriz, (x, y) corresponde a la coordenada del plano, (n, m) corresponde a la coordenada del centro de masa de

la imagen (m es el renglón y n es la columna); el origen del plano corresponde al centro de masa de la imagen.

Similarmente, la posición de un pixel que está representado como un punto en el plano se obtiene con las siguientes ecuaciones:

$$i = -y + m \quad (\text{A.3})$$

$$j = x + n \quad (\text{A.4})$$

donde cada una de las variables corresponden con la descripción dada anteriormente.

A.2. Pixel

En la sección anterior se define una imagen digital como una imagen particionada en un conjunto finito de puntos que forman una cuadrícula. A cada punto que forma parte de la imagen se le llama *pixel*.

Un pixel está caracterizado por dos elementos, el primero son las coordenadas que determinan la ubicación de dicho pixel en la imagen, y el segundo es la intensidad de luz, brillantez o luminosidad que tiene la imagen en este punto. La intensidad de luz se representa con un valor numérico.

A.3. Nivel de gris

El *nivel de gris* es el valor numérico que representa la intensidad de luz de un pixel. Generalmente este valor numérico está en el rango de 0 a 255 (usando 8 bits para representarlo) pero puede estar en un rango de 0 a 63 o de 0 a 1023 (usando 6 bits o 10 bits respectivamente para representarlo).

Si el nivel de gris de un pixel es 0 la imagen en este punto es negra, y si el nivel de gris es de 255 la imagen en este punto es blanca. Para los otros valores en el rango se distribuyen tonos de grises.

El nivel de gris también es conocido como: tono de gris, intensidad del tono de gris, intensidad de la imagen, brillantez, valor de imagen, luminosidad o valor del pixel.

A.4. Pixelización

Generalmente los objetos o formas que pertenecen a una imagen del mundo real son continuos, es decir, en la mayoría de los objetos su contorno es formado

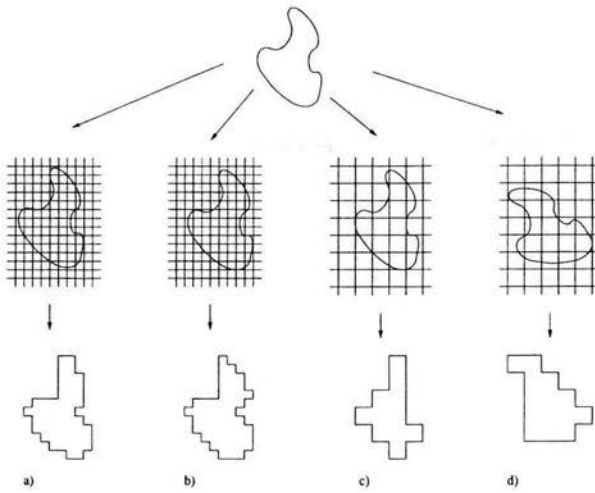


Figura A.1: Pixelización de un objeto. Las cuatro figuras son diferentes porque en A.1.a y A.1.b la posición del objeto sobre la cuadrícula no es la misma, la cuadrícula de la figura A.1.c es más grande que las anteriores, y en A.1.d la orientación del objeto no es igual a la de los demás.

por líneas curvas y/o líneas rectas; en los contornos escasamente aparecen esquinas bien formadas. Para el procesamiento digital o el reconocimiento de estos objetos se requiere que todo el contorno de dichos objetos sea discreto, es decir, que sólo esté formado de líneas rectas. Se sabe que una imagen digital está formada por píxeles, por lo tanto los objetos de la imagen también están formados por píxeles. Cuando se tienen los píxeles que forman parte de un objeto se dice que se tiene al objeto *pixelizado*; dicho de otra manera, cuando se tiene un objeto representado de manera digital (formado de píxeles) se dice que se tiene la *pixelización* del objeto.

Para realizar la pixelización de un objeto hay que tener en cuenta algunas consideraciones, tales como, posición, orientación y tamaño de los píxeles. En la figura A.1 se tiene a un objeto y cuatro de sus diferentes formas pixelizadas. Note que las formas de las figuras A.1.a y A.1.b son diferentes debido a que la posición del objeto sobre la cuadrícula no es la misma. La forma de la figura A.1.c difiere de las anteriores debido a que el tamaño de la cuadrícula es más grande, y la forma de la figura A.1.d también difiere debido a que la orientación del objeto sobre la cuadrícula es diferente a la del objeto original.

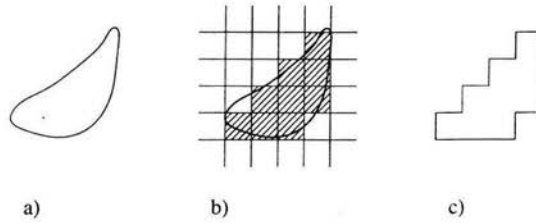


Figura A.2: a) Un objeto. b) Pixelización del objeto en A.2.a. c) Forma pixelizada del objeto en A.2.a.

La solución a estas diferencias de representación se encuentra estandarizando la manera de pixelizar a los objetos. La estandarización requiere: que el tamaño de la cuadrícula siempre sea el mismo (generalmente se toma una unidad de tamaño); que la orientación de los objetos sea la misma (en este caso se toma a los ejes principales como invariante bajo rotación, y éstos deben coincidir con los ejes coordenados); y que los objetos coincidan en posición en el centro de masa (que sean invariantes bajo traslación).

Al momento de realizar la pixelización se dice que un pixel es parte del objeto si dicho objeto cubre completamente al pixel o cubre más del 50% de su tamaño. En la figura A.2 se muestra un ejemplo de esta situación: la figura A.2.a es un objeto cualquiera, la figura A.2.b es la pixelización del objeto, y la figura A.2.c es el objeto pixelizado.

Apéndice B

Filtros

B.1. Filtro

En algunas ocasiones las imágenes que se obtienen, ya sea por medio de una cámara fotográfica o escáner u otro medio, no son del todo perfectas o como se desean. Es decir, la intensidad de luz, el contraste o manchas pueden ser factores para que las imágenes no se vean bien o no sean como se requieren. A estas características no deseadas se les conoce como *ruido*.

Algunos tipos de ruido que frecuentemente aparecen en las imágenes son:

- *Ruido de sal y pimienta*: contiene ocurrencias aleatorias de valores de intensidad blanco y negro.
- *Ruido de impulso*: contiene solamente ocurrencias aleatorias de valores de intensidad blanco.
- *Ruido Gaussiano*: contiene variaciones de intensidad que son distribuidos como una función Gaussiana o distribución normal.

Existen métodos para arreglar imágenes con ruido llamados *filtros*. Un filtro es un operador que transforma una imagen. Ambas imágenes, la original y la filtrada, son muy similares en cuanto a su geometría. Lo que hace el filtro es cambiar el valor de algunos píxeles para modificar su forma, teniendo así, una mejor apreciación de la imagen.

B.2. Modificación del histograma

Uno de los filtrados más sencillos es mediante la modificación del histograma.

Algunas imágenes contienen distribuidos desuniformemente valores de grises, y es común encontrar estas variaciones en un rango pequeño de la imagen. La equalización del histograma es un método que modifica el contraste de una imagen para que la distribución de grises sea uniforme. En conclusión, la modificación del histograma refuerza la calidad subjetiva de una imagen.

La manera de equalizar un histograma es escalando la imagen. Por ejemplo, si se tiene los pixeles en el rango $[a, b]$, estos se expanden para llenar el rango $[z_1, z_k]$. La ecuación que permite mapear el valor de un pixel del rango original al nuevo rango es:

$$z' = \frac{z_k - z_1}{b - a}(z - a) + z_1 = \frac{z_k - z_1}{b - a}z + \frac{z_1 b - z_k a}{b - a} \quad (\text{B.1})$$

donde z es el pixel de la imagen original y z' es el pixel de la nueva imagen.

El problema con este esquema es que cuando el histograma se modifica conforme a la ecuación B.1, el histograma resultante tiene huecos.

B.3. Filtros lineales

Un filtro lineal es un filtro que cambia la intensidad de cada pixel en la imagen con respecto a sus coordenadas. En la imagen filtrada, el valor para los pixeles a los cuales se les cambia la intensidad es el promedio de valores o combinación lineal de las intensidades de pixeles localizados a su alrededor. Un filtro lineal es a menudo usado para cambiar las características de frecuencias de la imagen.

Los filtros lineales son buenos para quitar ruido Gaussiano, y en muchos casos los otros tipos de ruido.

Un filtro lineal que es muy simple, se implementa asignando el valor promedio de todos los valores de los pixeles vecinos a un pixel dado. La cantidad de vecinos controla la cantidad a filtrar. Una cantidad de vecinos grande corresponde a una cantidad grande a filtrar (un filtro grande). Utilizar un filtro grande puede proporcionar reducciones grandes de ruido, pero puede ocasionar pérdida de detalles en la imagen.

B.4. Filtro media

El trabajo que realiza el filtro llamado media es reemplazar el valor de cada pixel con la media de los valores de los pixeles vecinos de dicho pixel.

Los filtros media son muy efectivos para remover ruido de sal y pimienta y ruido de impulso. Con el filtro media se retienen los detalles en las imágenes porque no

dependen de valores, los cuales son significativamente diferentes de los valores de los vecinos.

La implementación de este filtro es mediante los siguientes pasos:

1. Ordenar los pixeles de forma ascendente de acuerdo a sus niveles de gris.
2. Seleccionar el valor del pixel ubicado a la mitad de la lista como nuevo valor para el pixel a modificar.

Normalmente, una cantidad de vecinos impar es usada para calcular la media. Sin embargo, si el número de pixeles es par, la media se toma como el promedio de los dos pixeles que se encuentran a la mitad de la lista ordenada.

B.5. Filtro Gaussiano

El tema de filtros Gaussianos es muy amplio, aquí se expone, sin detalles, sobre este tema.

Los filtros Gaussianos [13] son una clase de filtros lineales con pesos escogidos de acuerdo a la forma de la función Gaussiana. EL filtro Gaussiano es muy bueno para remover ruido que tiene la forma de una distribución normal. El hecho de que los pesos del filtro son escogidos de una distribución Gaussiana y que el ruido es también distribuido como una Gaussiana es sólo coincidencia.

La función Gaussiana en una dimensión es:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (\text{B.2})$$

donde la Gaussiana cubre el parámetro σ que determina el ancho de la Gaussiana.

Para el procesamiento de imágenes, la función Gaussiana discreta es:

$$g[i, j] = e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (\text{B.3})$$

donde i y j son las coordenadas de los pixeles. Esta función es usada como un filtro.

Las funciones Gaussianas tienen propiedades que hacen particularmente útil el procesamiento de visión. Estas propiedades indican que los filtros Gaussianos son efectivos para filtros de bajo flujo. Su implementación es eficiente, y pueden ser usados efectivamente en aplicaciones de visión práctica.

Las propiedades son las siguientes:

- En dos dimensiones, las funciones Gaussianas son simétricas rotacionalmente. La propiedad de simetría rotacional implica que un filtro Gaussiano no se inclina, en el eje de detección, en una dirección en particular.

- La función Gaussiana tiene la forma de una campana. No es conveniente que un filtrado Gaussiano reemplace cada pixel de la imagen con el valor promedio de sus vecinos, tal que el ancho dado a un vecino se decremente monótonamente con la distancia de un pixel central. Esta propiedad es importante porque una operación de filtrado que otorga más significado a pixeles más lejanos, distorsionará las características de la imagen.
- El ancho, y por lo tanto el grado de filtrado, de un filtro Gaussiano es parametrizado por σ , y la relación entre σ y el grado de filtrado es muy simple. Una σ grande implica un filtro Gaussiano grande y un filtrado extenso. El grado de filtrado se puede ajustar.
- Filtros Gaussianos grandes pueden ser implementados muy eficientemente porque las funciones Gaussianas son separables, es decir, filtros Gaussianos grandes pueden ser implementados por repeticiones aplicadas a filtros Gaussianos pequeños. Una función Gaussiana en dos dimensiones puede ser desarrollada por una Gaussiana en una dimensión, entonces se obtiene el resultado con el mismo filtro con orientación en dirección ortogonal a la Gaussiana, usada en la primer etapa. El orden de computabilidad de la implementación de filtros Gaussianos en dos dimensiones crecen linealmente en el ancho del filtro en lugar de crecer cuadráticamente.

Una aproximación en el diseño de filtros Gaussianos es calcular la plantilla ancha directamente de la distribución Gaussiana discreta:

$$g[i, j] = ce^{-\frac{i^2+j^2}{2\sigma^2}} \quad (\text{B.4})$$

donde c es una constante normalizada. Reescribiendo ésta se tiene:

$$\frac{g[i, j]}{c} = e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (\text{B.5})$$

y cambiando un valor para σ^2 , se puede evaluar sobre una plantilla de $n \times n$, para la cual el valor en $[0,0]$ es igual a 1. Sin embargo, se diseña el filtro ancho para valores enteros para facilitar el cálculo. Por lo tanto, se toma el valor en una de las esquinas del arreglo, y se escoge k tal que este valor se convierte en 1.

Bibliografía

- [1] D. H. Ballard and C. M. Brown, *Computer Vision*, Englewood cliffs, New Jersey: Prentice Hall, 1982.
- [2] Rubén Bonifaz Nuño, *Olmecas: Esencia y Fundación*, El colegio nacional, páginas 5-13, 1992.
- [3] A. I. Borisenko, I. E. Tarapov, *Vector and Tensor Analysis with applications*, Dover Publications Inc. New York, 1968.
- [4] Ernesto Bribiesca, *Measuring 2-D Shape Compactness Using the Contact Perimeter*, Computers Math. Applic. Vol. 33, No. 11, páginas 1-9, 1997.
- [5] Ernesto Bribiesca and Adolfo Guzmán, *How To Describe Pure Form And How To Measure Differences In Shapes Using Shape Numbers*, Pattern Recognition, Pergamon Press, Vol. 12, No. 2, páginas 101-112, 1980.
- [6] Ernesto Bribiesca and R. G. Wilson, *A Measure of 2D Shape-of-Object Dissimilarity*, Appl. Math. Lett., Vol. 10, No. 6, páginas 107-115, 1997.
- [7] Kennet R. Castleman, *Digital Image Processing*, Englewood cliffs, New Jersey: Prentice Hall, 1995.
- [8] James R. Evans, Edward Minieka, *Optimization Algorithms For Networks and Graphs*, New York: Marcel Dekker, Inc., segunda edición, 1992.
- [9] H. Freeman, *On the Encoding of Arbitrary Geometric Configurations*, Institute of Radio Engineers, Transactions on Electronic Computers, EC-10, páginas 260-268, 1961.
- [10] Rafael C. González and R. E. Woods, *Digital Image Processing*, Reading Masachuset: Addison-Wesley Publishing Company, 1992.

-
- [11] Robert M. Haralick, Linda G. Shapiro, *Computer and Robot Vision*, Reading Masachusetts: Addison-Wesley, 1993.
 - [12] Anil K. Jain, *Fundamentals of Digital Image Processing*, Englewood cliffs, New Jersey: Prentice Hall, 1989.
 - [13] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, *Machine Vision*, New York: McGraw-Hill, 1995.
 - [14] H. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Q., Vol. 2, páginas 83-97, 1955.
 - [15] Martin D. Levine, *Vision in Man and Machine*, New York: McGraw-Hill Publishing Company, 1985.
 - [16] H. Samet, *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*, Addison-Wesley, Reading, MA, 1990.