

5



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

APLICACION DE UN CODIGO DE CADENA EN 3D PARA LA REPRESENTACION UNIDIMENSIONAL DE IMAGENES DIGITALES.

T E S I S

QUE PARA OBTENER EL TITULO DE LICENCIADA EN CIENCIAS DE LA COMPUTACION PRESENTA MARIBEL HERNANDEZ ROSALES



FACULTAD DE CIENCIAS UNAM

DIRECTOR DE ESTUDIOS PROFESIONALES DR. ERNESTO BRIBIESCA CORREA



FACULTAD DE CIENCIAS SECCION ESCOLAR

2003





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.



SECRETARÍA NACIONAL
DE EDUCACIÓN PÚBLICA

M. EN C. ELENA DE OTEYZA DE OTEYZA
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"Aplicación de un código de cadena en 3D para la representación
unidimensional de imágenes digitales"

realizado por **Maribel Hernández Rosales**

con número de cuenta 9304227-3 , quién cubrió los créditos de la carrera de **Ciencias de la
Computación**

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Proprietario

Dr. Ernesto Bribiesca Correa

E. Bribiesca

Propietario

Dr. Pedro Miramontes Vidal

P. Miramontes

Propietario

Mat. Carlos Velarde Velázquez

Carlos Velarde

Suplente

M. en C. Hermilo Sanchez Cruz

H. Sanchez Cruz

Suplente

M. en C. Pablo Ignacio Castillejo García

P. Castillejo

Consejo Departamental de Matemática



M. en C. *Guadalupe B. Ibarra* ~~Guadalupe B. Ibarra~~ *Guengaitia* González

FACULTAD DE CIENCIAS
CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

A MI MAMÁ :

Te quiero dedicar este trabajo ya que sin tu ayuda, comprensión y cariño no hubiera sido posible. Gracias por todo tu apoyo y por ser mas que mi madre una gran amiga. Te quiero mucho cachetoncita.

A MI PAPÁ :

Te dedico este trabajo por ser ése hombre tan importante en mi vida. Gracias por ayudarme siempre, gracias por confiar en mí, gracias por apoyarme en este camino, gracias por todo. Te quiero mucho.

A MIS HERMANOS :

A mi hermano Alfredo: te agradezco infinitamente por ser mi gran apoyo, por estar ahí siempre que te necesito, por ayudarme y compartir conmigo todos mis triunfos y todas mis derrotas. Te quiero mucho. A mi hermano Alonso: gracias por cada palabra de apoyo, gracias por siempre querer ayudarme, gracias por todos esos detalles que sin darte cuenta fueron grandes logros. Te quiero mucho. A mi hermano Iván: gracias por ser mi compañero de juegos, de sueños y de ilusiones. Gracias por compartir conmigo cada una de mis lágrimas y cada una de mis sonrisas. Te quiero un chorro.

A ALFREDO :

Gracias por todo el amor que me has dado. Gracias por compartir conmigo toda la aventura de la universidad. Gracias por abrirme tu corazón y permitirme entrar al tuyo. Gracias por toda tu ayuda, comprensión y apoyo incondicional. Gracias por todo. Te amo.

A MIS ABUELITOS :

A mi abuelita Serafina: por ser una mamá mas para mí. Por tanto cariño y consejos. La quiero mucho. A mi abuelito Mon: gracias por todo su cariño. Gracias por todas sus historias. Lo quiero mucho. A mi Mama Chona: gracias por todos esos ratos que me brinda de cariño. La quiero mucho.

A MIS AMIGAS Y AMIGOS :

Quiero agradecer especialmente a Ale por ser mas que mi amiga una hermana en la que puedo confiar siempre. Gracias por estar a mi lado todos los momentos que te necesité. Te quiero mucho. Agradezco a Arge, Idan y Gris por ser mis hermanas de la infancia y por seguir dándome tanto cariño a lo largo de todos estos años. Gracias por todo su apoyo. Las quiero mucho. Gracias a Gaby y Eli por ayudarme siempre que las necesitaba. Gracias por ser mis compañeras de juegos y ahora mis amigas. Las quiero mucho. Gracias también a mis demás amigas y amigos que han formado parte importante en mi vida: Erica, Sandra, Bety, Griselda, Karla, Jazmin, Nina, Celina, Cesarín, Emi, Sergio, Roberto, Aguayón, Mario, Trompis, Pablo, Arturo, Nawar, Rafa, a toda la gente de La Parota que me ha dado tanto cariño en tan poco tiempo y a todos los que no aparecen en esta lista y que se han llevado un pedazo de mi corazón. Los quiero mucho a todos.

A MIS SINODALES :

A Ernesto Bribiesca: gracias por darme la oportunidad de trabajar con usted. Gracias por guiarme y acercarme al mundo de la investigación. Gracias por sus enseñanzas, su paciencia y todo su apoyo. A Carlos Velarde: gracias por todo el tiempo que me brindo con sus enseñanzas. Gracias por ser la persona que siempre me ayudó con sus conocimientos teóricos y prácticos. Gracias por todo. A Pedro Miramontes: gracias por ayudarme siempre que lo necesite, gracias por sus consejos, gracias por todo su apoyo, por sus enseñanzas, pero sobre todo, mil gracias por todo su cariño. Lo quiero mucho. A Hermilo Sanchez: gracias por resolver siempre mis dudas. Gracias por el tiempo que me brindaste y compartiste conmigo dentro y fuera del ambiente académico. Gracias por las galletas y tu cariño. A Pablo Castillejo: gracias porque eres mas que un sinodal un gran amigo. Gracias por los libros y la música. Gracias por compartir conmigo tristezas y alegrías. Gracias por todo tu cariño.

A MI UNIVERSIDAD Y MAESTROS :

Gracias por haberme ayudado para mi formación profesional con sus conocimientos y experiencias. Un agradecimiento especial a Hector Benitez y Sonia Favela.

A DIOS :

Gracias por la vida y por permitirme seguir adelante. Te pido que siempre ilumines mi camino.

INDICE

	página
INTRODUCCION	1
CAPITULO 1	
REPRESENTACIÓN DE CURVAS: TRABAJOS PRECEDENTES	10
1.1. Polilíneas	10
1.2. Descriptores de Fourier	12
1.3. Secciones cónicas	13
1.4. B-splines	14
CAPITULO 2	
REPRESENTACIÓN DE FORMAS EN 2D Y 3D	16
2.1 Cadena de Freeman en 4 direcciones	16
2.2 Cadena de Freeman en 8 direcciones	17
2.3 Números de forma	18
2.3.1 Qué es una forma	18
2.3.2 Formas discretas y continuas	19
2.3.3 Convirtiendo una forma continua en discreta	19
2.3.4 Excentricidad	20
2.3.5 Derivada de la cadena de Freeman en 4 direcciones	21
2.3.6 Números de forma	22
2.3.7 El orden de un número de forma	24
2.4 Una estructura geométrica para formas bidimensionales	24
2.4.1 Conversión de una curva continua en una forma discreta	26
2.4.2 Independencia de translación	27
2.4.3 Independencia de rotación	27
2.4.4 Cambio de escala	27
2.4.5 Independencia de tamaño	28
2.6 Código de cadena basado en vértices (CCV)	29

2.6.1 Normalización con respecto al punto de inicio	30
2.6.2 Independencia de rotación	31
2.6.3 Independencia de transformación espejo	32
2.6.4 El CCV para formas compuestas de celdas triangulares y celdas hexagonales	33
2.7 Esquemas de representación para curvas en 3D	33
2.7.1 Métodos tridimensionales y n-dimensionales	35

CAPITULO 3

CÓDIGO DE CADENA PARA REPRESENTAR CURVAS EN 3D	43
3.1 Conceptos y definiciones	44
3.2 Cómo obtener la cadena de una curva dada	45
3.3 Independencia de Rotación	47
3.4 El inverso de una cadena	48
3.5 Normalización respecto al punto de inicio para curvas abiertas	50
3.6 Normalización respecto al punto de inicio para curvas cerradas	50
3.7 Transformación espejo (reflexión en un plano)	51
3.8 Comparación de curvas	51

CAPITULO 4

APLICACIÓN DEL CÓDIGO DE CADENA EN 3D PARA REPRESENTACIÓN DE IMÁGENES DIGITALES	54
4.1 Formato de la imagen	54
4.2 Conversión de la imagen a una matriz bidimensional	55
4.3 Binarización	57
4.4 Obtención de la cadena en 3D	61

CAPITULO 5

RESULTADOS, APLICACIONES Y PROPUESTAS DE TRABAJOS A FUTURO	70
5.1 Resultados	70
5.1.1 Compactación	70

5.1.2 Búsqueda de plantillas: uso de un editor de cadenas	71
5.1.3 Análisis de códigos de cadenas	74
5.2 Aplicaciones y propuestas de trabajos a futuro	80
5.2.1 Búsqueda de plantillas	86
5.2.2 Transformación de cadenas	88
5.2.3 Identificación de rasgos	90
CONCLUSIONES	93
APÉNDICE	103
REFERENCIAS	111

INTRODUCCION.

El estudio de las *formas* de los objetos del mundo real es importante en muchas áreas, tales como: biología, paleontología, geografía, medicina, química, geología, anatomía, botánica, y otras más; en las cuales se busca obtener información a partir de una representación de los objetos, que permita una mejor manipulación de ellos para así facilitar la *extracción* (ver apéndice) de datos útiles, como por ejemplo: regiones, rasgos, alteraciones, similitud, regularidades, etc.

En esta tesis se propone un método para representar una forma como cadena unidimensional, mediante la aplicación de un código de cadena a imágenes digitales. Este método permite la manipulación y extracción de información de los objetos representados por cadenas.

Para casi cualquier ser humano es natural identificar a los objetos por su forma, ya que por medio de ésta puede definir fácilmente métodos de clasificación, o simplemente asociar grupos de acuerdo a lo que ve. Es por esto que las formas juegan un papel importante en la vida cotidiana. A continuación se presentan algunos ejemplos de cómo la forma de los objetos ha participado en el desarrollo de la ciencia.

Muchos fenómenos naturales y procesos biológicos describen trayectorias o formas; por ejemplo, la doble hélice de ADN descrita en 1953 por Watson y Crick, ha llegado a ser la piedra angular de la biología molecular. Sin embargo, la forma asumida por casi todos los seres vivos es llamada B ADN, aunque bajo ciertas condiciones, el ADN puede tomar otras formas: la forma A cuando es deshidratado, la forma Z en el seno de ciertos solventes (ver [33]).

Aunque en las tres formas el ADN consiste de dos cadenas de nucleótidos, entrelazadas una con otra, lo cual traza una suave hélice, el estudio de las mismas permite la clasificación de ellas, como se muestra en la figura 1.



(a)



B-ADN



Z-ADN



A-ADN

(b)

Fig. 1. (a) Forma conocida del ADN. (b) Tres formas del ADN.

Un área para la cual son muy importantes las formas es la botánica. El difícil trabajo de catalogar y clasificar el gran número de plantas existentes contribuyó al desarrollo en el campo de la taxonomía de las plantas. Los taxonomistas trataban de formar grupos y dar nombres a las especies. Inventaban sistemas de clasificación para agrupar los organismos de acuerdo con sus similitudes en la estructura o comportamiento. Estos sistemas ayudaron a aumentar el interés por estudiar la estructura de las plantas. La mayoría de estos sistemas se basaba en la apariencia exterior de estos organismos. Una especie estaba relacionada con otra, según el grado de similitud de sus estructuras. Como resultado, la clasificación de las plantas en la actualidad se da de la siguiente manera: división, orden, familia, género y clase. Esta clasificación está basada, como ya se mencionó, en la estructura o forma de las plantas. En la figura 2 se muestran 8 hojas de roble, que por su forma pueden ser clasificadas de una manera sencilla. En esta figura se puede observar que la forma o estructura de las hojas tiene un papel importante en la clasificación de las plantas (ver [29]).

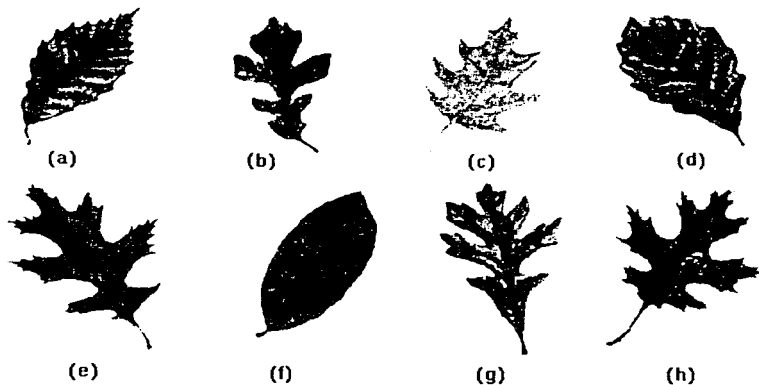


Fig. 2. Ocho tipos de hojas de roble: (a) roble de bellotas: las hojas están toscamente dentadas en el borde; (b) roble de codillo: la hoja está cortada casi a la mitad de la nervadura; (c) roble rojo: el borde está cubierto de púas y disminuye gradualmente desde la base; (d) roble de los pantanos: el borde de las hojas es dentaondulado; (e) roble de alfilerillas: el lóbulos está cubierto de púas; (f) roble Shingle: la hoja no tiene lóbulos cortados; (g) roble blanco: la hoja es angosta y el borde tiene estrechas sinuosidades; (h) roble escarlata: los lóbulos son dentados y cubiertos de púas.

Como en la botánica, en la zoología también es muy importante la forma de los organismos para su clasificación, por ejemplo hay animales con aletas y otros con alas, por lo que es natural colocar las especies con alas en un grupo, y en otro todas las especies de animales con aletas, aunque esta clasificación no es muy eficiente, puesto que solo se basa en su apariencia estructural, sin embargo ayuda a los taxonomistas a comenzar una clasificación de especies. Aunque la clasificación de las especies se basa también en los pulmones, branquias, etc., en ésta influye la forma de dichos órganos.

Otro ejemplo sencillo es el análisis de la huella que deja una víbora al arrastrarse. Si se quisiera conocer qué tipo de víbora dejó cierta huella, se podría ver si la huella es angosta o ancha, o si la trayectoria es recta u ondulada, si hay alguna marca en particular que se repita a lo largo de la huella, etc.; de esta manera se podrán descartar muchos tipos de víboras, así el conjunto restante será reducido, aunque seguirá siendo grande.

En las células se presenta una manera de diferenciación sencilla por medio de las formas. Una célula vegetal puede fácilmente diferenciarse de una célula animal por medio de su forma y de las formas dentro de cada célula, como se muestra en la figura 3. Así,

cuando se analiza tejido de manera microscópica, se puede decir de cual reino proviene (ver [30]).

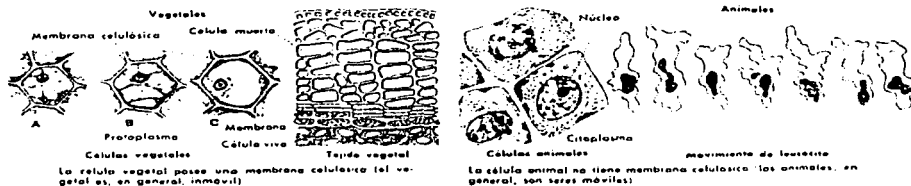


Fig. 3. Caracteres diferenciales entre ambos reinos.

En paleontología las *formas* también han tenido gran importancia, como en la historia de los reptiles (ver [30]). Estos se consideran como el grupo de enlace entre los primeros vertebrados terrestres, es decir, los anfibios, y los superiores: aves y mamíferos.

Los reptiles aparecieron en el Paleozoico medio superior y los restos de mayor importancia son por lo común los *esqueletarios*, y entre ellos el de mayor interés es el cráneo. En reptiles fósiles hay cuatro tipos de cráneo, según la forma y la presencia de cavidades temporales, a saber: cráneo *anápsido*, cuando solo posee las cavidades orbitarias; cráneo *parápsido*, cuando posee, además de las cavidades orbitarias, una cavidad a cada lado, ubicada entre el *hueso escamoso* y el *postorbitario*; cráneo *sinápsido*, cuando el hueso escamoso y el postorbitario se unen por encima de la cavidad ya mencionada para el cráneo parápsido; cráneo *diápsido*, cuando posee además de las cavidades orbitarias, dos cavidades temporales más a cada lado del cráneo y el hueso escamoso se une con el postorbitario en medio de ambas cavidades. Estos caracteres craneales, mostrados en la figura 4, han servido de base para la clasificación de los reptiles.

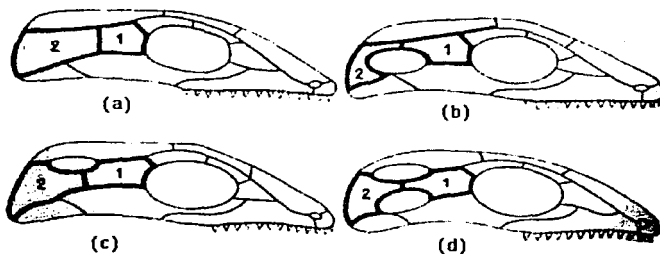


Fig. 4. Esquema de los tipos de cráneo de reptiles fósiles y actuales: (a) anápsido; (b) sinápsido; (c) parápsido; (d) diápsido. 1. Hueso escamoso. 2. Hueso postorbitario.

Un área muy importante en la actualidad y que se basa en el estudio de las formas es la antropología. La antropología es una ciencia que estudia al ser humano en sus distintos aspectos, considerándolo aisladamente o como parte de un grupo cultural, político o meramente estadístico. Parte de esta ciencia es llamada antropografía o antropología física, la cual estudia la descripción del ser humano, estático, o en sus funciones. Dejando a un lado los valores métricos, para el antropólogo resultan de interés la forma de la cabeza, partes de la misma (ojos, nariz, fosas nasales, boca, orejas) y la piel (ver [30]).

Al analizar las formas de distintas partes del cuerpo, se pueden conocer raza, cultura, nacionalidad y otras características de la persona en estudio. En la figura 5, se muestran dos tipos de ojos, de acuerdo a la raza de la persona. En las figuras 6-9 se muestran los tipos perfil, fosas nasales, boca y oreja, y su clasificación por medio de la forma.

Como se puede observar en éstas figuras, el estudio de las formas humanas nos da una descripción individual por cada ser humano.



Fig. 5. Dos tipos de ojo de acuerdo a su forma: (a) ojo mongólico; (b) ojo europeo.

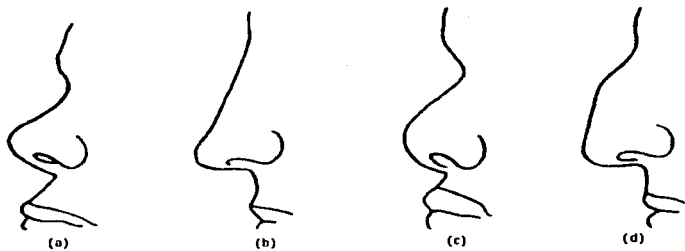


Fig. 6. Tipos de perfiles de acuerdo a su forma: (a) cóncavo; (b) recto; (c) convexo; (d) quebrado.

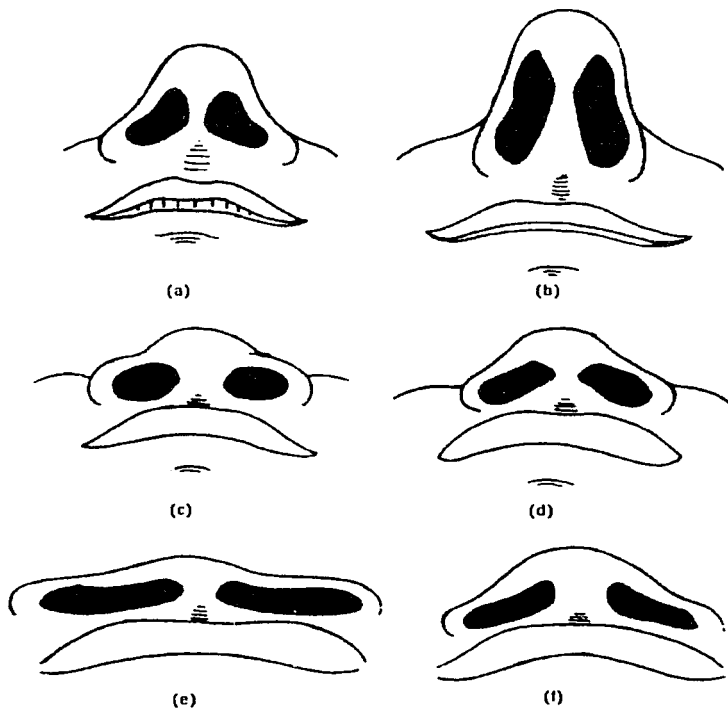


Fig. 7. Orificios de la nariz según Topinard (ver [30]): (a) y (b) raza europea; (c) y (d) razas amarillas; (e) y (f) razas negras.

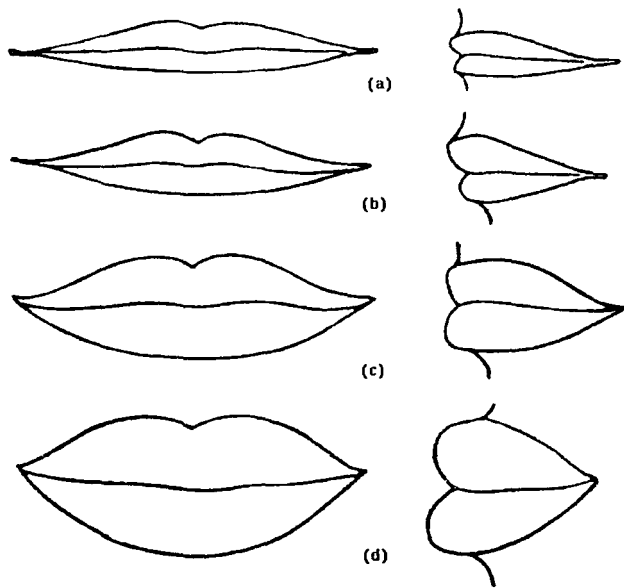


Fig. 8. Tipos de labios de acuerdo a su forma: (a) muy finos; (b) mediano; (c) grueso; (d) muy grueso o abultado.

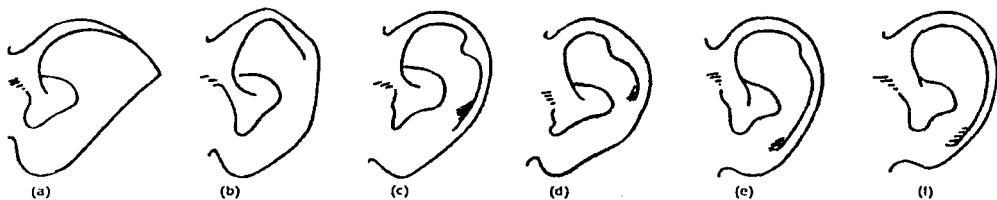
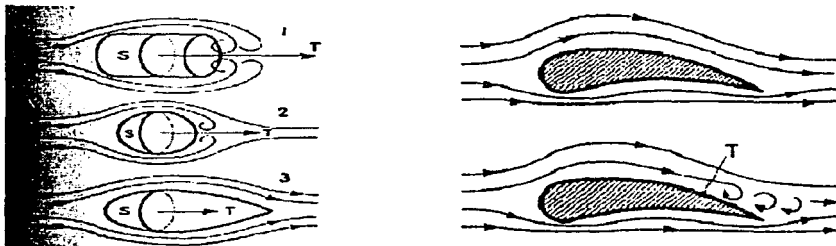


Fig. 9. Clasificación de las orejas según la punta de Darwin (ver [30]): (a) macaquiforme; (b) cercopitiforme; (c) con punta de Darwin aguda; (d) con tubérculo de Darwin; (e) con tubérculo de Darwin vestigial; (f) sin tubérculo de Darwin.

En otras áreas, como es la aerodinámica, el estudio de las formas también juega un papel importante.

Por ejemplo, al estudiar la forma de las alas de los aviones, se deben tomar en cuenta la formación de torbellinos de aire y se debe buscar la forma de reducirlos, para así agilizar la rapidez con la que avanza el avión. En la figura 10 se muestran tres tipos de alas de avión en las cuales se puede apreciar la ventaja de usar ciertas formas sobre otras. Se puede observar que las formas angulosas provocan torbellinos porque la vena de aire se despega de la superficie del cuerpo; se reduce la formación de torbellinos por el empleo de formas redondeadas y fusiformes. Los cuerpos 1, 2 y 3 tienen la misma sección perpendicular al viento. El cilindro limitado por planos opone la mayor resistencia; la forma de huso, la menor.

En la figura 11 se puede apreciar que dando al ala un perfil ahusado se reducen los torbellinos que se forman, a grandes velocidades, en el borde de salida, detrás del punto de transición.



Figs. 10 (izquierda) y 11 (derecha). Eliminación de los torbellinos por medio de formas y dispositivos especiales en las alas (T: punto de transición).

Hasta aquí se han visto diversos ejemplos de cómo influyen las formas de los objetos en diferentes áreas: para clasificación de especies, reconocimiento de razas, mejoramiento de tecnología, etc.

El primero en representar las *curvas* matemáticamente (la definición de curva es: una figura obtenida de extender o alargar, contraer, doblar, etc., un segmento de línea y una curva cerrada define un contorno o una región), fue René Descartes, quien introduce las

coordenadas rectangulares (conocidas como cartesianas) y, merced a éstas, la representación gráfica de las funciones; es decir, establece una correspondencia entre el ente analítico $y=f(x)$ y los puntos de una curva determinados por los pares de números x y y . Descartes también clasifica las curvas según el grado de las ecuaciones que las representan.

A partir de Descartes, varios autores han buscado representar formas y regiones por medio de curvas, usando diferentes métodos: coordenadas rectangulares, coordenadas polares, coordenadas cilíndricas, matrices, etc. Sin embargo estos métodos dependen del plano en el cual se está representando la curva. Por esta razón surgió la necesidad de estudiar nuevos métodos de representación de formas por medio de curvas que sean independientes bajo rotación, translación, punto de inicio, cambio de escala y transformación espejo.

En el capítulo 1 se muestran algunos métodos clásicos que se han usado en la representación de curvas y formas.

En el segundo capítulo se muestra el estado del arte en la representación de formas en dos dimensiones y tres dimensiones.

En el capítulo 3 se introducen los conceptos y definiciones del código de cadena que se aplica en esta tesis, además se muestran las propiedades más importantes de este código: independencia de translación y rotación, normalización respecto al punto de inicio, inverso y transformación espejo. Enseguida y como contribución importante a este trabajo, en el capítulo 4 se introduce el método para la aplicación del código de cadena mostrado en el capítulo 3 para la representación unidimensional de formas de imágenes digitales.

Por último, en el capítulo 5 se muestran algunos resultados importantes y aplicaciones realizadas, además se proponen algunos trabajos a futuro.

Finalmente en las conclusiones se da un panorama general de este trabajo y de las aportaciones aquí realizadas y se muestran algunas imágenes con su respectiva representación en cadena.

CAPITULO 1.

REPRESENTACIÓN DE CURVAS: TRABAJOS PRECEDENTES

En este capítulo se mostrarán algunos métodos clásicos que se han usado en la representación de curvas y formas.

1.1 Polilíneas

La representación de un segmento de línea mediante sus dos puntos extremos se extiende fácilmente a una *polilínea*, la cual representa una concatenación de segmentos de línea como una lista de puntos. Así, la lista de puntos x_1, x_2, x_3 representa la concatenación de los segmentos de línea de x_1 a x_2 y de x_2 a x_3 . Si el primer punto es el mismo que el último, se representa una frontera cerrada.

Las polilíneas pueden aproximar útilmente curvas a cualquier grado de precisión. Encontrar una aproximación poligonal buena para una curva dada, básicamente involucra métodos de segmentación. El problema es encontrar esquinas o vértices que produzcan la mejor polilínea (para mas detalles ver [21]-[26]).

Si se conoce que una frontera (representada sobre una rejilla discreta) es una secuencia de segmentos lineales, entonces, ésta es especificada mediante sus vértices. Para encontrarlos, se puede buscar a lo largo de la frontera, monitoreando el ángulo entre dos segmentos de línea. Dado un punto x_i sobre la frontera, un segmento se encuentra entre los puntos x_{i-1} y x_i y el otro segmento entre x_i y x_{i+1} . Cuando el ángulo entre estos dos segmentos alcance un máximo sobre algún umbral, un vértice es declarado en el punto actual. En la figura 1.1 se muestra un procedimiento para la búsqueda de dichos vértices.

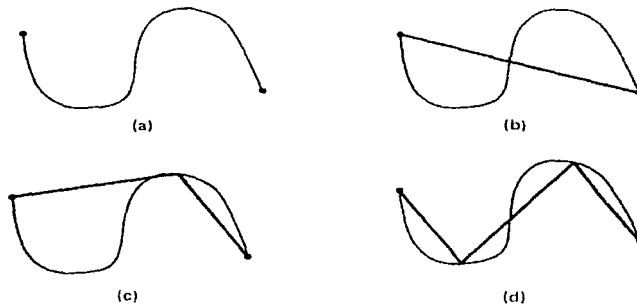


Fig. 1.1. Procedimiento en la representación de una curva por medio de una polilínea (figura tomada de [17]).

Las soluciones de *banda de tolerancia* sitúan un punto en un lado de la curva a una distancia de error máximo permisible y después encuentran el segmento más largo de la curva que se encuentre completamente entre líneas paralelas a través de dos puntos, un ejemplo de este proceso se muestra en la figura 1.2 (para mayor referencia ver [23]).

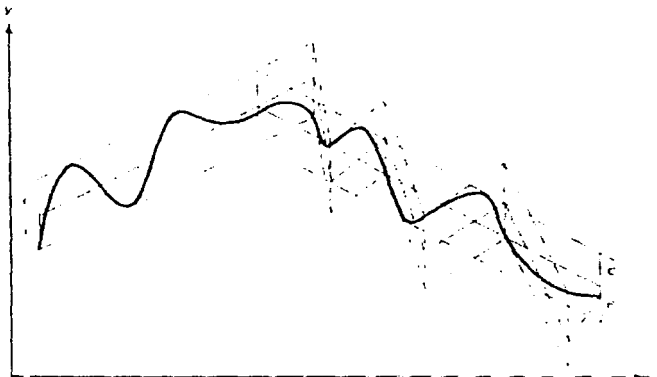


Fig. 1.2. Solución simple de banda de tolerancia (líneas punteadas). Una mejor solución (líneas continuas). (Figura tomada de [17]).

Una aproximación de una curva con una polilínea con error de longitud mínima (a lo más un pixel de error) es mostrada en [24]. Cada pixel en la curva es considerado un cuadrado y la estructura de pixeles resultante es *4-conectada* (ver apéndice). La aproximación describe a la forma como un hilo elástico situado en la estructura de pixeles (ver figura 1.3). Este método tiene dificultades con curvas que son relativamente abruptas al tamaño de la rejilla.

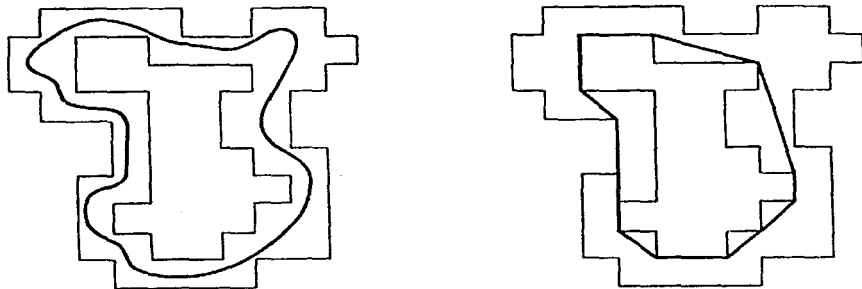


Fig. 1.3. Polilínea de longitud mínima (figura tomada de [17]).

1.2. Descriptores de Fourier

Los descriptores de Fourier representan la frontera de una región como una función periódica que puede expandirse en una serie de Fourier. Existen varias parametrizaciones resumidas en [27]. Estas descripciones proporcionan una precisión en la representación de la forma mejorando con el número de coeficientes que se incluyan.

La frontera en sí misma puede proporcionar los parámetros para la transformada de Fourier como se muestra en la figura 1.4.

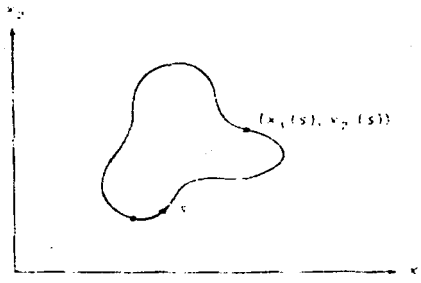


Fig. 1.4. Parametrización mediante expansiones de series de Fourier (figura tomada de [17]).

Donde las expansiones son:

$$\mathbf{x}(s) = \sum X_k e^{jk\omega_0 s} \quad \omega_0 = 2\pi/P, \quad P = \text{perímetro}$$

y los coeficientes discretos X_k de la serie de Fourier están dados por:

$$X_k = (1/P) \int_0^P \mathbf{x}(s) e^{-jk\omega_0 s} ds$$

Una característica común de los descriptores de Fourier es que típicamente la forma general se da por pocos términos de orden bajo, en la expansión de la frontera de la curva. Los coeficientes, propiamente parametrizados, son independientes del tamaño, translación y rotación de la forma que es descrita. Los descriptores no permiten la realización de una buena reconstrucción de la frontera: la curva resultante puede no ser cerrada si se usa únicamente un número finito de coeficientes para la reconstrucción.

1.3. Secciones cónicas

Las polinomiales son una elección natural para la representación de curvas, y ciertas polinomiales de grado 2 (llamadas círculos y elipses, mostradas en la figura 1.5) son curvas cerradas y por lo tanto definen regiones. Los círculos pueden ser representados

con tres parámetros, las elipses con cinco, y cónicas generales con seis. Las cónicas frecuentemente son buenos modelos para curvas físicas tales como los lados de objetos manufacturados.

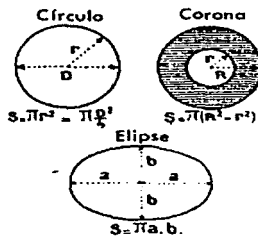


Fig. 1.5. Curvas cerradas: secciones cónicas.

Las secciones cónicas son muy útiles porque proporcionan curvas cerradas en dos dimensiones, las cuales ocurren en muchas imágenes, y son polinomiales familiares y bien comportadas de bajo grado.

Todas las formas cónicas estándar pueden estar sujetas a rotación, translación y cambio de escala para moverlas alrededor del plano. Estas operaciones afectan la ecuación de la cónica de una manera predecible.

1.4. B-esplines

Las B-esplines son curvas polinomiales que están relacionadas a un *polígono guía*. Las polinomiales cúbicas son las más frecuentemente usadas por esplines ya que son el orden más bajo en el cual la curvatura puede cambiar de signo. Las esplines son muy útiles en visión por computadora porque permiten modelos precisos manipulables de formas complejas. Los modelos pueden ser usados para guiar y monitorear tareas de segmentación y reconocimiento. Con B-esplines es posible la generación interactiva de modelos de formas complejas y el hecho de que las curvas esplines complejas tienen representaciones concisas (como sus polígonos guía) permite el uso de programas para

manipularlas fácilmente. En la figura 1.6 se muestra un ejemplo de la representación de una curva B-espline (ver [28]).

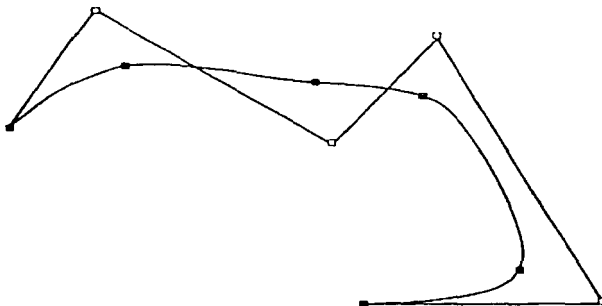


Fig. 1.6. Una curva espline y su polígono guía (figura tomada de [17]).

Hasta el momento se han presentado varios métodos de representación de curvas, aunque ellos sólo pueden representar curvas en dos dimensiones.

En visión por computadora, la cual es el área de interés en esta tesis, la representación de curvas en tres dimensiones (3D) es una parte muy importante. Por lo que en capítulos subsecuentes se tratará la representación de curvas digitales en 3D por medio del uso de codificación en cadenas. Este tipo de codificación es unidimensional.

CAPITULO 2.

REPRESENTACIÓN DE FORMAS EN 2D Y 3D.

Una de las cosas más importantes que se busca determinar en una imagen es la *forma* de los objetos en ella; mientras que el color, la textura, el movimiento y la intensidad son cuantificados relativamente fácil, la forma es mucho más compleja. El estudio de las formas es una parte importante en el campo de Reconocimiento de Patrones y Visión por computadora.

Para realizar un análisis global de las formas, varios autores han usado cadenas de Freeman, transformación de ejes, descomposición en subconjuntos primarios convexos, coordenadas polares, descomposición en vértices cóncavos; descomposición por clustering, ejes espejo y detectores stroke. Estos y otros métodos han sido revisados por Pavlidis (ver [2]). Otros autores (ver la referencia [3]), han analizado la teoría de codificación para el reconocimiento de patrones para diferentes representaciones, Ballard y Brown (ver [4]) describen las principales estructuras geométricas. Los descriptores de Fourier representan la frontera de una región como una función periódica, la cual puede ser expandida en una serie de Fourier (ver referencia [5]).

En este capítulo se presentan trabajos previos al presente y a la introducción del código de cadena que es utilizado, de esta manera se tendrá un estado del arte del tema que nos interesa.

2.1 Cadena de Freeman en cuatro direcciones

El primero en representar formas por medio de un código de cadena fue Freeman en 1961 (ver [6]). El código de cadena de Freeman constaba de sólo cuatro direcciones en forma ortogonal. Para una región dada dentro de una rejilla a cuadros de orientación y

tamaño fijos, la forma discreta resultante estará formada por los cuadros de la rejilla que son cubiertos en más de su mitad por la región. La cadena de Freeman en cuatro direcciones es la curva que se obtiene recorriendo en sentido de las manecillas del reloj dicha forma discreta, como se muestra en la figura 2.1.

2.2 Cadena de Freeman en ocho direcciones

Después de presentar su código de cadena en cuatro direcciones, Freeman, presenta en el mismo año (1961), su código de cadena en ocho direcciones. Las principales características de este código son:

1. Se usa una rejilla (al igual que en su código en cuatro direcciones).
2. Es independiente bajo translación.
3. El rango de direcciones va de 0 a 7 (cada 45°).
4. El tamaño de los segmentos de los elementos de número par es l y el de los elementos de número impar ($l\sqrt{2}$).

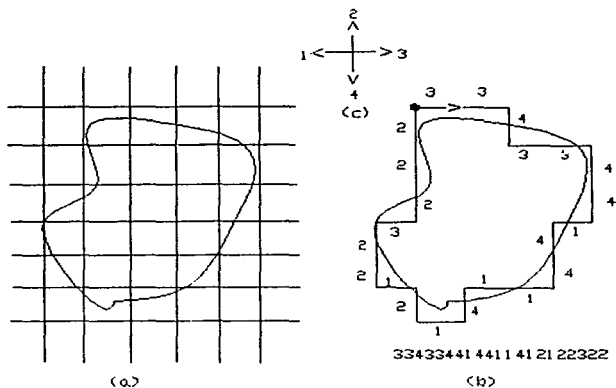


Fig. 2.1. Cadena de Freeman: (a) La región y la rejilla sobrepuesta; (b) La cadena de Freeman en 4 direcciones de la región; (c) Las cuatro direcciones.

Este método da un-descriptor más preciso de la forma, como se muestra en la figura 2.2. Sin embargo, esta notación no es independiente bajo rotación; Freeman presenta varias soluciones para resolver este problema de su código de cadena, utilizando para ello la distancia entre dos puntos, la intersección de dos cadenas, la simetría de cadenas, los momentos alrededor de un eje diagonal, los centroides y otros (ver [7]).

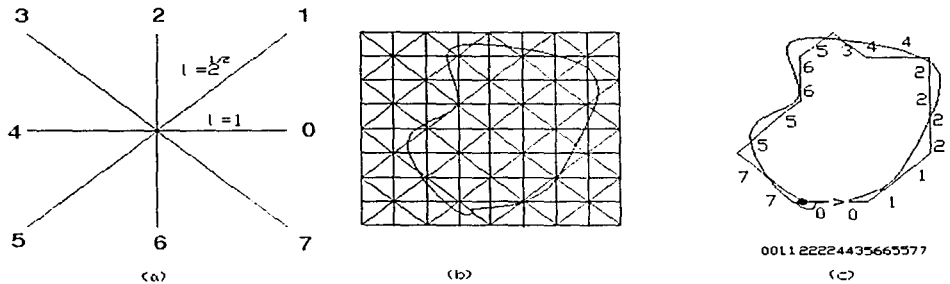


Fig. 2.2. (a) Las ocho direcciones ($l=1$ para elementos de número par, $l=2^{1/2}$ para elementos de número impar). (b) La rejilla sobre puesta sobre la región. (c) Cadena de Freeman en ocho direcciones.

2.3 Números de Forma

Un punto importante, en la presente sección, es la suposición de que la *entidad* ha sido aislada del plano. Esta es llamada *forma*, y es definida como resultado de un procesamiento previo: las formas están compuestas de puntos discretos.

En esta sección se estudiarán los *números de forma*.

2.3.1 Qué es una forma

Para analizar una forma primero se tienen que definir varios conceptos.

Una *región* es una porción de plano conectada simplemente y limitada por una curva frontera. Esto significa que la curva frontera no debe intersectarse a sí misma. Esto es

una frontera cerrada. Cualquier región dada tiene un tamaño, una posición y una orientación en el plano. Esto define una región plana, la cual es definida por la curva que tiene como frontera.

Una *forma* es una región pero despreciando su tamaño, posición y orientación en el plano. Esto es, dos regiones tienen la misma forma si se puede hacerlas coincidir realizando translaciones y rotaciones en el plano, además de un cambio de escala uniforme. En general, una región y su *imagen espejo* (ver apéndice) no tendrán la misma forma. Esta definición responde al concepto intuitivo de "forma".

Si se usa una notación para representar la forma de una región, debería ser independiente de la posición, orientación y tamaño de la región; además, ésta debería ser reproducible, de esta manera, si una región fuera trasladada, ampliada y rotada, la nueva región debería dar la misma descripción, como si no hubiera sido transformada. Entonces, dos regiones con diferentes formas, deberían generar diferentes descripciones. Finalmente, la descripción de la forma debería ser única para una región dada; por ejemplo, ésta no debería depender de un punto de inicio arbitrario o de un sistema coordenado particular.

Si la notación puede ser deducida exclusivamente de la región, por ejemplo sin comparar con una tabla de formas canónicas o descriptores de forma, en muchos casos es posible el ahorro de memoria y tiempo de cómputo si el procedimiento que genera la descripción de la forma es eficiente.

2.3.2 Formas discretas y continuas

En esta sección se dirá que la frontera de la región de una forma es *discreta* si está formada por segmentos de una rejilla cuadrada.

2.3.3 Convirtiendo una forma continua en discreta

Una rejilla cuadrada puede ser sobrepuesta en la parte más alta de una forma continua para obtener una forma discreta. La conversión de una forma continua en una discreta se hace como sigue: un cuadro de la rejilla pertenece a la forma discreta si más

del 50% del mismo queda cubierto por la forma continua. La forma discreta dependerá del tamaño, la orientación y la posición de la rejilla. Por lo tanto no puede ser un descriptor de la forma continua (como se puede apreciar en la figura 2.3).

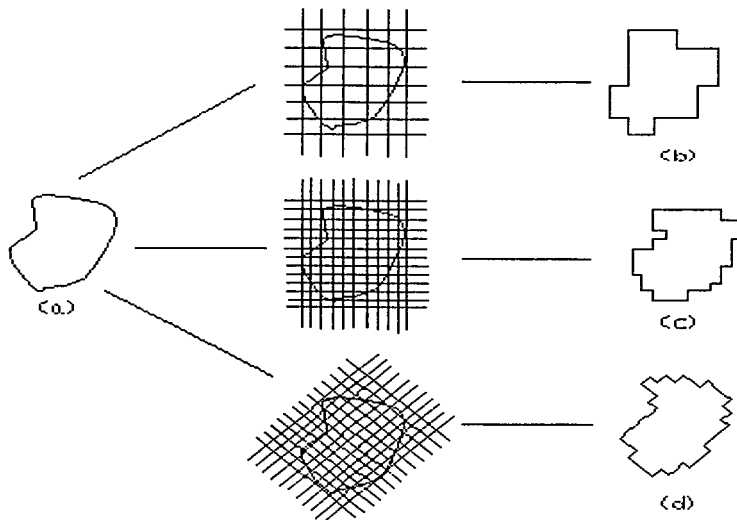


Fig. 2.3. Formas discretas y continuas. La forma continua en (a), genera varias formas discretas como en (b), (c) y (d). Si se necesitara tener una *única* forma discreta de la forma en (a), sería necesario especificar el tamaño de la rejilla (relacionado con el *orden* de la forma discreta), así como su orientación y posición, respecto a la forma continua en (a). De esta manera, para un orden dado n , la forma discreta correspondiente a la forma continua en (a) será única.

2.3.4 Excentricidad

La excentricidad (la relación del eje mayor al eje menor, ver figura 2.4) de una región es un descriptor que depende únicamente de su forma.

El *eje mayor* de una región es la línea que une dos puntos de su perímetro, tal que éstos sean dos puntos de la región lo mas alejados posible uno del otro sobre la región. El *eje menor* es perpendicular al eje mayor, y su tamaño es igual a la altura del rectángulo frontera (ver apéndice) que encierra la región. Este rectángulo es llamado el *rectángulo básico*.

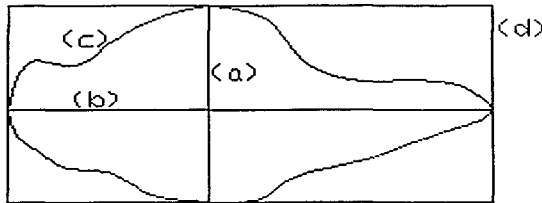


Fig. 2.4. El rectángulo básico: (a) eje menor de (c); (b) eje mayor de (c); (c) la región; (d) el rectángulo básico de (c). La excentricidad $e = (b)/(a)$ es siempre mayor o igual a 1.

2.3.5 Derivada de la cadena de Freeman en cuatro direcciones

La derivada de la cadena de Freeman (en cuatro direcciones), es el número que se obtiene al recorrer la frontera de la región en sentido de las manecillas del reloj, y reemplazando cada ángulo convexo de la cadena de Freeman por un 1, cada ángulo llano por un 2, y cada ángulo cóncavo por un 3, como se muestra en la figura 2.5. El número obtenido en (f), será diferente si se cambia la orientación de la rejilla. En la siguiente sección se muestra un método que hace la "derivada de la cadena de Freeman" independiente de estos cambios. Esta nueva derivada es llamada *número de forma*.

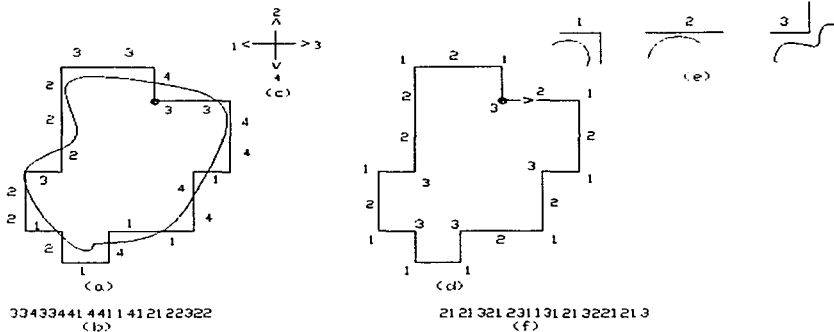


Fig. 2.5. Cadenas: (a) la cadena de Freeman en cuatro direcciones; (b) su número de cadena; (c) las cuatro direcciones usadas para codificar (a) en (b); (d) la nueva derivada de la cadena de Freeman; (e) los tres tipos de ángulos para codificar (d) en (f).

2.3.6 Números de Forma

Con los números de forma se trata de definir un procedimiento para medir la semejanza entre dos formas cualesquiera. El método para encontrar el número de forma de una región es el siguiente:

1. Una rejilla de celdas de tamaño del *rectángulo básico* (ver apéndice) es sobrepuesta en la región. Una región "negra" es formada con todas las celdas que están rellenas en al menos un 50 por ciento por la región.
2. La frontera de la región negra será representada por una cadena. Esta cadena es denotada por la notación de la derivada de la cadena de Freeman. La figura 2.6 muestra un ejemplo de cadena rodeando la región en sentido de las manecillas del reloj. Se puede observar que hay varias cadenas de dígitos 1, 2 y 3 correspondientes a la región codificada dependiendo del punto de inicio:

- A. 221231212132131131
- B. 212312121321311312
- C. 123121213213113122
- D. 231212132131131221
- E. 312121321311312212

- F. 121213213113122123
- G. 212132131131221231
- H. 121321311312212312
- I. 213213113122123121
- J. 132131131221231212
- K. 321311312212312121
- L. 213113122123121213
- M. 131131221231212132
- N. 311312212312121321
- O. 113122123121213213
- P. 131221231212132131
- Q. 312212312121321311
- R. 122123121213213113

Una de éstas cadenas es mínima, cuando se toma como un número en base 3: O en este ejemplo.

3. Seleccionar la cadena que es mínima como la cadena que representa a la región. En este ejemplo es: 113122123121213213. Se puede observar que la cadena mínima siempre empieza con un 1, ya que cada forma discreta contiene al menos cuatro 1's.

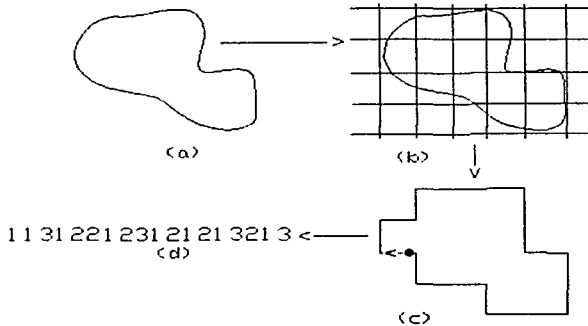


Fig. 2.6. (a) La forma continua. (b) Convirtiendo la forma continua en forma discreta. (c) La forma discreta. (d) El número de forma.

La postura adoptada en los métodos de codificación de formas, es que la orientación de la rejilla puede ser normalizada, pero el tamaño es un parámetro que permite variar la precisión del número de forma. La orientación de la rejilla no es arbitraria, ya que se hace coincidir con el eje mayor de la región, de esta manera si la región es rotada, la rejilla rotará de la misma manera, así el número de forma obtenido es invariante bajo rotación.

2.3.7 El orden de un número de forma

El orden de un número de forma es el número de dígitos que la forma contiene, este número es siempre par, porque la frontera de la forma es cerrada. En el ejemplo anterior, el orden de la forma es 18.

Como se ha visto hasta ahora, el número de forma depende de la región. Los números de forma pueden ser de diferente orden; a más alto orden se tendrá una mayor precisión de la descripción de la forma.

Los números de forma fueron definidos para medir similitud entre dos regiones, para lo cual se definió un *grado de semejanza*; así como la *distancia* entre dos formas (para mayor información consultar [1]).

2.4 Una estructura geométrica para formas bidimensionales

En esta sección se describirán trayectorias o formas mediante elementos discretos en un espacio continuo. Son usadas secuencias numéricas llamadas *series*, las cuales pueden generar formas de un gran número de características.

Tómese el ejemplo del ADN el cual puede ser descrito usando la notación de cambio de inclinación (NCI). Una serie es como una *semilla* que contiene toda la información de la forma, entonces una serie puede generar el universo de todas las hélices discretizadas de ADN con una exactitud predefinida.

El objetivo de la NCI es simplificar algunos procesos matemáticos y geométricos usados en el análisis de formas y superficies (ver [8]).

La NCI es similar a otros códigos de cadena, ya que ésta también usa secuencias numéricas, pero también tiene otras importantes diferencias con ellos.

Las principales ventajas de esta notación sobre las cadenas de Freeman son:

1. NCI no usa una rejilla
2. NCI es independiente bajo translación y rotación, y opcionalmente, de tamaño.
3. El rango de cambios de inclinación es ilimitado (va continuamente de -1 a 1); esto facilita la generación de secuencia numéricas. Los vértices siempre tocan la curva, por lo tanto el descriptor de la forma obtenido es mejor.
4. El tamaño de los segmentos de línea recta (l) es siempre el mismo en toda la forma.

Para tener una idea más clara de cómo se usa la NCI, se describirán algunos de sus conceptos.

Cuando se usa la NCI, los cambios de dirección varían continuamente de -1 a 1 . La figura 2.7 muestra el rango de cambios de dirección y la figura 2.8 muestra algunos ejemplos de cambios de inclinación. Cualquier cambio de inclinación en el intervalo $[-1, 1]$ es válido.

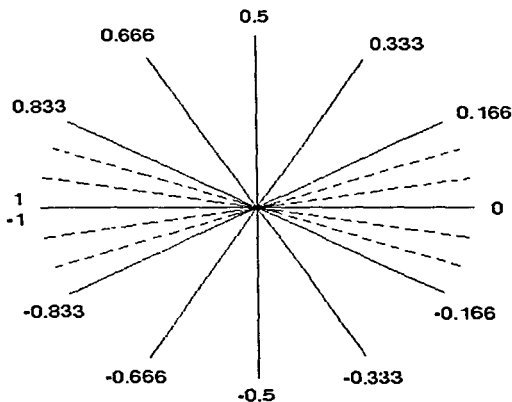


Fig. 2.7. Rangos de cambios de dirección $[0,1]$ y $[0,-1]$.

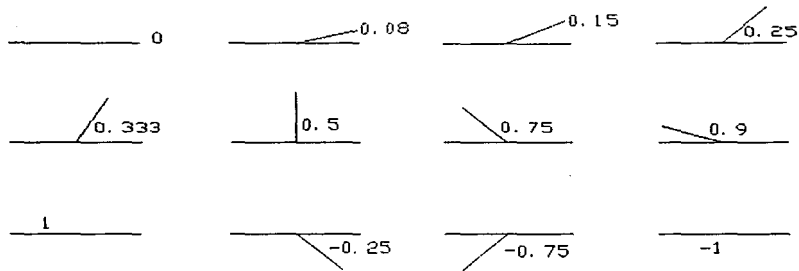


Fig. 2.8. Algunos ejemplos de cambios de inclinación.

2.4.1 Conversión de una curva continua en una forma discreta

Para convertir una curva continua en una forma discreta, primero se debe escoger la longitud de los segmentos de línea recta que serán usados para describirla. Después se debe seleccionar un punto de la curva como el origen y el extremo de uno de los segmentos de línea recta se debe hacer coincidir con ese origen. El punto opuesto de ese segmento es también sobrepuesto a la curva, de manera que su punto extremo se posiciona sobre la curva, y en este mismo punto es donde será colocado un extremo del siguiente segmento, cuyo otro extremo a su vez será sobrepuesto a la curva, y así sucesivamente.

De esta manera, se podrá determinar el cambio de ángulo entre segmentos de línea recta contiguos. Este cambio de ángulo deberá ser ajustado para estar dentro del rango $[0,1]$; cuando el cambio de ángulo sea un valor negativo, entonces será ajustado para estar dentro del rango $[-1,0]$. Este cambio de ángulo es el que es llamado *cambio de inclinación*. La secuencia de cambios de inclinación es la serie que define la forma discreta de la curva continua (ver figura 2.9).

Si se observa la serie mostrada en la figura 2.9(e) se puede notar que tiene una exactitud predeterminada (10^{-3}) en los cambios de inclinación, por supuesto, esto puede ser cambiado.

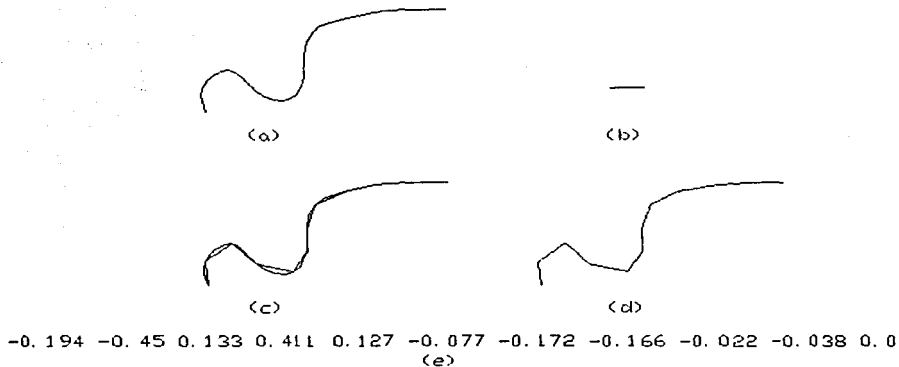


Fig. 2.9. Conversión de una curva continua a una forma discreta: (a) una curva continua; (b) definición del tamaño del segmento de línea recta; (c) Recorriendo la curva usando segmentos de línea recta para determinar los cambios de inclinación (nótese que los vértices de los segmentos están sobre la curva); (d) la forma discreta; (e) la serie de la forma discreta es formada por una secuencia de elementos, los cuales indican los cambios de inclinación.

2.4.2 Independencia de translación

En la NCI cualquier forma es independiente bajo translación. Esto es debido a que no hay una posición como referencia en el plano coordenado.

2.4.3 Independencia de rotación

Usando la NCI cualquier forma es independiente bajo rotación. Esto es debido al uso de cambios de inclinación alrededor de la forma. La figura 2.10 ilustra la independencia bajo rotación.

2.4.4 Cambio de Escala

El escalamiento de cualquier forma está dado por el tamaño del segmento de línea recta l .

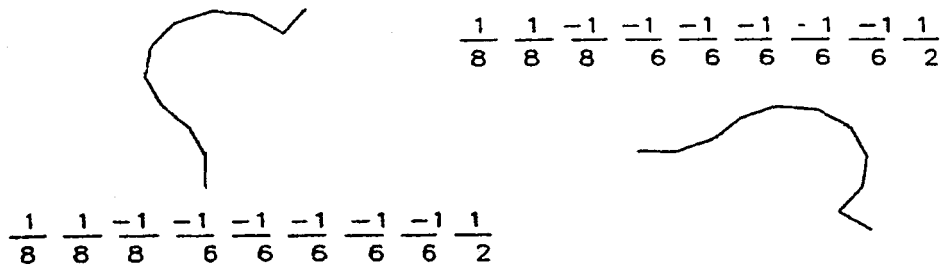


Fig. 2.10. Independencia de rotación (figura tomada de [8]).

2.4.5 Independencia de tamaño

En reconocimiento de formas, la independencia de tamaño es muy importante, por ejemplo, en la medida de similitud entre formas. La independencia de tamaño de una forma es obtenida normalizando la longitud de su perímetro a un número fijo de elementos alrededor de él. La figura 2.11 muestra la independencia de tamaño de dos formas iguales con tamaños diferentes.

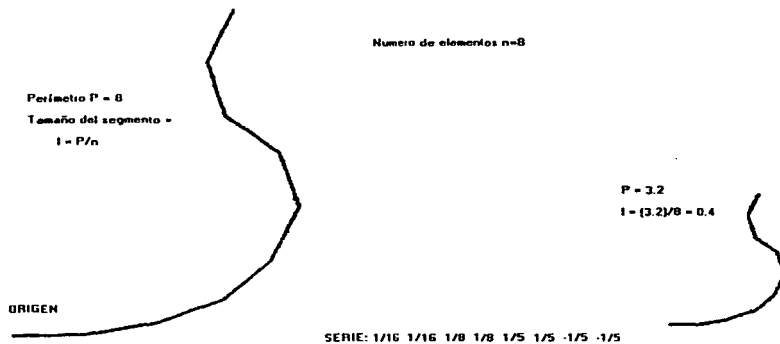


Fig. 2.11. Independencia de tamaño.

2.5 Código de cadena basado en vértices (CCV)

En 1997 Bribiesca propuso un nuevo código de cadena (ver [10]). Éste código fue definido para formas compuestas de *celdas regulares* (ver apéndice), donde a cada vértice de la frontera se le asocia el número de celdas que lo comparten. A este nuevo código de cadena se le ha llamado *código de cadena basado en vértices* (CCV). El CCV es invariante bajo translación y rotación, su punto de inicio puede ser normalizado; también puede ser invariante bajo transformación espejo. Usando este concepto de código de cadena es posible relacionar la longitud de la cadena con el *perímetro de contacto*, el cual corresponde a la suma de las longitudes de los segmentos que son comunes a dos celdas (ver [11]) y también para relacionar los nodos de la cadena con los vértices de contacto, los cuales corresponden a los vértices de las celdas vecinas. De esta manera las relaciones entre la cadena y el interior de la forma permiten obtener propiedades interesantes. El trabajo que se presenta en esta sección fue impulsado por la idea de obtener varias características de la forma calculadas directamente del CCV, sin la necesidad de transportarse a la representación en coordenadas cartesianas. En la figura 2.12 se muestra una forma discreta compuesta de diferentes tipos de celda: (a) triangular, (b) rectangular y (c) hexagonal. La longitud de cada lado de cualquier celda es igual a uno.

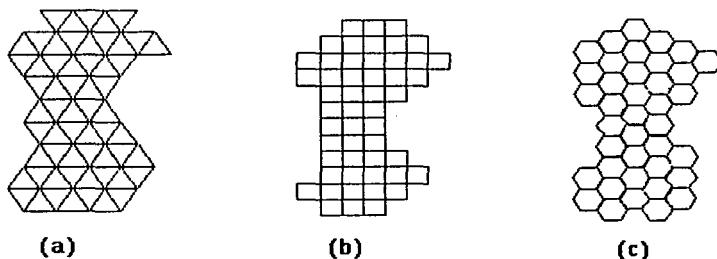


Fig. 2.12. Ejemplos de formas discretas de diferentes tipos de celdas: (a) forma compuesta de celdas triangulares; (b) forma compuesta de celdas rectangulares; (c) forma compuesta de celdas hexagonales.

Las fronteras o contornos de cualquier forma discreta compuesta de celdas regulares puede ser representada por cadenas, por lo tanto, éstas cadenas representan fronteras cerradas, las cuales son consideradas en esta sección.

2.5.1 Normalización con respecto al punto de inicio

Usando la CCV para formas compuestas de celdas regulares todas las cadenas son cerradas. De esta manera el CCV puede ser normalizado bajo su punto de inicio, escogiendo tal punto de manera que la secuencia resultante de elementos forme un entero de magnitud mínima. Por lo tanto, la cadena mostrada en la figura 2.13(c) puede normalizarse rotando los dígitos hasta que el número sea mínimo. En la figura 2.13(d) se presenta la cadena así obtenida.

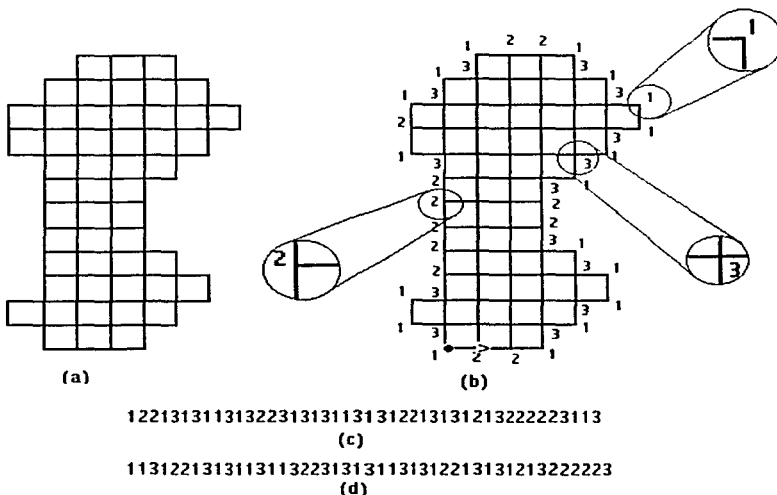


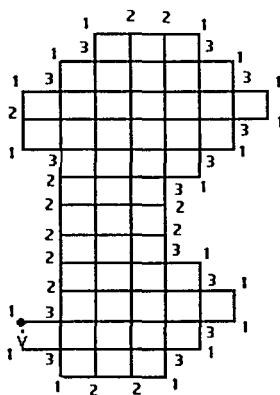
Fig. 2.13. El CCV: (a) una forma compuesta de píxeles; (b) los elementos del CCV; (c) la cadena de la forma mostrada en (a), en dirección contraria a las manecillas del reloj; (d) la cadena en (c) normalizada.

Como se puede observar, cuando se usan pixeles (cuadrados), hay tres diferentes números de vértices formados por las celdas en el contorno de la forma: 1, 2 y 3.

2.5.2 Independencia de rotación

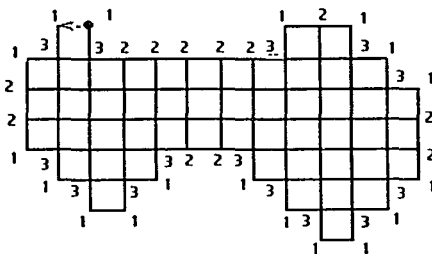
El CCV es invariante bajo rotación; esto es debido a que está basado en el número de celdas por vértice que están en contacto con la frontera de la forma. La figura 2.14(a) muestra una forma discreta recorrida en sentido contrario a las manecillas del reloj y su correspondiente cadena, dicha cadena está normalizada bajo punto de inicio.

La figura 2.14(b) ilustra la independencia de rotación de la forma presentada en la figura 2.14(a).



11312213131131132231313113131221313121322223

(a)



11312213131131132231313113131221313121322223

(b)

Fig. 2.14. Independencia de rotación: (a) una forma y su correspondiente cadena; (b) la forma presentada en (a) y su correspondiente cadena, mostrando independencia de rotación.

2.5.3 Independencia de transformación espejo

El CCV es invariante bajo transformación espejo. Cualquier forma puede ser construida directamente de su código de cadena. La figura 2.15(a) muestra una forma recorrida en el sentido de las manecillas del reloj y si se obtiene su imagen especular, el perímetro se recorre al contrario construyendo la misma cadena tal como se muestra en la figura 2.15(b).

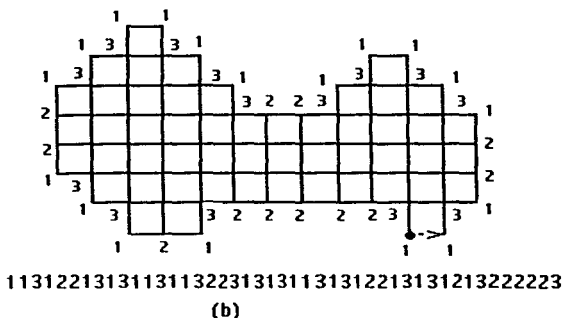
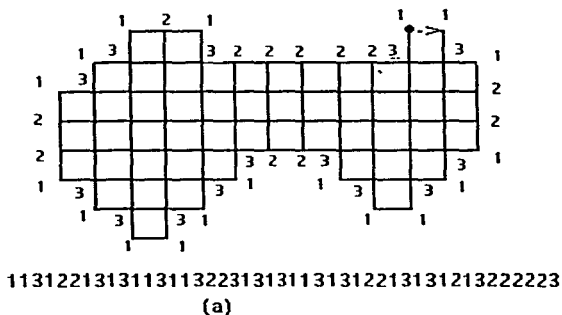
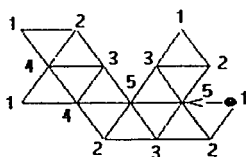


Fig. 2.15. Independencia de transformación espejo: (a) una forma discreta y su cadena; (b) la forma y su cadena mostrando independencia de transformación espejo.

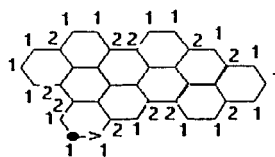
2.5.4 El CCV para formas compuestas de celdas triangulares y celdas hexagonales

Las propiedades presentadas anteriormente para formas compuestas de celdas rectangulares o pixeles, son también válidas para celdas triangulares y hexagonales. La figura 2.16(a) muestra una forma compuesta de 13 celdas triangulares y su correspondiente cadena. Un vértice de la frontera puede ser común a 1, 2, 3, 4 ó 5 celdas. La figura 2.16(b) muestra una forma compuesta de 11 celdas hexagonales y su correspondiente cadena. Cuando se usa este tipo de celda, se tienen únicamente dos tipos de vértices en la frontera, de acuerdo a su pertenencia a 1 ó 2 celdas.



152135321414232

(a)



11212211211121211221122112111221

(b)

Fig. 2.16. Formas compuestas de celdas triangulares y celdas hexagonales: (a) una forma compuesta de 13 celdas triangulares y su cadena; (b) una forma compuesta de 11 celdas hexagonales y su cadena.

En [10] se hace una comparación entre el perímetro de contacto y los vértices de contacto, ahí mismo definidos. Además se muestran poliedros como formas en el plano y su codificación en cadenas CCV.

2.6 Esquemas de representación para curvas en 3D

Los esquemas típicos de representación digital consisten de un procedimiento de cuantización (o digitalización), seguido de una codificación de cadena. Los códigos de

cadena son el formato de entrada estándar para numerosos análisis de forma y para algoritmos de visión por computadora. En ésta tesis se han considerado varios esquemas de representación de curvas en 2D.

Los detalles del esquema de representación digital de curvas pueden tener una gran influencia en el desempeño de subsecuentes algoritmos. Por ejemplo, Kulpa (ver [12]) resolvió una controversia entre dos aproximaciones para la medida del área, mostrando que ellas se relacionaban a diferentes esquemas de representación. Se puede ver que los métodos para la estimación del perímetro original de una forma en 2D mediante códigos de cadena han evolucionado, desde el trabajo original de Freeman hasta recientes contribuciones. Apparently el cuidadoso análisis de los esquemas de digitalización y sus efectos en las propiedades de segmentos de línea recta digitalizadas, han sido la clave para avanzar en el desempeño de la estimación de perímetro.

La creciente importancia del análisis de formas tridimensionales en imágenes médicas, procesamiento de imágenes de rango, análisis de secuencias de imágenes, navegación por medio de robots y otros dominios, necesitan la definición, evaluación y una selección bien fundamentada de los esquemas de representación de curvas en tres dimensiones.

El código de cadena en 3D fue definido por Freeman en 1974 (ver [13]), pero sin una especificación explícita del método de cuantización. En la última década se han sugerido varios métodos de representación de curvas en 3D.

La experiencia en el análisis de curvas, trayectorias y superficies en 3D ha proporcionado un mejor entendimiento de los esquemas de representación digital de curvas en 3D, que es crucial para el desarrollo de un desempeño más alto en análisis de algoritmos de formas en 3D. La extensión de esquemas en 2D a esquemas en 3D es no trivial. Propiedades importantes de esquemas en 3D no son obvias, son poco intuitivas y han escapado de la atención de varios investigadores.

En esta sección se muestra cómo el concepto de esquema de representación de curvas en 3D "bien definida" matemáticamente, ha sido formalizado y resumido en una lista de requerimientos. Esto ha sido usado como base para comparación cuantitativa y cualitativa de varios métodos que han sido sugeridos (para mayor información acerca de esta sección consultar [14]).

2.6.1 Métodos tridimensionales y n-dimensionales [14]

La mayoría de los esquemas de representación digital en 2D no permiten ser generalizados a más dimensiones. Los métodos Cuantización de la Frontera del Objeto (OBQ por sus siglas en inglés, ver figura 2.17(a)) y Cuantización de la Frontera del Fondo (BBQ por sus siglas en inglés, ver figura 2.17(b)) no pueden ser usados en tres o mayores dimensiones, puesto que el lado de un objeto y el lado de la frontera de una curva no existen. En Cuantización Convexa en 3D (ver figuras 18 y 19), la conectividad no se mantiene siempre, ya que la curva puede pasar entre dominios sin cruzarlos. Por ejemplo, si se usa dominios esféricos como una generalización de Quantization Convexa usando círculos a tres dimensiones, una línea infinita o finita puede pasar entre las esferas sin intersectar alguna de ellas, como se muestra en el figura 2.19.

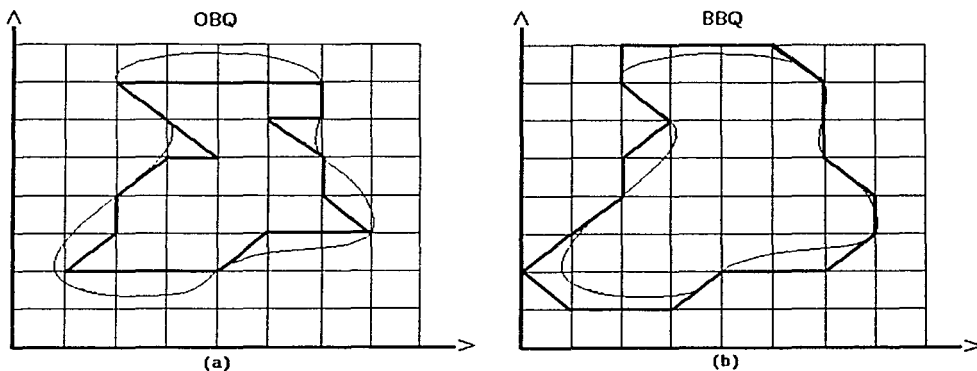


Fig. 2.17. Los métodos OBQ y BBQ para representar curvas en 2D. En cada intersección de la curva con un borde de cambio de píxel, uno de los dos puntos finales del borde es seleccionado e incluido en la representación. En OBQ la representación queda dentro de la forma, y en BBQ fuera de ella.

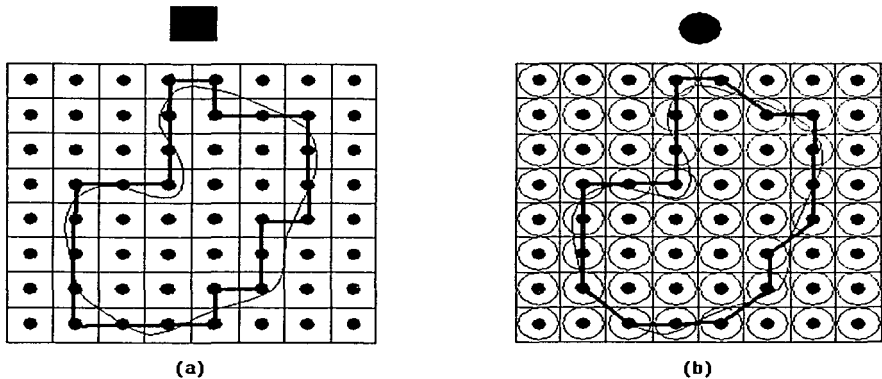


Fig. 2.18. Cuatización Convexa en dos dimensiones. Un dominio convexo es posicionado alrededor de cada punto-celda, y el punto es incluido en la representación si la curva intersecta el dominio. (a) Cuantización usando cuadros. (b) Cuantización convexa usando círculos.

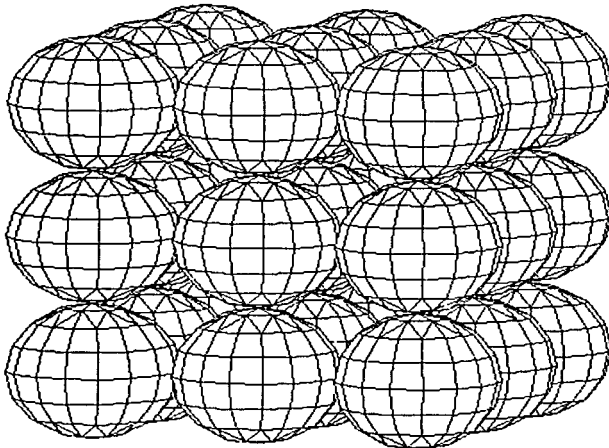


Fig. 2.19. Un sencillo intento por generalizar Cuantización convexa usando círculos en 2D a Cuantización convexa usando esferas en 3D. Una línea infinita podría pasar entre las esferas sin intersectar alguna de ellas, por lo que aquél método no puede ser generalizado a 3D.

En contraste Cuantización usando Cuadros (ver figura 2.20) si se puede generalizar a Cuantización usando Cubos (CQ por sus siglas en inglés), usando "voxels centrados" (ver apéndice) en lugar de "píxeles centrados" (ver figura 2.21).

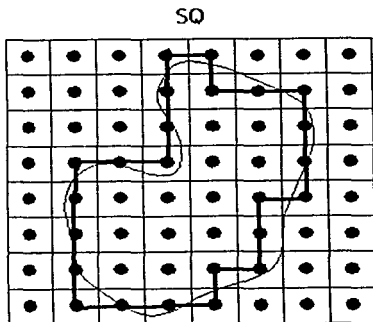


Fig. 2.20. El método Cuantización usando cuadros para representar curvas en 2D.

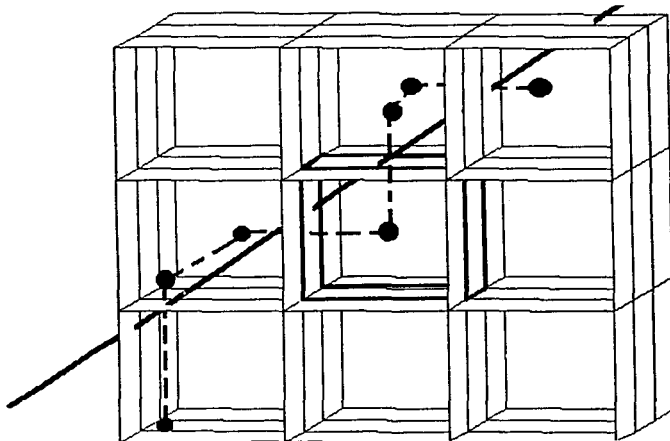


Fig. 2.21. El método Cuantización usando cubos para representar curvas en 3D es una generalización del método Cuantización usando cuadros en 2D.

El método Cuantización de la Intersección con la Rejilla (GIQ por sus siglas en inglés, ver figura 2.22) puede ser generalizado a 3D. En 3D GIQ, para cada intersección de la curva con la cara del voxel se toma el vértice más cercano de los cuatro puntos-celda como se muestra en la figura 2.23.

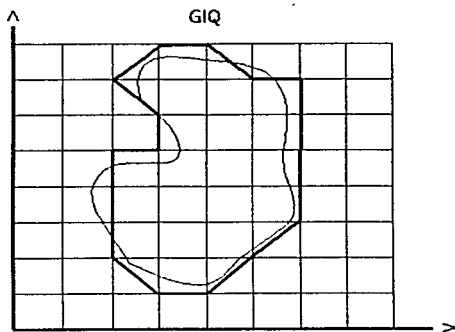


Fig. 2.22. El método GIQ para representación de curvas en 2D. En cada intersección de la curva con el borde de un píxel, el vértice más cercano es seleccionado e incluido en la representación.

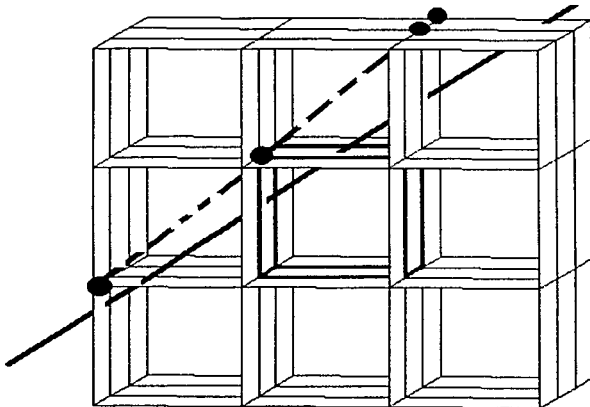


Fig. 2.23. GIQ tridimensional. En cada intersección de la curva con la cara de un voxel, el vértice más cercano es seleccionado e incluido en la representación.

El método Quantización de Longitud Máxima (MLQ por sus siglas en inglés) es aplicable en cualquier dimensión. La curva se divide en segmentos por medio de sus intersecciones con hiperplanos (líneas en 2D, planos en 3D), perpendiculares a uno de los ejes coordenados y son equidistantes a los hiperplanos vecinos de coordenadas enteras sobre dicho eje.

Para cada segmento, se selecciona el punto-celda con el más largo subsegmento en su *spel* (según Lohmann [16]) centrado (pixel en 2D, voxel en 3D). Este proceso es repetido para cada uno de los ejes. Los puntos-celda seleccionados son ordenados de acuerdo al orden de los subsegmentos correspondientes en la curva.

Una ilustración de 2D MLQ se muestra en la figura 2.24.

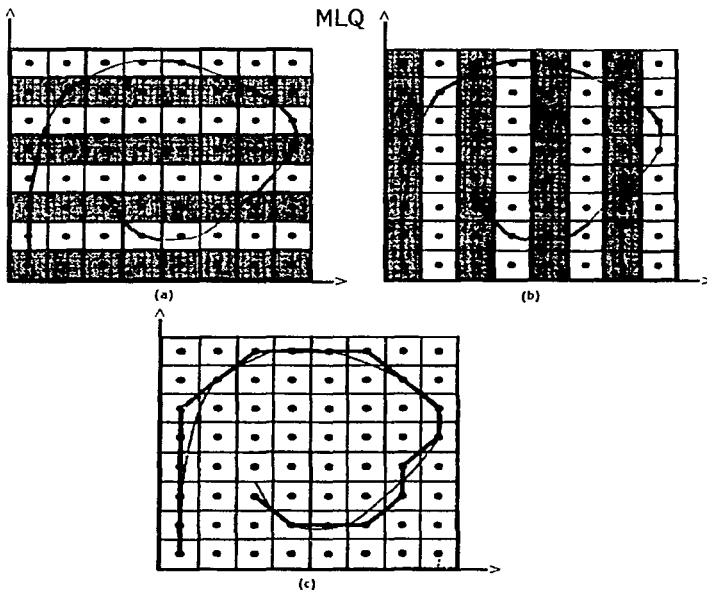


Fig. 2.24. MLQ en 2D. En (a) la curva es segmentada de acuerdo a la intersección con las líneas que son perpendiculares al eje Y y son la mitad del camino entre las líneas coordenadas enteras. Para cada segmento el punto-celda con el más largo subsegmento y su pixel centrado es seleccionado. En (b), el proceso es repetido con respecto del eje X. La cadena resultante es la mostrada en (c).

La figura 2.24(a) demuestra la división de la curva por segmentos y sus intersecciones con líneas (horizontales) de la forma. Cada segmento de la curva pasa a través de uno o más píxeles dando lugar a respectivos subsegmentos. Se elige el punto-celda correspondiente al subsegmento mas largo. En la figura 2.24(b), el proceso es repetido con respecto al eje X. La figura 2.24(c) muestra la unión de los puntos-celda seleccionados en las dos etapas previas, para obtener la representación 2D MLQ final. Nótese que en 2D, para una línea recta, MLQ es equivalente a GIQ.

A diferencia de otros métodos, en MLQ, la representación digital no depende sólo de ciertos puntos especiales (como en la intersección de puntos en una malla), pero integra la contribución de todos los puntos de la curva. El método MLQ es extendible a 3D.

Otro método usado es Cuantización usando cubos de adelgazamiento (TCQ por sus siglas en inglés). La cadena obtenida mediante éste método es un subconjunto de los puntos de la cadena obtenidos mediante Cuantización usando cubos (ver figura 2.25).

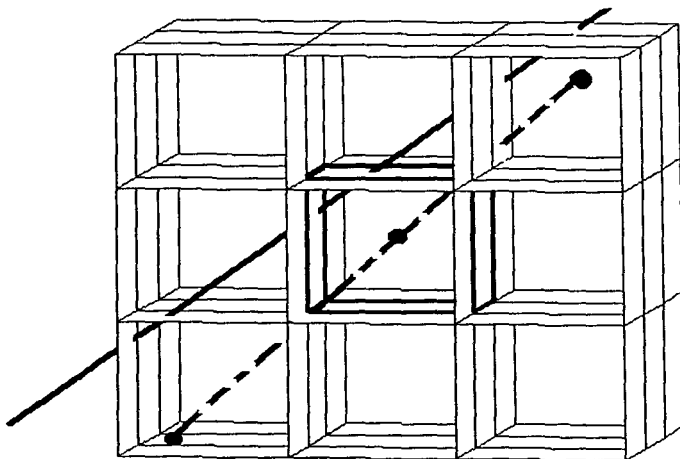


Fig. 2.25. El método TCQ para representación de curvas en 3D.

A continuación se formaliza el concepto de esquema de representación de curvas en 3D matemáticamente bien definido y se resume en una lista de requerimientos. Estos se usan como una guía para comparación cualitativa y cuantitativa de varios métodos que han sido sugeridos. Los requerimientos que preferiblemente deben cumplir los métodos propuestos son los siguientes:

- (a) Simetría de los ejes coordenados: la cuantización debería ser simétrica con respecto a los cambios en las etiquetas y direcciones de los ejes.
- (b) Simetría en la dirección de la curva: invirtiendo el punto de inicio y el punto final de la curva continua, se debería obtener la misma cadena en orden inverso. Éste requerimiento hace que la representación digital de una curva sin dirección sea invariante a un atributo de dirección arbitrariamente asignado.
- (c) Invarianza bajo translación entera: translación entera de una curva continua (o equivalente al origen de un sistema coordenado) debería dar la misma cadena trasladada.
- (d) Memoria finita: la representación de un segmento de curva no debería depender de segmentos distantes arbitrariamente. Además para asegurar una complejidad de memoria constante y una complejidad de tiempo lineal del algoritmo de digitalización de la curva, este requerimiento implica que una perturbación en la curva continua debido a *ruido* (ver apéndice) u otras razones, tendrá solo un efecto local en la representación digital. Esto está conceptualmente relacionado a la definición del problema bien planteado matemáticamente.
- (e) De línea a arco: la representación discreta de una línea recta debería ser un arco digital. Éste requerimiento es necesario para la minimización del número de elementos en la cadena discreta. Puesto que la cadena mas corta entre dos puntos-celda es necesariamente un arco digital, es razonable requerir que el segmento de línea recta digital sea un arco digital.
- (f) La propiedad proyección: la proyección de una cadena en 3D sobre un plano perpendicular a cualquiera de los ejes del sistema coordenado debería ser idéntica a la representación digital en 2D de la proyección de la curva continua sobre el plano. Cuando se generaliza un esquema de representación digital en 2D a 3D,

esta propiedad mejora el entendimiento y simplifica el análisis del esquema en 3D, relacionando sus operaciones a aquellas del esquema bien entendido en 2D.

- (g) Minimización de distancia: el proceso debería de minimizar, en algún sentido, la distancia entre la cadena discreta y la curva continua original. Obviamente, hay muchas maneras para definir la distancia.
- (h) Compacidad: la cadena debería consistir de tan pocos elementos como sea posible, sujeto a los demás requerimientos.

Esta guía puede ser usada en la selección o diseño de procedimientos de cuantización matemática o computacional. En procesos de cuantización física, tales como percepción de rango o tomografías, los requerimientos físicos para una aplicación específica (que frecuentemente son difíciles de modificar) manejarán sus características matemáticas.

En [14] se hace una comparación de los esquemas de representación de curvas digitales en 3D de acuerdo a la lista de requerimientos proporcionada.

CAPITULO 3.

CODIGO DE CADENA PARA REPRESENTAR CURVAS EN 3D

El código de cadena que se usa en esta tesis para representar curvas tridimensionales fue definido por Ernesto Bribiesca (ver [24]), quien establece que cualquier curva continua en 3D puede ser digitalizada y representada como una curva discreta en 3D. Esta curva discreta en 3D está compuesta de segmentos de línea recta. Los elementos de la cadena representan los cambios de dirección ortogonales de cada uno de los elementos de línea recta de la curva discreta. Este código de cadena fue usado para representar y procesar curvas discretas en 3D que representan un terreno obtenido de un *modelo de elevación digital* (ver apéndice). También fue usado para representar sólidos compuestos por voxels, donde cada sólido representa un modelo de elevación digital el cual es descrito por una cadena única.

El estudio de la representación de curvas en 3D es una parte importante en visión por computadora. Aquí se trata con la representación de formas en 3D basado en la codificación de cadena.

Las técnicas de código de cadena son ampliamente usadas porque preservan información importante y permiten la reducción de los datos considerablemente (ver [2,6,7,9,10,20]). Los códigos de cadena son el formato de entrada estándar para algoritmos de análisis de formas. Muchos autores han usado técnicas de codificación de cadena, esto es debido al hecho de que varias características de las formas pueden ser computadas directamente desde tal representación. La representación de curvas discretas en 3D por medio de un código de cadena es un desafío importante en visión por computadora.

En este trabajo, las curvas discretas están compuestas de segmentos de línea recta de tamaño constante, dos segmentos de línea recta contiguos definen un cambio de dirección, y dos cambios de dirección definen un elemento de la cadena. Este código de

cadena considera únicamente cambios de dirección relativos, constituyendo así un descriptor de la curva invariante bajo rotación y translación. Además, esto puede ser normalizado bajo el punto de inicio.

3.1 Conceptos y definiciones.

Una importante simplificación es la suposición de que las curvas discretas han sido aisladas del mundo real y son definidas mediante un procesamiento previo que consiste en convertir una curva continua que representa cualquier objeto real en una curva discreta. La figura 3.1(a) muestra un ejemplo de una curva continua y la figura 3.1(b) ilustra la correspondiente representación discreta de dicha curva. Se puede notar que la curva discreta está compuesta de segmentos de línea recta del mismo tamaño. Se considera que el tamaño del segmento de línea recta es igual a uno.

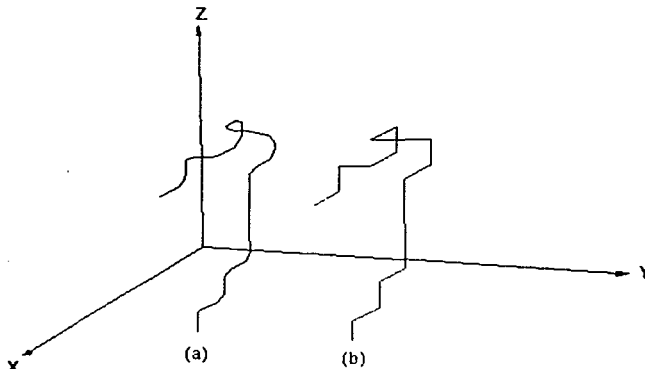


Fig. 3.1. Un ejemplo de una curva en 3D: (a) una curva continua en 3D; (b) la representación discreta de la curva mostrada en (a).

El contorno de cualquier forma discreta en 3D compuesta de segmentos de línea recta de tamaño constante puede ser representado por cadenas.

Se tienen únicamente 5 posibles cambios de dirección para representar cualquier curva discreta en 3D. Estos se asignan de manera similar al caso de los elementos de los números de forma tratados en la sección 2.3 del capítulo anterior.

Las figuras 3.2(a)-3.2(e) ilustran los 5 posibles elementos de cadena usados para representar curvas en 3D. Cada uno de estos elementos es un número que se asocia a un vértice y representa un cambio de dirección con relación al ángulo recto inmediato anterior a dicho vértice.

En (a) el elemento '0' representa que no hay cambio de dirección, por lo que se puede ver como "ir recto" a lo largo de los segmentos contiguos; (b) muestra el elemento '1', el cual indica un cambio de dirección "hacia la derecha"; (c) ilustra el elemento '2' el cual representa un cambio de dirección "hacia adelante"; en (d) el elemento '3' indica un cambio de dirección "hacia la izquierda"; finalmente el elemento '4' mostrado en (e) significa el cambio de dirección "hacia atrás".

Así, dos segmentos de línea recta contiguos definen un cambio de dirección y dos cambios de dirección (de 90° el primero) definen un elemento de la cadena.

Definición 1. Un elemento a_i de cadena es un número del 0 al 4.

Definición 2. Una cadena A es una secuencia ordenada de elementos, y es representada por

$$A = a_1 a_2 a_3 \dots a_n$$

donde n indica el número de elementos de la cadena.

3.2 Cómo obtener la cadena de una curva dada

La cadena de una curva se obtiene calculando los cambios de dirección relativos alrededor de la curva. Entonces la cadena obtenida estará compuesta de un número finito de elementos que se puede leer como un número en base 5. La figura 3.2(f) muestra un ejemplo de una curva discreta, el origen de la curva es considerado en el lado más bajo y es representado por un punto. En la misma figura 3.2(f) se ilustra el primer elemento de la

cadena, el cual corresponde al elemento "3", se debe notar que el primer cambio de dirección compuesto de dos segmentos que forman un ángulo recto es necesario para definir el primer elemento distinto de "0" y es usado únicamente como referencia. La figura 3.2(g) muestra el siguiente elemento de la cadena, este segundo elemento es "3" indicando un cambio de dirección hacia la izquierda con relación al ángulo recto inmediato anterior, en este caso formado en el elemento "3" anterior. Las figuras 2(h) – 2(n) ilustran los siguientes elementos de la cadena obtenidos paso a paso. La figura 3.2(n) muestra la curva discreta y su correspondiente cadena, la cual es compuesta de 9 elementos.

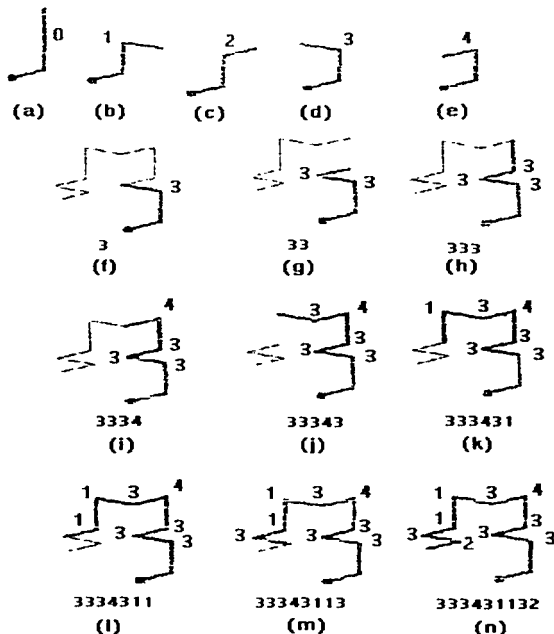


Fig. 3.2. Los cinco posibles cambios de dirección para representar curvas discretas en 3D: (a) el elemento "0"; (b) el elemento "1"; (c) el elemento "2"; (d) el elemento "3"; (e) el elemento "4"; (f) ejemplo de una curva discreta y el primer elemento de su cadena; (g) – (n) los siguientes elementos de la cadena.

Es claro que la longitud de una curva discreta (de las que estamos tratando) es igual a la suma de las longitudes de los segmentos que la forman (cada una de longitud l , que estamos suponiendo es igual a 1). Entonces, a partir de la cadena que representa a una curva discreta, tendremos que la longitud de obtendrá de la siguiente manera:

Definición 3. Sea $A = a_1 a_2 a_3 \dots a_n$ una cadena. Su longitud es:

$$L(A) = n+2 \quad n = \text{número de elementos de la cadena}$$

Para la cadena mostrada en la figura 3.2(n), $L(333431132) = 9+2 = 11$.

3.3 Independencia de Rotación

Este código de cadena es invariante bajo rotación, esto es porque se usan cambios de dirección relativos.

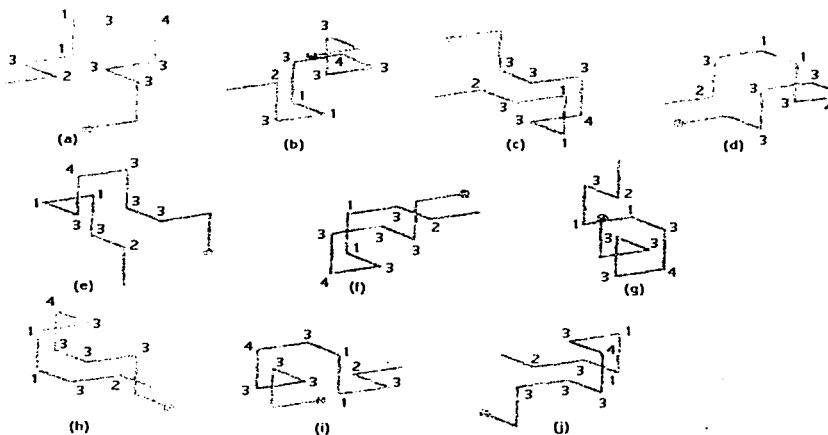


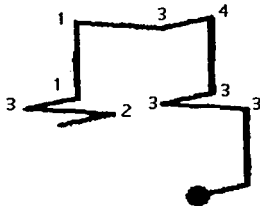
Fig. 3.3. Independencia de rotación: (a) una curva discreta y su cadena; (b) – (d) rotaciones de la curva mostrada en (a) sobre el eje de las “X”; (e) – (g) rotaciones sobre el eje de las “Y”; (h) – (j) rotaciones sobre el eje de las “Z”.

La figura 3.3(a) muestra la curva discreta representada en la figura 3.2(n) y su cadena correspondiente. Las figuras 3(b)-(d) muestran algunas rotaciones de la curva discreta mostrada en la figura 3.3(a) como una transformación rígida de R^3 alrededor del eje de rotación X. Las figuras 3(e)-(g) muestran las rotaciones de la curva presentada sobre el eje de rotación Y. Finalmente las figuras 3(h)-(j) ilustran las rotaciones de la curva sobre el eje de rotación Z. Note que todas las cadenas son iguales, por lo tanto, ellas son invariantes bajo rotación.

3.4 El inverso de una cadena

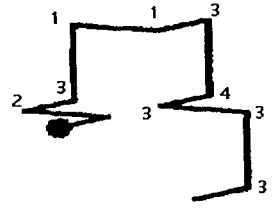
El inverso de una cadena es la cadena que se obtiene recorriendo la curva discreta en la dirección opuesta. La figura 3.4(b) muestra el inverso de la cadena presentada en (a), nótese que en este caso se puede obtener leyendo de derecha a izquierda la cadena original. Sin embargo, esto no es cierto para toda cadena, como lo ilustran las cadenas 3.4(c) y 3.4(d), mismas que tienen algún elemento "0".

Cuando se recorre una curva para obtener sus elementos de cadena y se encuentran elementos cero, se necesita conocer cual elemento no-cero fue el último, para poder definir el siguiente elemento (lo cual permite conocer el ángulo recto inmediato anterior). En el caso mostrado en la figura 3.4(c), el primer elemento "2" encontrado después de la secuencia de elementos "0" fue obtenido con referencia al elemento previo "2" el cual no es "0" y forma un ángulo recto. Así, el inverso de la cadena mostrado en la figura 3.4(c) corresponde a la cadena presentada en (d), nótese que si se leyera la cadena original en sentido contrario (de derecha a izquierda) el elemento que antecede la secuencia de "0" (en este caso "2") se ve como si "saltara" sobre ella para obtener el inverso de la cadena.



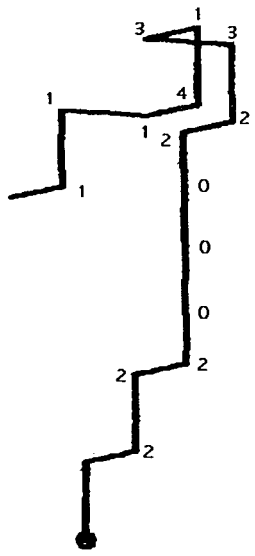
333431132

(a)



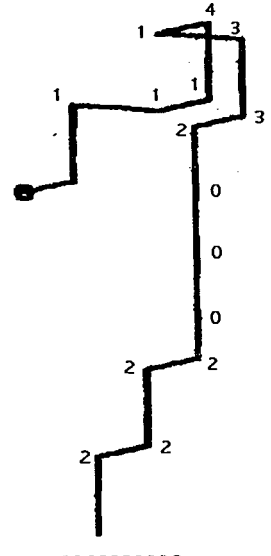
231134333

(b)



222000223314111

(c)



111413320002222

(d)

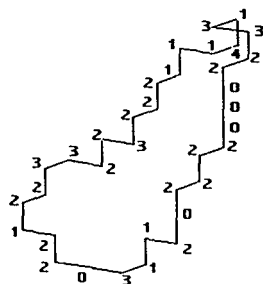
Fig. 3.4. El inverso de una cadena: (a) una curva discreta y su cadena; (b) el inverso de la cadena presentada en (a); (c) una curva discreta y su cadena, la cual contiene algunos elementos cero; (d) el inverso de la cadena mostrada en (c).

3.5 Normalización respecto al punto de inicio para curvas abiertas

Usando el concepto del inverso de una cadena, esta notación puede ser normalizada bajo el punto de inicio, escogiendo éste de tal manera que la secuencia de elementos resultantes sea, como número en base 5, de magnitud mínima. Por ejemplo, la cadena de la curva abierta mostrada en la figura 3.4(a) representa el siguiente número entero: 333431132; y la cadena mostrada en (b) representa el número 231134333, respectivamente. Entonces el entero de magnitud mínima corresponde a la cadena mostrada en la figura 3.4(b). Así la cadena es normalizada bajo el punto de inicio.

3.6 Normalización respecto al punto de inicio para curvas cerradas.

Las curvas cerradas descritas bajo este código de cadena pueden ser hechas invariantes bajo punto de inicio, escogiéndolo de tal manera que la secuencia de elementos resultante forme un número entero de magnitud mínima.



(a)

00022331411122232233221220311202222

(b)

Fig. 3.5. Normalización respecto al punto de inicio para curvas cerradas: (a) un ejemplo de una curva discreta cerrada y sus correspondientes elementos. Dependiendo del vértice que se tome como origen y del sentido del recorrido se determinan distintas cadenas para la curva, interpretándolas como números en base 5, se toma la menor como la representación canónica o normal de la curva; (b) la cadena de la curva cerrada mostrada en (a), la cual ya está normalizada respecto al punto de inicio.

La cadena de la curva discreta en \mathbb{R}^3 presentada en la figura 3.5(a) puede ser normalizada respecto a su punto de inicio rotando los dígitos hasta encontrar el número en base 5 de magnitud mínima. Finalmente la figura 3.5(b) muestra la cadena de la curva discreta mostrada en la figura 3.5(a), la cual está normalizada con respecto al punto de inicio.

3.7 Transformación espejo (reflexión en un plano)

La reflexión de una curva en un plano (espejo), produce otra curva. La cadena de ésta puede obtenerse de la primera reemplazando los elementos "1" por elementos "3" y viceversa. Se dirá que se trata de una transformación espejo y se llamará cadena espejo a la cadena resultante.

Definición 4. La cadena espejo de una cadena A es la cadena que se obtiene reemplazando los elementos '1' de A por elementos '3' y viceversa.

La figura 3.6 ilustra esta transformación. En la figura 3.6(a) el plano espejo es paralelo al plano 'XY'. En la figura 3.6(b) el plano de reflexión es paralelo al plano 'XZ' y en (c) es paralelo al plano 'YZ'.

3.8 Comparación de curvas

Definición 5. Dos curvas discretas en \mathbb{R}^3 son isométricas si tienen la misma longitud.

Usando las invarianzas arriba mencionadas tenemos un único descriptor de curva basado en el código de cadena. Entonces para determinar si dos curvas isométricas tienen la misma forma, solo es necesario ver si sus cadenas son iguales. También, es posible decidir si una forma local ocurre dentro de otra forma comparando sus respectivas cadenas.

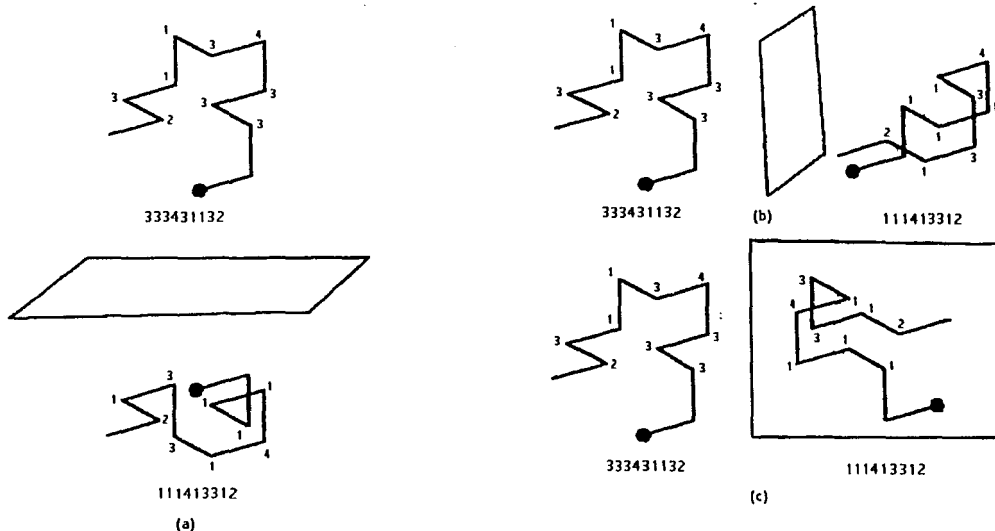


Fig. 3.6. Transformación espejo: (a) el plano espejo es paralelo al plano estándar "XY", (b) reflexión en el plano "XZ", (c) reflexión en el plano "YZ".

En este capítulo se ha presentado el código de cadena para representar curvas discretas en 3D. Este código es invariante bajo rotación y translación. La cadena normalizada respecto al punto de inicio proporciona un único descriptor, el cual facilita la comparación de curvas.

Se ha presentado un número de conceptos, definiciones y ejemplos, los cuales permiten encontrar algunas propiedades interesantes de la curva, tales como: comparación de curvas, representación de curvas discretas, y como último concepto, se presenta la representación de objetos para sólidos. Se puede usar este código de cadena para representar sólidos, por ejemplo el Modelo de Elevación Digital mostrado en la figura 3.7. Esto, además, puede ser extendido para representar imágenes digitales, de lo cual se hablará en los capítulos siguientes.

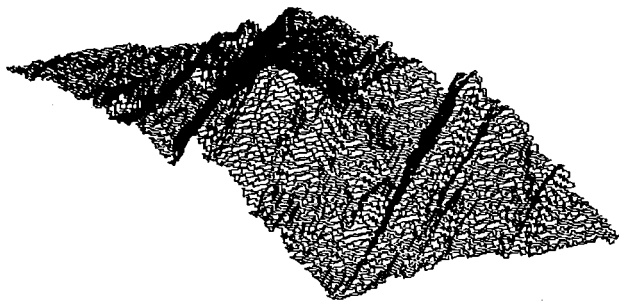


Fig. 3.7. El volcán Iztaccihuatl representado por una curva discreta en 3D, misma que tiene asociado su código de cadena.

CAPITULO 4.

APLICACIÓN DEL CÓDIGO DE CADENA EN 3D PARA REPRESENTACIÓN DE IMÁGENES DIGITALES.

En éste capítulo se pretende mostrar el procedimiento y metodología que se usa en el sistema realizado al lo largo de esta tesis, para representar imágenes digitales como una cadena en 3D. Cabe mencionar que ésta es la parte central de éste trabajo, puesto que es un método nuevo propuesto para el análisis de formas.

En cada sección se presentará una etapa del proceso por el que pasa la imagen hasta obtener su cadena.

4.1 Formato de la imagen

El sistema realizado necesita como entrada una *imagen digital* (ver apéndice). Por lo tanto, se usaron los formatos de imágenes PGM o PPM, ya que éstos definen imágenes digitales. Estos formatos de imágenes contienen un encabezado, el cual indica el tamaño de la imagen, y el rango de niveles de gris que se encuentran en la imagen (256, en la mayoría de los casos), y los datos en binario.

La imagen digital se verá como una matriz de dos dimensiones, donde cada elemento de la matriz da el valor de cada pixel. El valor que tome cada pixel es el nivel de gris.

Cualquier imagen, sencilla o compleja, en los formatos conocidos (como jpeg, bmp, tiff, pcx, etc.), puede ser convertida a alguno de los formatos que se usan (ppm o pgm , como ya se mencionó).

En la figura 4.1 se muestra un ejemplo de imagen en su formato original, jpg, y su imagen digital en formato pgm. En la figura 4.2 se muestra otra figura 4.en formato bmp, y

su imagen digital (formato pgm). Observe que aunque son formatos diferentes no hay perdida de información respecto a la forma de los objetos contenidos en ellas.



Fig. 4.1. Una imagen en formato JPG y su respectiva imagen digital.



Fig. 4.2. Una imagen en formato BMP y su respectiva imagen digital (cuadros.pgm).

4.2 Conversión de la imagen a una matriz bidimensional

Como ya se mencionó, el sistema necesita como entrada una imagen que pueda ser procesada, como una matriz de dos dimensiones, en la cual cada elemento representa el nivel de gris de cada pixel de la imagen. Por lo tanto, las imágenes (en tonos grisáceos) mostrados en las figuras 4.1 y 4.2, son una muestra de las imágenes de entrada del sistema.

Tomando el ejemplo de la imagen en niveles de gris de la figura 4.2, en la figura 4.3 se muestra la matriz bidimensional generada a partir de esta imagen de entrada, a la cual se le llamará cuadros.pgm.

Se debe tomar en cuenta que cada valor que aparece en la matriz, es un "valor normalizado" a partir del valor original de nivel de gris. Este valor lo da la siguiente función:

$$y = \text{ceil} \left(N * \frac{x - \text{minval}}{\text{maxval} - \text{minval}} \right) \quad (1)$$

Donde x es el valor original de nivel de gris de un pixel; N es el parámetro de normalización (el número máximo que puede tomar un número en la matriz); maxval y minval son los valores máximo y mínimo de nivel de gris que se encuentran en la imagen; ceil es la función Mínimo Entero Superior usada en matemáticas; y finalmente y , es el nuevo valor que tomará parte de la matriz.

1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	1
1	2	5	5	5	5	5	5	2	1
1	2	5	9	9	9	9	5	2	1
1	2	5	9	10	10	9	5	2	1
1	2	5	9	10	10	9	5	2	1
1	2	5	9	9	9	9	5	2	1
1	2	5	5	5	5	5	5	2	1
1	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1

Fig. 4.3. Matriz bidimensional con los valores normalizados de nivel de gris de cada pixel de la imagen cuadros.pgm

En éste ejemplo, el parámetro de normalización es $N = 10$; los valores máximo y mínimo de nivel de gris encontrados en la imagen son $\text{maxval} = 202$ y $\text{minval} = 48$; respectivamente.

Cabe mencionar que después de que la imagen es procesada y convertida a una matriz bidimensional, como se muestra en la figura 4.3, el valor máximo que se puede encontrar en ella es N , y el valor mínimo siempre será 1.

Entonces, los valores originales de nivel de gris por pixel, no son los que van a formar parte de la matriz, ya que pasarán por la función de normalización y los resultantes formarán parte de ella. El criterio de normalización será dado por el usuario del sistema, permitiéndole así la manipulación de resultados para satisfacer sus necesidades particulares.

Como se ha visto, el segundo paso en el proceso de conversión de una imagen a una cadena en 3D, es la representación de la imagen digital en una matriz bidimensional con valores enteros (a la cual se le llamará *matriz de valores*).

4.3 Binarización

Después de que la imagen es procesada y convertida a una *matriz de valores* de dos dimensiones, ésta a su vez es convertida a una matriz de tres dimensiones.

Se ha visto que un valor en la matriz de valores representa el nivel de gris de un pixel en la imagen digital, el cual es un número que indicará la altura que deberá representar en la matriz de tres dimensiones a partir de la superficie de la misma. En la figura 4.4 se muestra una representación de la matriz de tres dimensiones que se obtiene a partir de la matriz de valores.

La manera en que es convertida la matriz de valores de dos dimensiones a una de tres dimensiones es la siguiente: cada renglón de la matriz de valores generará un arreglo de dos dimensiones y todos estos arreglos de dos dimensiones, a su vez, formarán un arreglo de tres dimensiones, el cual se puede ver como una matriz en tercera dimensión cuyas dimensiones son: capa, renglón y columna.

El número de renglones de una capa será el número de elementos o valores que se encuentren en cada renglón de la matriz de valores, es decir, el número de columnas de la matriz de valores y el número de columnas estará dado por el parámetro de normalización; por lo tanto, cada una de las capas que formen parte de la matriz

tridimensional tendrá como número de renglones el número de columnas de la matriz de valores y como número de columnas el número dado por el usuario como parámetro de normalización.

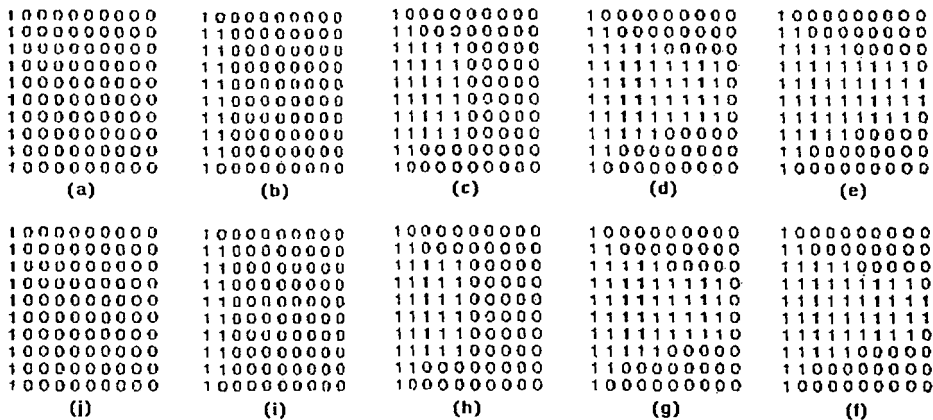


Fig. 4.4. Representación de una matriz de tres dimensiones (10x10x10) como varias matrices de dos dimensiones (10 matrices de 10x10 en este ejemplo). Cada matriz mostrada representa una capa de la matriz tridimensional.

Ahora, cada número o valor de nivel de gris que aparece en un renglón de la matriz de valores, formará un renglón de 1's y 0's, en una matriz que formará parte de la matriz tridimensional. El número de 1's estará dado por el valor que se tome del renglón de la matriz de valores y el número de 0's, estará dado por el número que resulte de restar el valor del parámetro de normalización del valor de la matriz de valores que sea tomado.

En el ejemplo anterior, la matriz de valores es una matriz de 10 renglones por 10 columnas, como cada uno de los renglones genera un arreglo de dos dimensiones, habrá 10 arreglos de dos dimensiones en la matriz tridimensional. Como el número de columnas de la matriz de valores es 10, cada arreglo de dos dimensiones tendrá 10

renglones; y como el parámetro de normalización es 10, éste será el número de columnas que tenga cada uno de dichos arreglos que formarán parte de la matriz de tres dimensiones. Por lo tanto, la matriz de tres dimensiones será una matriz de 10 renglones por 10 columnas por 10 niveles de profundidad.

Siguiendo con el mismo ejemplo, el primer elemento del primer renglón de la matriz de valores es un 1, por lo tanto el primer renglón de la primera matriz generada estará formado por un 1 en la primera posición y 9 0's en las demás posiciones. El primer renglón de la matriz de valores solo contiene 1's, por lo que, todos los renglones de la primera matriz serán iguales, ésta matriz se muestra en la figura 4.4(a). El segundo renglón, de la matriz de valores tiene como primer elemento un 1, por lo tanto este renglón será como cualquiera de los renglones de la primera matriz, el segundo elemento del segundo renglón de la matriz de valores es un 2, por lo que, el segundo renglón de la segunda matriz tendrá dos 1's y ocho 0's y cada renglón se va generando de la misma manera, la figura 4.4(b) muestra como queda esta segunda matriz generada con los valores del segundo renglón de la matriz de valores. Todas las matrices que formarán la matriz tridimensional son generadas de la misma manera, las figuras 4.4(a)-4.4(j) muestran todas las matrices generadas a partir de los valores de la matriz de valores.

La matriz tridimensional generada y formada solamente por los elementos 1 y 0 será llamada la *matriz binaria*.

La matriz binaria puede ser representada por un conjunto de voxels (ver apéndice).

Cada elemento 1 de la matriz binaria puede ser representado por un voxel y los elementos 0 no aparecerán; de esta manera la imagen original será representada como una forma en tres dimensiones.

En este ejemplo, la primera matriz será representada por una hilera de diez voxels; la segunda matriz será representada por una hilera de diez voxels y otra hilera de ocho voxels; la tercer matriz será representada por cinco hileras de voxels: una de diez, una de 8 y tres de 6; y así sucesivamente; como la matriz de valores es simétrica (en este ejemplo), las matrices 6^{a} a 10^{a} serán representadas de la misma manera que las matrices 1^{a} a 5^{a} . La figura 4.5 muestra la representación de cómo se vería la matriz binaria con voxels. La figura 4.6 muestra otra vista (desde arriba) de la misma representación, en ésta se puede apreciar mejor el parecido con la imagen original.

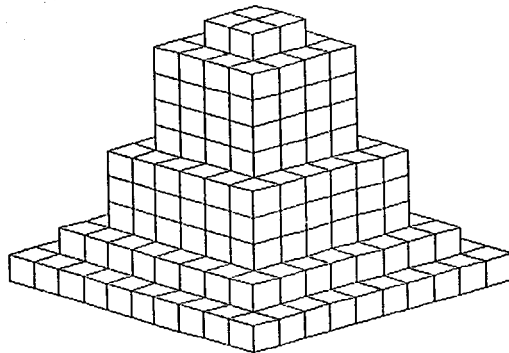


Fig. 4.5. La matriz binaria representada con voxels. Cada voxel representa un elemento 1 de la matriz binaria.

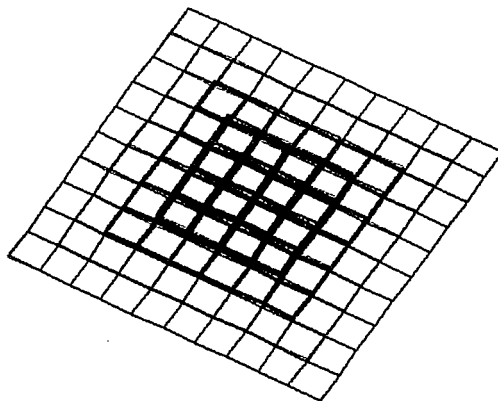


Fig. 4.6. Una vista diferente de la representación con voxels de la matriz binaria.

En esta tesis el método de voxelización sólo se usa como una manera de ver la matriz binaria como una forma en tres dimensiones.

Como se ha visto hasta el momento, la tercera etapa del proceso de conversión de una imagen en una cadena, es la representación de la matriz de valores de dos dimensiones en una matriz binaria de tres dimensiones. Este procedimiento es llamado *binarización*.

4.4 Obtención de la cadena en 3D

En esta sección se mostrará cómo una imagen digital puede ser representada como una cadena en 3D, usando el código de cadena visto en la sección anterior. Se tomará en cuenta que la matriz binaria se ve como un arreglo de voxels, los cuales serán recorridos por sus bordes o contornos, de esta manera un borde de un voxel es visto como un segmento de línea recta de longitud 1.

En la figura 4.7 se muestran los elementos del código de cadena, visto en el capítulo anterior, que usaremos para esta representación.

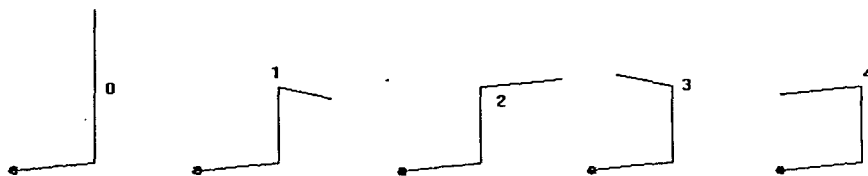


Fig. 4.7. Los elementos del código de cadena.

El procedimiento que se sigue para obtener la cadena de una imagen es el siguiente:

1. Se toman dos segmentos de línea recta arbitrarios de longitud 1, tales que formen un ángulo recto para definir el primer elemento de la cadena, el cual estará formado por tres segmentos: los dos mencionados y el borde de un voxel, mismo que se selecciona en el siguiente paso.

2. Conviene ver cada matriz bidimensional o capa como si estuviera rotada 90° , como se muestra en la figura 4.8. Se toma la primer capa de la matriz tridimensional binaria, recorriendo el borde de los voxels representados por los elementos 1 de dicha capa, de izquierda a derecha, poniendo un segmento recto de longitud uno cada vez que se encuentre un voxel en el borde, como se muestra en la figura 4.9(a).
3. A partir del último segmento colocado en el borde de los voxels representados por los elementos 1 de la primera capa, se sigue el borde del voxel que permite "alcanzar" la siguiente matriz.
4. La segunda capa es recorrida en orden inverso, de derecha a izquierda, como se muestra en la figura 4.9(b).
5. Se repiten los pasos 2, 3 y 4 hasta llegar a la última capa. Se puede notar que si numeramos cada una de las capas a partir de 1, las capas cuyo índice sea impar, serán recorridas de izquierda a derecha, y las de índice par en sentido contrario.
6. Al terminar el recorrido, se tendrá una forma discreta en 3D. A esta forma discreta se le construye su código de cadena. La figura 4.10 muestra la forma discreta del ejemplo considerado (en las figuras 4.11-4.13 se muestran varias vistas de la misma forma). La figura 4.14 muestra la cadena correspondiente.

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

Fig. 4.8. La matriz o capa de la izquierda es rotada 90° , de manera que se vea como la de la derecha.

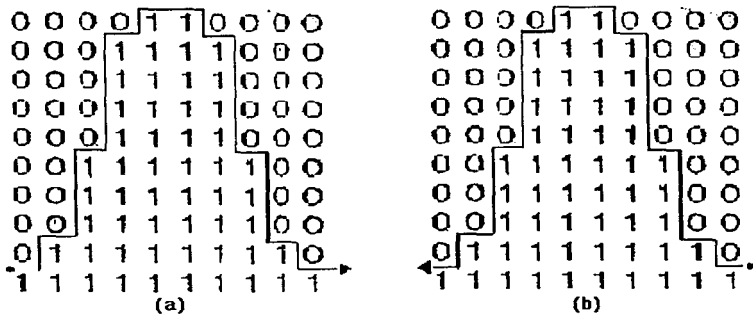


Fig. 4.9. (a) El borde que forman los voxels representados por elementos 1 de la capa, es recorrido de izquierda a derecha. (b) El borde que forman los voxels de la capa es recorrido en orden inverso que en (a), es decir, de derecha a izquierda.

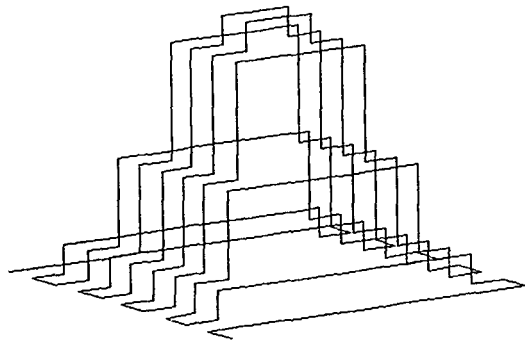


Fig. 4.10. La forma discreta de la matriz tridimensional recorrida por borde de los voxels representados por 1's.

Como se puede observar en el código de cadena (mostrado en la figura 4.7), los elementos 1 y 3, nos indican un cambio de plano.

Al recorrer cada capa, el procedimiento se ve como si camináramos en un plano y al cambiar de capa, se ve como si cambiáramos de plano. Por lo tanto al caminar sobre un mismo plano, solo obtendremos elementos 0,2 y 4 y en cuanto se presenta un cambio de plano se obtendrá elementos 1 y 3.

A partir de la cadena, la forma discreta puede ser obtenida, así como las coordenadas en 3D (conviniendo en colocar el punto de inicio en el origen). En las figuras 4.15 - 4.18 se muestran varias imágenes, con su correspondiente imagen digital, parte de su correspondiente cadena (esta no es mostrada completamente ya que ocuparía varias páginas) y la forma discreta obtenida a partir de ella.

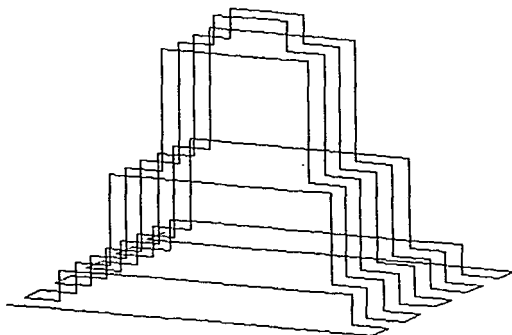


Fig. 4.11. Otra vista de la forma discreta mostrada en la figura 4.10.

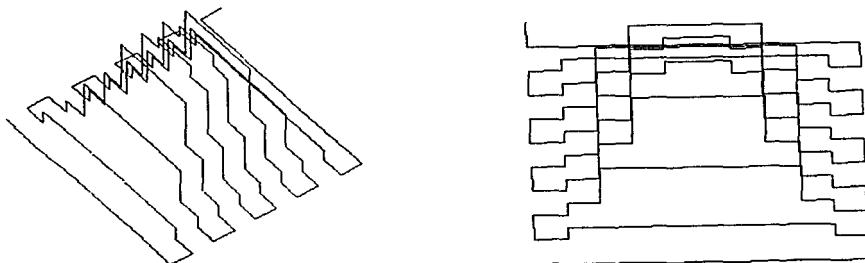


Fig. 4.12-4.13. Vistas de la forma mostrada en la figura 4.10.

```

200000000024120000000421432200200000400222341220022000200040002200
222143220022000222042200022002223412200220002220422000220022214322
002200020004000220022234122002000004002221432000000042340000000002
    
```

Fig. 4.14. Cadena obtenida de la forma discreta mostrada en las figuras 4.10-4.13.

En este capítulo se ha mostrado el proceso para representar una imagen digital como una cadena en 3D, usando el código de cadena visto en el capítulo anterior.

Como se pudo observar, una imagen que consta de tres dimensiones, ancho, largo y profundidad dada por la escala de gris de los píxeles que la forman puede ser representada como un arreglo de una sola dimensión, lo cual nos permite usar técnicas de compactación; como por ejemplo, cada elemento de la cadena puede ser representado por tres bits, lo que reduciría en gran medida el tamaño de la información.

Esta representación también nos permite usar técnicas de búsqueda y extracción de objetos, como se verá en el siguiente capítulo.

CAPITULO 5.

RESULTADOS, APLICACIONES Y PROPUESTAS DE TRABAJOS A FUTURO

En este capítulo se muestran varios resultados que se han considerado importantes. También se proponen aplicaciones y trabajos que se pueden realizar a partir de este trabajo.

5.1 Resultados

Como se mostró en el capítulo anterior, una imagen digital, que puede ser vista como una entidad de tres dimensiones, dadas por el ancho, largo y la profundidad, obtenida por la escala de niveles de gris, es representada por una cadena, que se puede ver como un ente unidimensional. Esta propiedad es muy importante, ya que nos permite la realización de técnicas de *compactación* (ver apéndice).

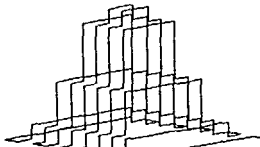
5.1.1 Compactación

* Una técnica de compactación propuesta consiste en representar los dígitos de la cadena en notación binaria tradicional. Puesto que el dígito más grande del código de cadena es 4, solo se necesitarán 3 bits para representar cada uno de los dígitos. Por lo tanto, como un byte está formado de 8 bits, esto quiere decir, que podremos representar 8 dígitos en 3 bytes. Así, el elemento '0' será representado por los bits '000', el elemento 1 por los bits '001', el elemento '2' por '010', el elemento '3' por '011' y por último el elemento '4' por '100'.

Continuando con el ejemplo del capítulo anterior, en la figura 5.1(a) se muestra la imagen digital, en la figura 5.1(b) se muestra la representación de (a) como curva discreta, en la figura 5.1(c) se muestra la cadena correspondiente y por último, en la figura 5.1(d) se muestra la cadena representada con bits usando el método de compactación propuesto. La longitud de la cadena es 198, por lo que con el método de compactación se tendrán 594 bits, o bien 75 bytes, lo cual reduce considerablemente el tamaño de la cadena y de la imagen digital (que es de 132 bytes).



(a)



(b)

```
20000000002412000000421432200200000400222341220022000200040002200
222143220022000222042200022002223412200220002220422000220022214322
002200020004000220022234122002000004002221432000000042340000000002
```

(c)

```
01 0000000000000000000000000000010100001010000000000000000000100
01000110001101001000000010000000000000010000000001001001001100
0010100100000000100100000000001000000000100000000000000000000000
0100100100011000110100100000001001000000000100100100000000000010
0000000001001000000001001001001110000101001000000010010000000000
01001001000010001001000000000010010000000010010010001100011010010
000000010010000000000100000000010000000000100000000001001000000010010010
0111000010100100000001000000000000000100000000010010010001100011
0100000000000000000001000100111000000000000000000000000000000010
```

(d)

Fig. 5.1. (a) Imagen digital. (b) La forma discreta de (a); (c) Cadena de (a); (d) Representación en bits de la cadena en (c).

5.1.2 Búsqueda de plantillas: uso de un editor de cadenas

Durante este trabajo también se realizó un editor de cadenas. Este editor de cadenas permite encontrar subcadenas que se repiten frecuentemente en una cadena. La manera en que éste editor opera es la siguiente:

1. El editor recibe un archivo en el cual se encuentra la cadena obtenida usando el sistema descrito en el capítulo anterior.
2. Dado un número N definido por el usuario, el editor tomará una subcadena de longitud N a partir del primer elemento de la cadena.
3. El editor buscará secuencialmente, dicha subcadena a lo largo de toda la cadena dada. Cabe mencionar que el usuario puede definir cada cuántos elementos la subcadena avanzará en cada paso (por ejemplo cada 2 o 5 elementos).
4. Dada una función de incremento para N (como por ejemplo $N=N+1$, $N=N*2$, $N=N*N$, etc.), el editor iniciará una nueva búsqueda a lo largo de toda la cadena, pero ahora con una subcadena de longitud diferente, definida por dicha función.
5. La búsqueda termina en dos casos: (a) cuando el usuario define un "tope" de búsqueda, como por ejemplo el número de veces que la subcadena debe buscarse en la cadena o cuando el tamaño de la subcadena toma una longitud específica; o (b) cuando el último elemento de la subcadena coincide con el último elemento de la cadena, en este caso se iniciará una nueva búsqueda o se terminará el proceso.
6. El editor termina reportando cuales fueron las subcadenas que se encontraron dentro de la cadena mas de una vez y su respectiva longitud, pero no son tomadas en cuenta las cadenas que solo contienen elementos 0.

La búsqueda de patrones en imágenes es una parte muy importante en el área de visión por computadora, sin embargo es un tema difícil de atacar debido al gran trabajo que esto implica y no siempre se obtienen los mejores resultados. En éste sistema, el editor trabaja bien cuando los objetos en la imagen tienen una forma "sencilla", es decir, cuando se encuentran bien definidas las fronteras de los objetos y éstos aparecen completos (no se intersectan, no son solo parte de un objeto) y la imagen tiene un color de fondo uniforme, por ejemplo cuando en la imagen se encuentran figuras geométricas, como se muestra en la figura 5.2.

La figura 5.2(b) muestra la forma discreta de la imagen digital mostrada en la figura 5.2(a). Se puede apreciar que hay una subcadena que se repite en la forma, que es la mostrada en 2(d). En la figura 5.2(e) se muestra la cadena de la forma mostrada en 2(d), la cual se repite seis veces en toda la cadena. De esta manera también se puede definir

algún método de compactación de información. Ya que si un patrón se repite muchas veces en una imagen, este puede ser aislado y sustituido por cierta clave en los lugares donde aparezca el patrón en la cadena. De esta manera nos quedará una cadena más compacta.

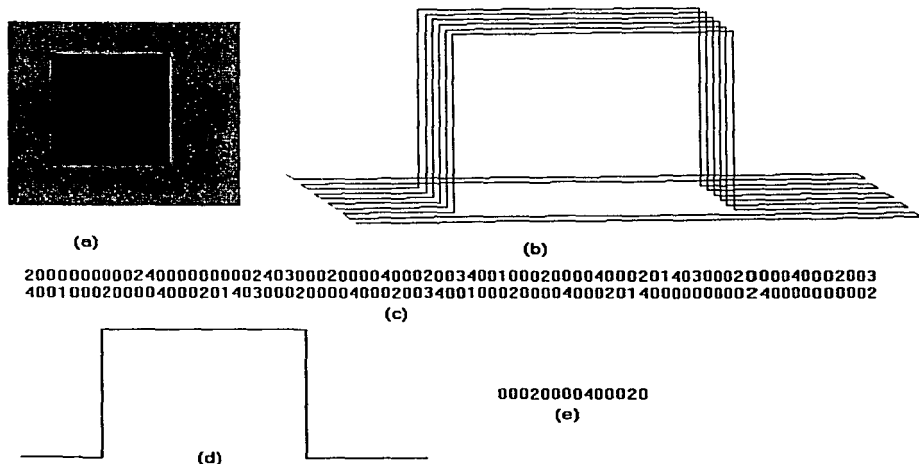


Fig. 5.2. (a) Una imagen digital "sencilla". (b) La forma discreta de la imagen digital. (c) La cadena de la forma mostrada en (b). (d) El patrón que se repite en la forma. (e) La cadena de la forma en (d).

En imágenes que no son "sencillas", es difícil encontrar un patrón que se repita, sin embargo, los que son encontrados, normalmente están formados por una secuencia de 0's y al principio y/o al final otro elemento del código de cadena diferente de 0, lo cual permite proponer un método de segmentación que se verá mas adelante.

El mismo sistema que encuentra subcadenas que se repiten dentro de la cadena, permite al usuario introducir una cadena, para revisar si el patrón que ésta representa se encuentra dentro de la imagen. Aunque es un método poco eficiente, es una manera de buscar *plantillas* (ver apéndice) dentro de la imagen.

5.1.3 Análisis de códigos de cadenas

En las figuras 5.3-5.9 se muestra una forma y sus cadenas obtenidas a partir de los códigos de cadena vistos en el capítulo dos: cadena de Freeman en 4 y 8 direcciones, números de forma, notación de cambio de inclinación (NCI) y código de cadena basado en vértices (CCV). También se hace el análisis de la misma con coordenadas cartesianas y finalmente con el código de cadena en 3D aplicado en esta tesis.

Como se puede observar, la forma presentada en cada una de estas figuras es la misma, esto se hace con la finalidad de realizar un análisis cualitativo de dichos métodos.

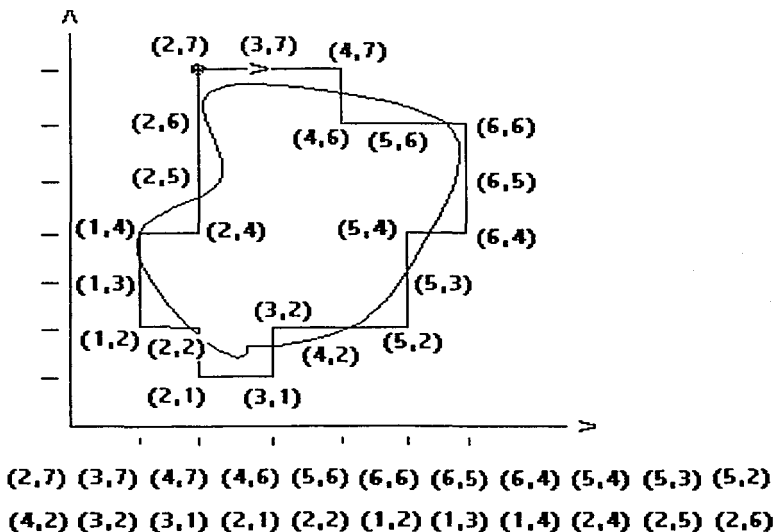


Fig. 5.3. Coordenadas Cartesianas

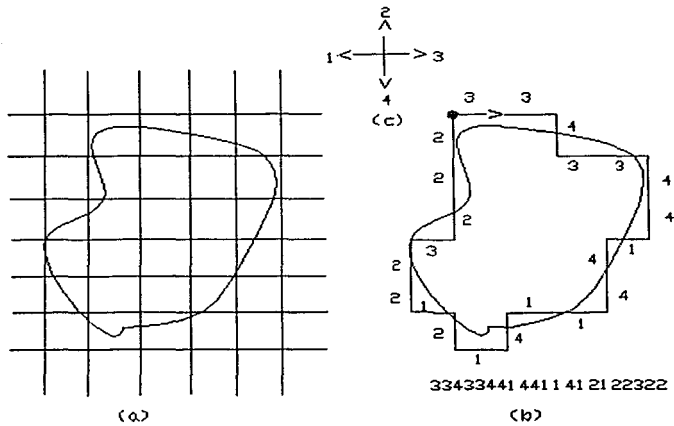


Fig. 5.4. Cadena de Freeman en 4 direcciones

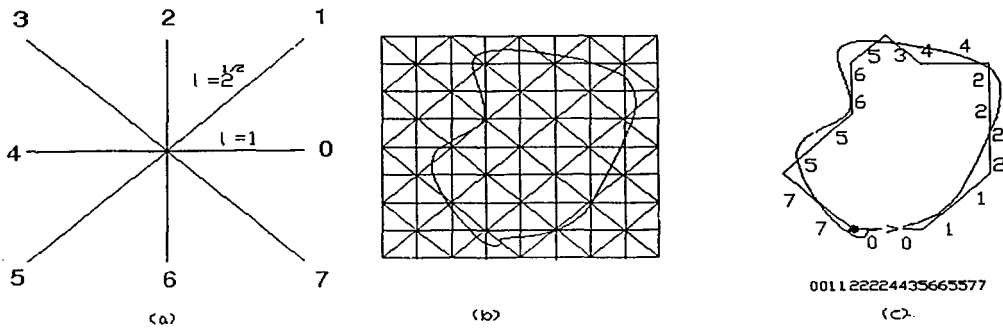


Fig. 5.5. Cadena de Freeman en 8 direcciones

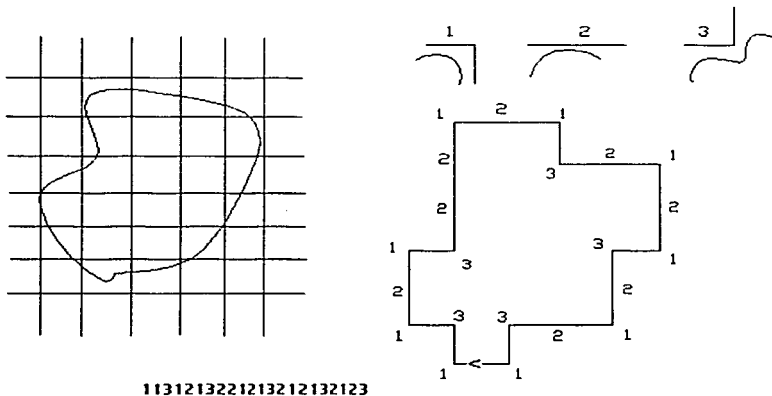


Fig. 5.6. Números de Forma

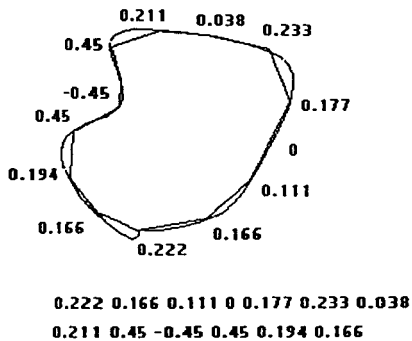


Fig. 5.7. Notación Cambio de Inclinación

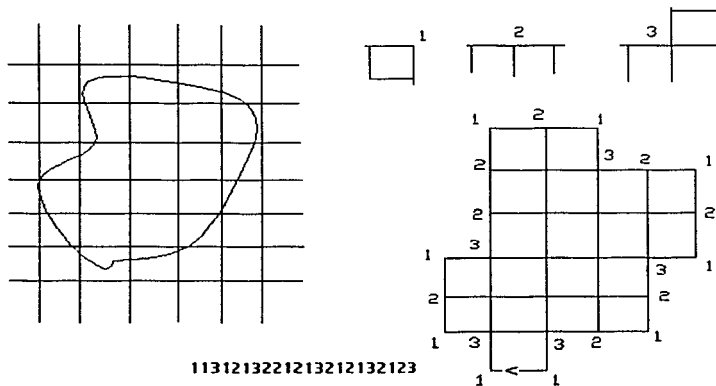


Fig. 5.8. Código de Cadena basado en Vértices

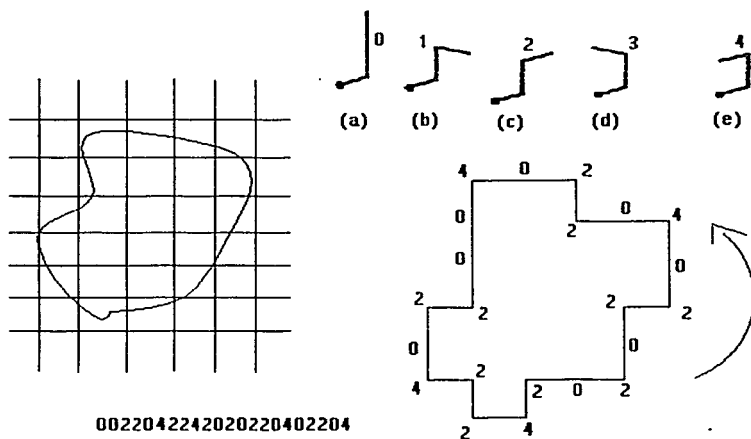


Fig. 5.9. Código de cadena en 3D

Al obtener las respectivas cadenas con cada uno de dichos códigos, se puede observar lo siguiente:

1. Los métodos de representación de formas por medio de coordenadas cartesianas y cadenas de Freeman son dependientes bajo translación, rotación, y sus cadenas no son normalizadas con respecto al origen.
2. Los códigos de cadena basados en número de forma, cambio de inclinación, en vértices y el propuesto en 3D son invariantes bajo rotación, translación y la cadena puede ser normalizada con respecto al origen.
3. Al representar una forma por medio de sus coordenadas cartesianas se puede observar que no hay pérdida de información ya que cada punto de la forma es independiente de los demás, así como los números de forma y el código de cadena basado en vértices. En los códigos de cadena propuesto por Freeman no hay pérdida de información si siempre se conoce el sentido en el que se está recorriendo la forma. En los demás métodos la pérdida de información es más probable, ya que cada uno de los elementos de la cadena depende del anterior.
4. En estas figuras se toma en cuenta que cada cuadro de la rejilla que se usa representa un pixel, por lo que en ciertos métodos la exactitud de la representación de la forma depende del método que se aplique, por ejemplo: el código de Freeman en 8 direcciones es más preciso que el de cuatro direcciones; el número de forma y el CCV dan exactamente la misma precisión; el NCI da una precisión muy buena cuando el tamaño del segmento de línea recta que se elige es muy pequeño. El código de cadena en 3D para formas planas funciona de la misma manera que los números de forma.
5. Se puede ver que la longitud de las cadenas en todos los casos se asemeja mucho, a excepción de las coordenadas cartesianas que doblan en longitud las cadenas si se ve cada coordenada como un par de números enteros. Por lo que al aplicar el método de compactación mostrado en el capítulo cinco en cada uno de los casos, el porcentaje de compactación de datos sería similar.

6. El uso de éstos códigos de cadena en el área de reconocimiento de patrones ha sido importante gracias a la precisión que dan, si se toma como unidad de procesamiento el pixel.
7. Si se toman en cuenta solo formas en 2D éstos códigos en general funcionan bien, pero cuando se habla de formas en 3D, el método aplicado en esta tesis es el ideal para este tipo de procesamiento.

Se pretende que este breve análisis sirva a los investigadores a elegir el código de cadena que más se adecue a sus necesidades, de tal manera que se conozcan las ventajas y desventajas del método a aplicar.

A continuación se presenta una tabla donde se resume las principales características de cada uno de estos métodos.

Método	Invarianza en Traslación	Invarianza en Rotación	Normalización del origen
<i>Coordenadas cartesianas</i>	NO	NO	NO
<i>Cadenas de Freeman</i>	NO	NO	NO
<i>Números de Forma</i>	SI	SI	SI
<i>Not. Cambio Inclinación</i>	SI	SI	SI
<i>CCV</i>	SI	SI	SI
<i>Código de Cadena en 3D</i>	SI	SI	SI

Método	Sensibilidad al ruido	Compactación de datos	Perdida de Información
<i>Coordenadas cartesianas</i>	Poco	SI	Poca
<i>Cadenas de Freeman</i>	Regular	SI	Regular
<i>Números de Forma</i>	Poco	SI	Poca
<i>Not. Cambio Inclinación</i>	Mucho	SI	Poca
<i>CCV</i>	Poco	SI	Poca
<i>Código de Cadena en 3D</i>	Poco	SI	Regular

Método	Facilitar procesamientos (Eficiencia)	Usos en Reconocimiento de Patrones
<i>Coordenadas cartesianas</i>	Mucho	SI
<i>Cadenas de Freeman</i>	Mucho	SI
<i>Números de Forma</i>	Mucho	SI
<i>Not. Cambio Inclinación</i>	Mucho	SI
<i>CCV</i>	Mucho	SI
<i>Código de Cadena en 3D</i>	Mucho	SI

5.2 Aplicaciones y propuestas de trabajos a futuro

Durante la realización de esta tesis, se proporcionaron algunas imágenes biológicas, las cuales fueron codificadas con el procedimiento ya descrito. Las figuras 5.10-5.14 muestran algunas de ellas, las cuales son imágenes de nucleótidos y parte del proceso de mitosis, que después de ser codificadas servirán a los especialistas de la biomedicina como ayuda para un análisis más sencillo de las mismas, y de esta manera contribuir a su investigación de una forma más eficiente.

En las figuras 5.10-5.13 se muestran las imágenes digitales del proceso de mitosis. La figura 5.14 muestra la imagen digital de nucleótidos. En cada caso se muestra la forma discreta en 3D que representa la cadena de la imagen digital codificada.

La aplicación que se propone realizar en el futuro, es un editor que, con la ayuda de expertos en este tipo de imágenes, permita definir plantillas. Por ejemplo, teniendo la plantilla de un bacilo en particular y varias imágenes en las que sea posible que aparezca este tipo de bacilo, se puede realizar una búsqueda sobre la imagen codificada, ya que una cadena representará la plantilla que se buscará. Cabe mencionar que la plantilla dada puede no aparecer exactamente en las imágenes, pero tal vez haya patrones parecidos, por lo tanto, se tendrá la necesidad de definir una operación de similitud, como la operación "like" usada en los lenguajes de bases de datos. Aunque también es posible definir un procedimiento que mida el grado de similitud entre dos cadenas dadas.

Por otra parte, las imágenes muestran que tiene un color de fondo uniforme, en este caso blanco, por lo que se puede decir que la base de la imagen es "lisa". Cuando se usa el editor de cadenas, las subcadenas que se repiten, frecuentemente están formadas por una secuencia de 0's y al principio y/o al final otro dígito del código de cadena que no es cero. Esto quiere decir que cuando recorremos una secuencia larga de 0's, es como si estuviéramos caminando sobre el piso y en cuanto se encuentra un elemento que no es 0, es que encontramos un objeto. Así, definiendo un proceso que encuentre este tipo de cambios, se podría definir un método de segmentación.

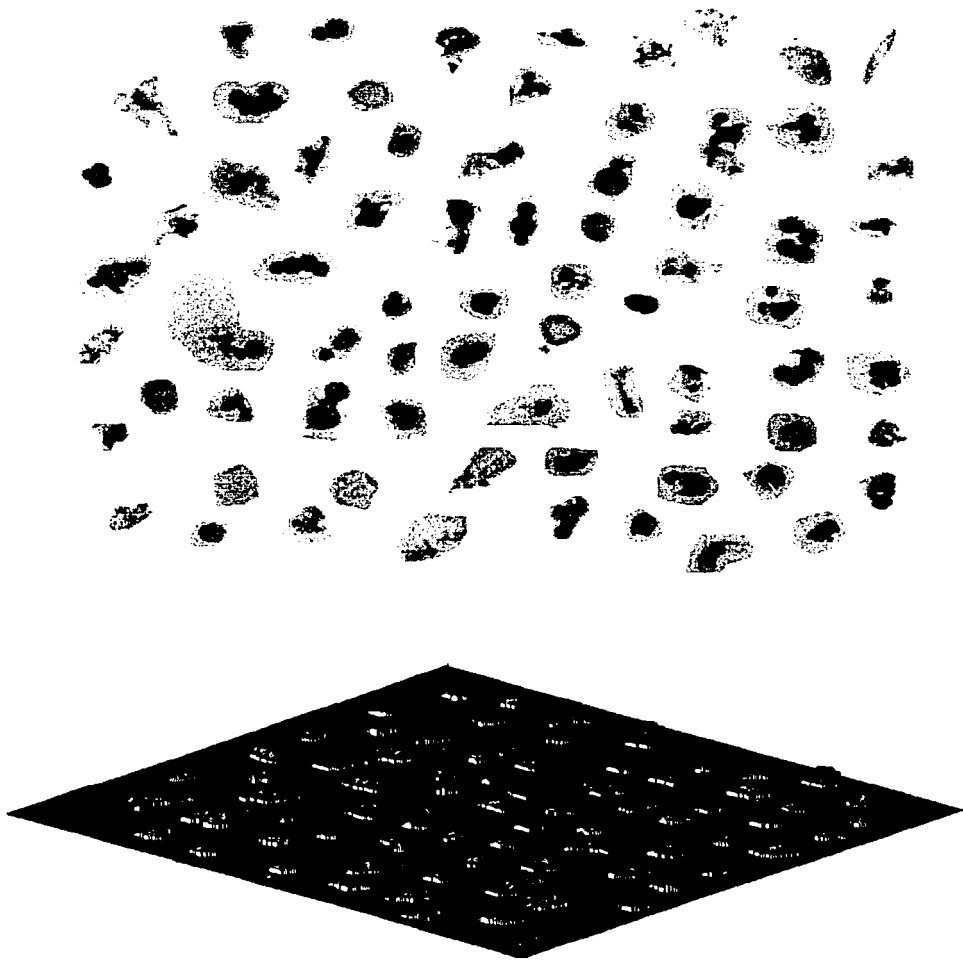


Fig. 5.10. Imagen digital del proceso de mitosis y su correspondiente representación en 3D como curva discreta usando el código de cadena. Esta y las imágenes mostradas en 5.11-5.14 fueron proporcionadas por el Dr. Gabriel Corkidi del Instituto de Biotecnología, UNAM.

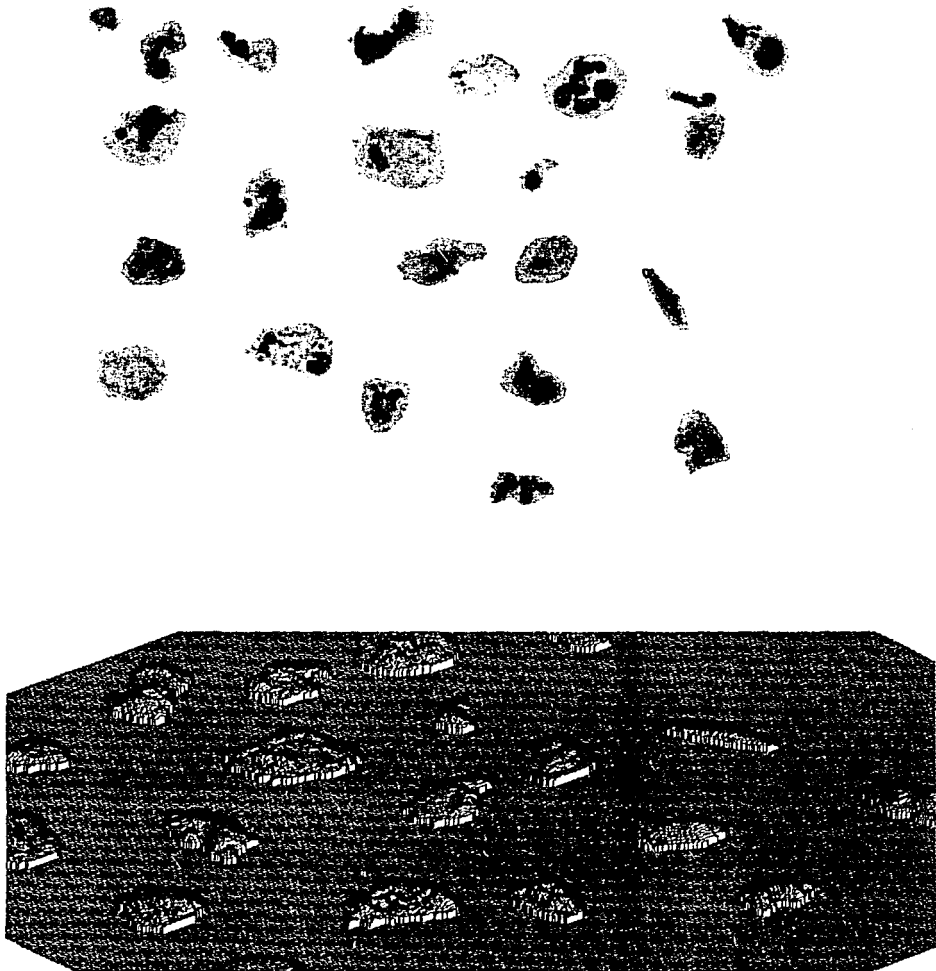


Fig. 5.11. Imagen digital del proceso de mitosis y su respectiva representación en 3D usando el código de cadena.

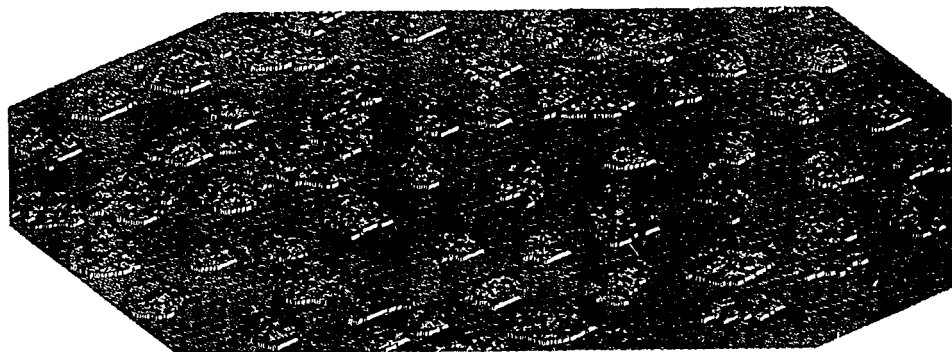
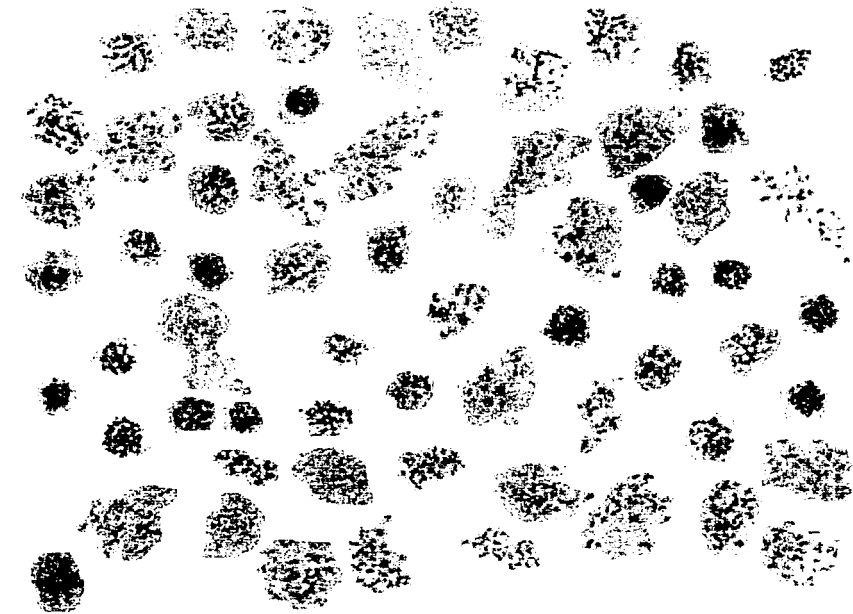


Fig. 5.12. Imagen digital de mitosis y su respectiva representación en 3D usando el código de cadena.

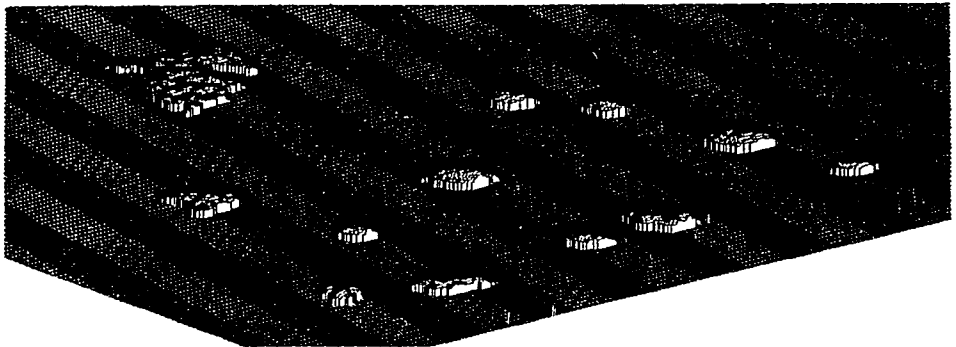
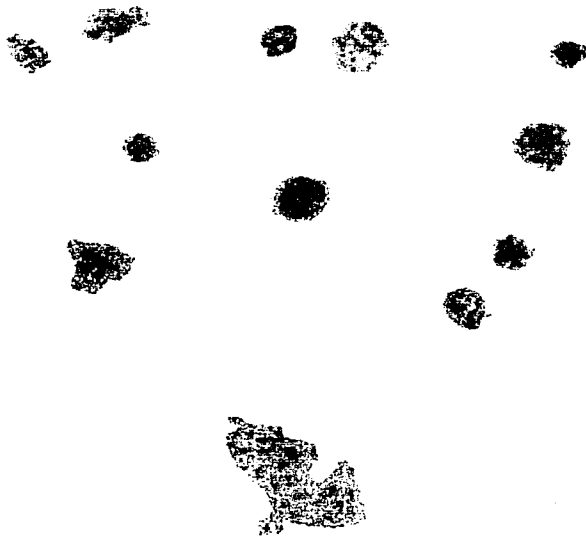


Fig. 5.13. Imagen digital de mitosis y su respectiva representación en 3D usando el código de cadena.

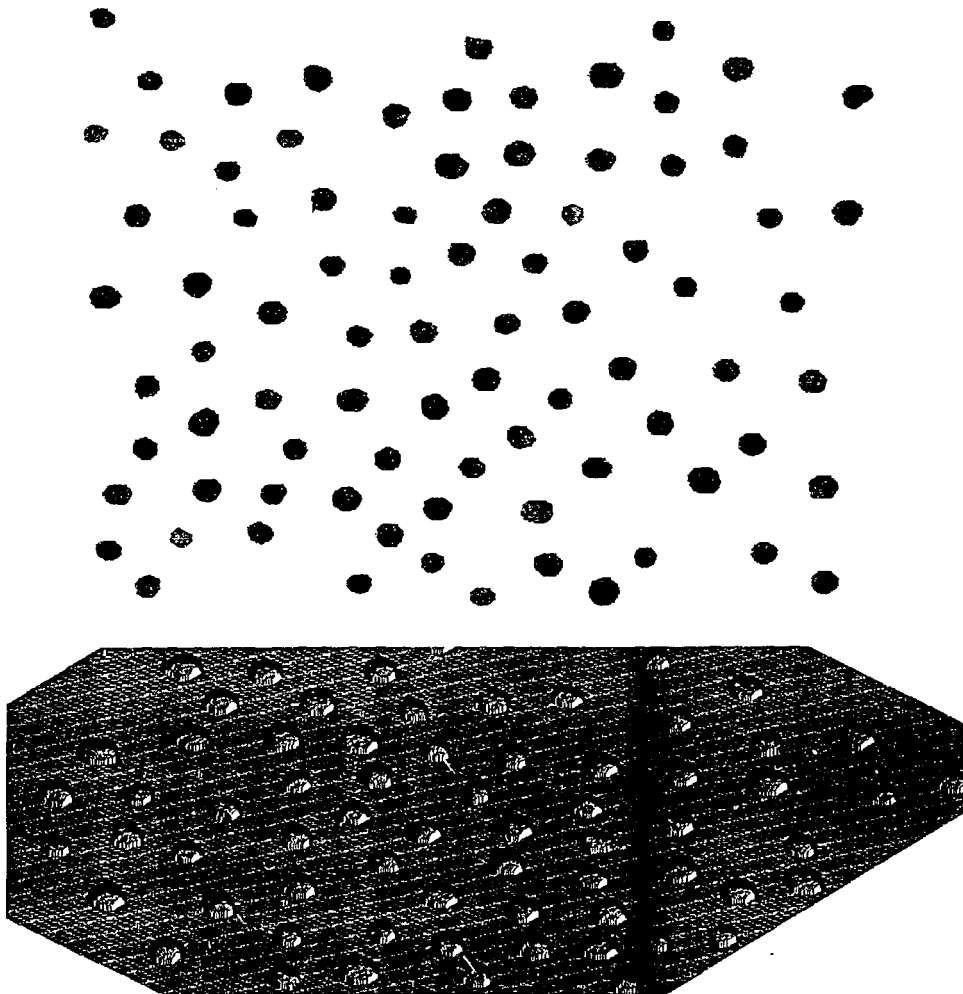


Fig. 5.14. Imagen digital de nucleótidos y su respectiva representación en 3D usando el código de cadena.

5.2.1 Búsqueda de plantillas

Una aplicación que se podría realizar en imágenes como las mencionadas, es la búsqueda de regiones dentro de ella, siguiendo un proceso consistente en marcar la frontera de las regiones que se encuentren, como se muestra en la figura 5.15. De ésta manera se podría definir algún método de extracción de objetos.

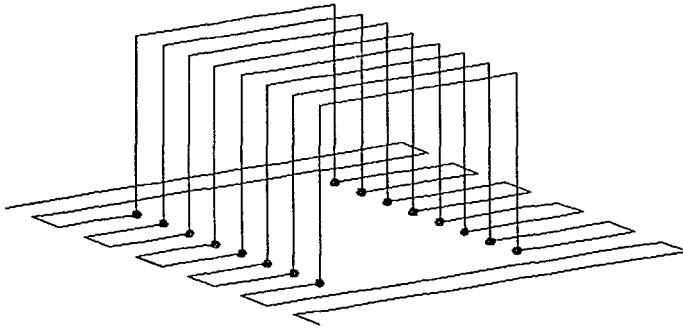


Fig. 5.15. Propuesta de marcado de regiones por medio de puntos para la extracción de objetos.

En las imágenes de las figuras 5.10-5.14 se puede apreciar que, con un método sencillo, como el de marcar fronteras de los objetos dentro de la imagen, se podrían extraer objetos como cadenas, despreciando las secuencias largas de ceros (dado un parámetro de umbral para el número máximo de ceros), que indicarían que solo se encuentran líneas continuas representando el fondo de la imagen. En la figura 5.16 se muestra la imagen de una célula tomada de una de las imágenes de mitosis y su respectiva representación como forma discreta en 3D. En la figura 5.17, se muestra la misma forma discreta, pero con el objeto aislado de la imagen. Lo que se busca es tener las subcadenas de la forma mostrada en ésta figura y con ellas definir una representación del objeto correspondiente extraído de la imagen.

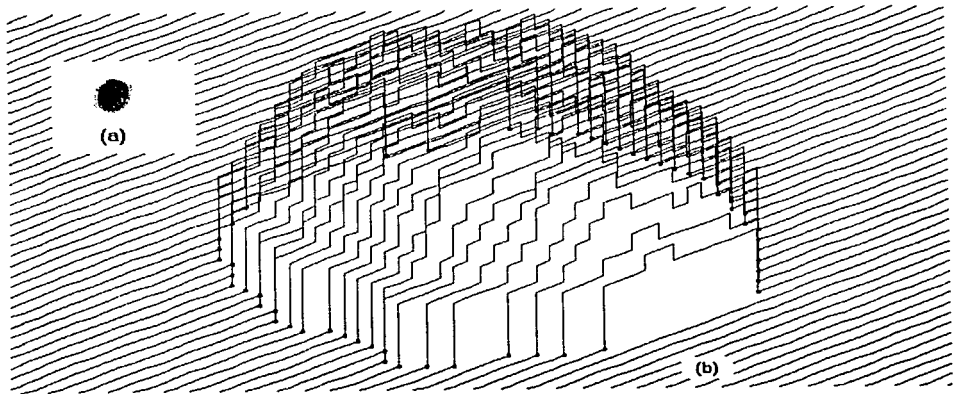


Fig. 5.16. (a) Imagen digital de una célula de mitosis. (b) Propuesta de marcado de objetos en la representación discreta de la cadena de la imagen mostrada en (a).

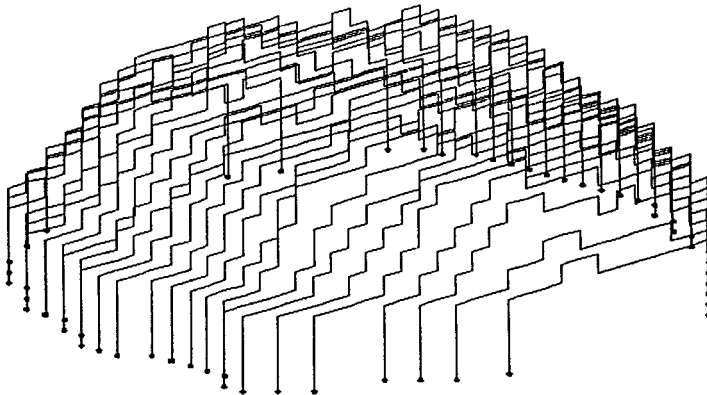


Fig. 5.17. El objeto extraído de la imagen en la figura 5.9.

5.2.2 Transformación de cadenas

Como se ha mencionado anteriormente, la búsqueda de objetos por medio de plantillas en imágenes es una parte muy importante en visión por computadora, pero en particular en el área de reconocimiento de patrones.

Para poder realizar la búsqueda de una plantilla dentro de una imagen, es necesario definir una medida de similitud, por ejemplo la que está basada en una búsqueda de invariantes así como de transformaciones de objetos (ver [32]).

Un procedimiento efectivo para medir la similitud entre dos objetos sería: obtener la cadena de los objetos y convertir un objeto en otro. Éste proceso se puede realizar mediante el uso de los elementos de cadena de cada objeto, convirtiendo los elementos de un objeto en los del otro. Cabe mencionar que las dos cadenas deben ser de igual longitud, para que esta función sea uno a uno.

Por ejemplo, para convertir de un elemento 2 a un elemento 3, solo se necesita un movimiento de 90° del último segmento; pero para convertir de un elemento 1 a un elemento 3, son necesarios dos movimientos. En la figura 5.18 se muestran ejemplos de estas conversiones realizando sólo un movimiento. En la figura 5.19 se muestran ejemplos de conversiones realizando dos movimientos. Como se puede observar, basta uno o dos movimientos para cambiar de un elemento a otro

El total de movimientos requerido para convertir una cadena en otra dará una medida de similitud entre las cadenas.

Entre menos movimientos se realicen para convertir de una cadena a otra, el grado de similitud es mayor entre los objetos.

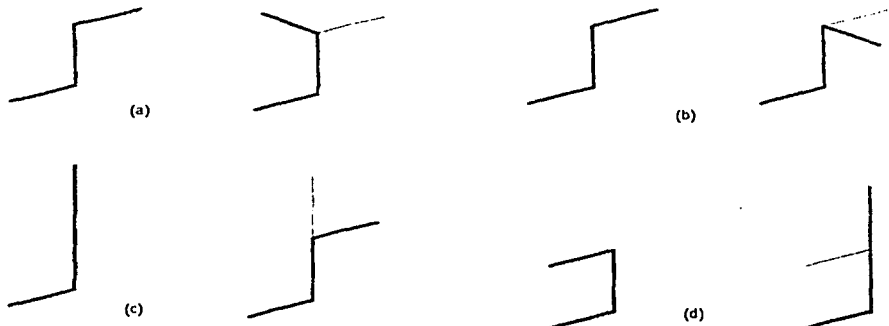


Fig. 5.18. Transformaciones de un elemento del código de cadena a otro con un solo movimiento de 90° : (a) conversión del elemento 2 al elemento 3; (b) conversión del elemento 2 al elemento 1; (c) conversión de elemento 0 al elemento 2; (d) conversión del elemento 4 al elemento 0.

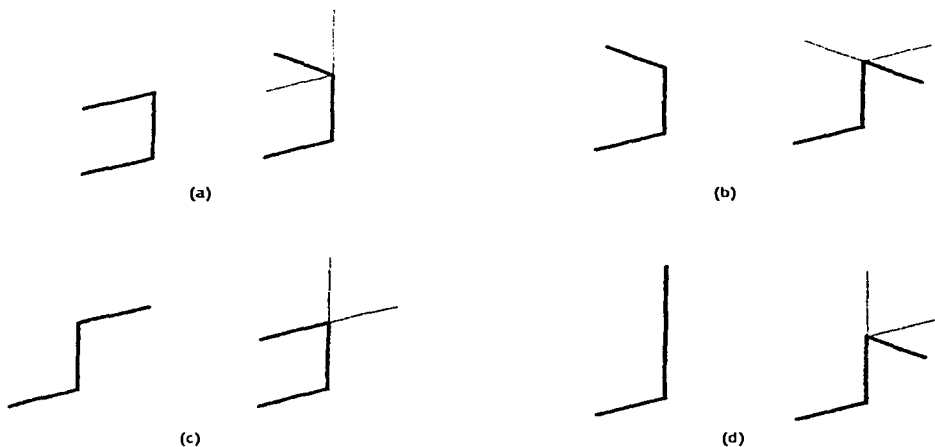


Fig. 5.19. Transformaciones de un elemento del código de cadena a otro con dos movimientos de 90° : (a) conversión del elemento 4 al elemento 3; (b) conversión del elemento 3 al elemento 1; (c) conversión de elemento 2 al elemento 4; (d) conversión del elemento 0 al elemento 1.

5.2.3 Identificación de rasgos

En el mundo actual la identificación de individuos es muy importante, es por eso que se han creado varios métodos como son: lectura del iris del ojo, asignación de un código a cada persona, etc. El método clásico de identificación de individuos ha sido la huella digital. En esta sección se propone la codificación (mediante el método propuesto en el capítulo anterior) de imágenes tomadas de huellas digitales.

Si por medio de un fuerte análisis de las cadenas de imágenes de una muestra de huellas digitales se pudiera asegurar que para cada una de ellas la cadena es única, ésta sería una forma de identificar a cada uno de los individuos. En las figuras 5.20 y 5.21 se muestran dos huellas digitales y sus respectivas cadenas, como se puede observar las cadenas son diferentes.

Como es conocido, todas las huellas digitales son diferentes si pertenecen a diferentes personas, por lo que es de esperarse que las cadenas de imágenes de huellas diferentes también sean diferentes.

Como se puede observar, en este capítulo se ha visto que:

1. Mediante el proceso de codificación, se pueden usar técnicas de compactación, lo cual permite la reducción del espacio en memoria requerido para almacenar imágenes.
2. La codificación de imágenes biomédicas contribuye al proceso de análisis en una de las investigaciones en curso del Instituto de Biotecnología.
3. Obteniendo la cadena de cualquier imagen dada, se puede realizar la búsqueda de plantillas dentro de la imagen.
4. La identificación de individuos mediante la codificación de la huella digital es posible.

El estudio de las formas en tres dimensiones es y seguirá siendo una parte muy importante en visión por computadora, es por esto que algunos trabajos como los propuestos aquí han empezado a realizarse.

CONCLUSIONES.

En esta tesis se ha mostrado la importancia de la representación de los objetos del mundo real y también cómo esta representación tiene un papel en diferentes áreas de investigación.

Fueron presentados varios métodos clásicos para la representación de formas, que han sido de importancia en el procesamiento de imágenes y visión por computadora.

Los antecedentes y el estado del arte del código de cadena que se aplicó han sido presentados para mostrar la relevancia que ha tenido el uso de éste y otros códigos. Así como también se presentaron las propiedades más útiles del código aplicado: independencia respecto a translación y rotación y normalización respecto al punto de inicio. Además se definió la transformación espejo de cadenas, el inverso de una cadena y se sugirió una manera de utilizar el código de cadena para la comparación de curvas.

Durante el desarrollo de esta tesis se realizó un sistema para la codificación en forma unidimensional de una imagen digital.

Mediante dicho sistema, el proceso de codificación comprende tres etapas:

1. Conversión de la imagen digital a una matriz bidimensional.
2. Binarización, consistente en la conversión de la matriz bidimensional en otra de tres dimensiones, binaria.
3. Obtención de la cadena de la imagen a partir de la matriz tridimensional.

Se presentaron ejemplos de formas discretas, mostrando las gráficas de las respectivas cadenas que representan a dichas formas.

También se elaboró un editor de cadenas, el cual permite obtener las subcadenas que se repiten dentro de una cadena.

Durante este trabajo se han estudiado las aplicaciones para las cuales serán útiles estos sistemas ya que teniendo una imagen representada por una cadena, se permite la aplicación de varios métodos, como son:

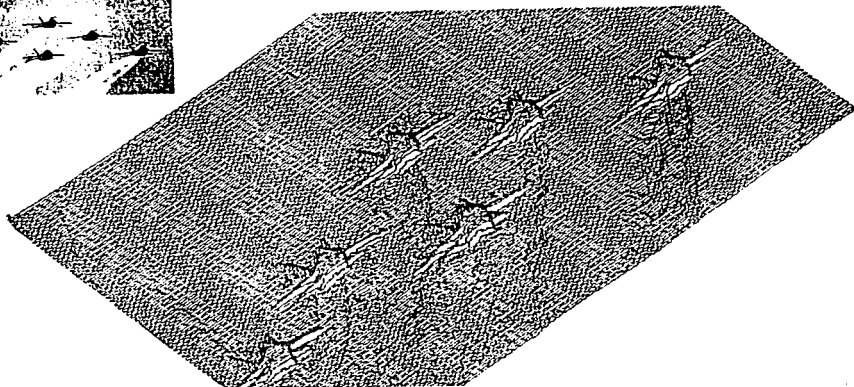
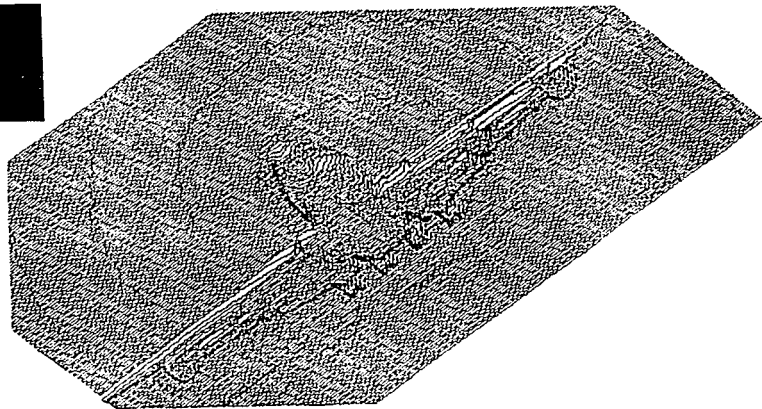
- Segmentación de imágenes
- Extracción de objetos
- Uso de plantillas
- Técnicas de compactación de información
- Uso de operaciones gramaticales. Ejemplos: si en una cadena se buscan tres "palabras", cada una de éstas palabras consistirá de una plantilla, o bien, si se buscan ciertos "picos" dentro de la cadena, cada uno de éstos consistirá de una subcadena que representará una plantilla.
- Búsqueda de textura, ejemplo: si una cadena tiene muchos ceros, quiere decir que esta formada por muchas regiones planas, en cambio si tiene muchos cambios de plano o muchos picos, esto querrá decir que se encuentran varias texturas dentro de la imagen.
- Cadenas planas: si una cadena tiene muchos elementos '2' consecutivos, quiere decir que se encuentra varias veces sobre un mismo plano; si la cadena contiene muchos elementos '0' consecutivos, querrá decir que la representación discreta tiene muchos lados rectos, pero si una cadena no contiene elementos '3' ni '1' entonces es plana.
- Búsqueda de identificación de objetos o imágenes que pueden ser similares, por medio de una cadena
- Técnicas de detección de esquinas en objetos.

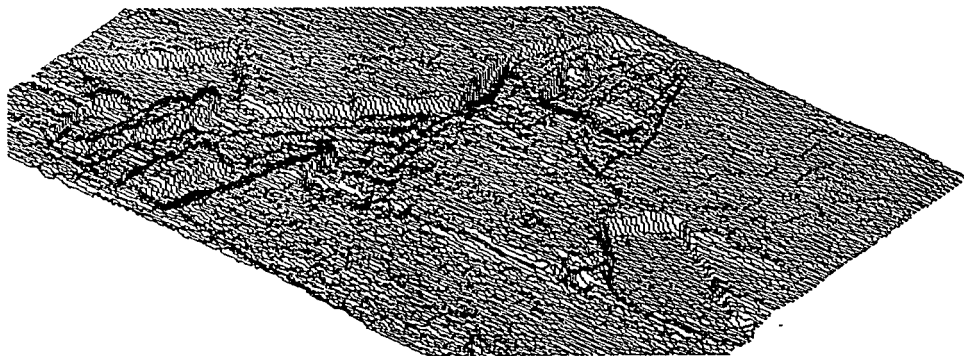
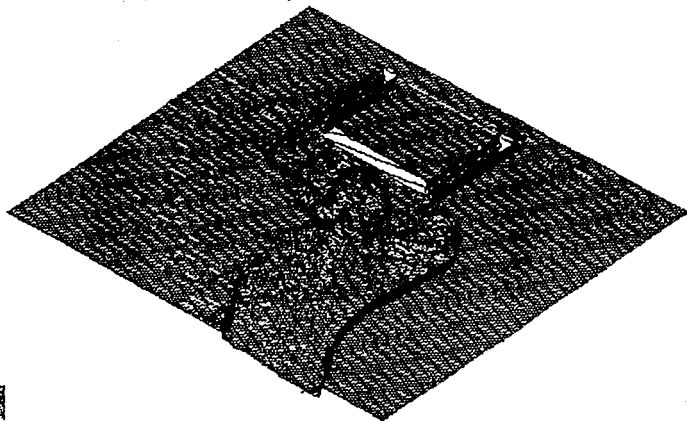
Se presentaron algunos ejemplos de este tipo de aplicaciones, así como también ejemplos de trabajos que podrían realizarse o que se han comenzado

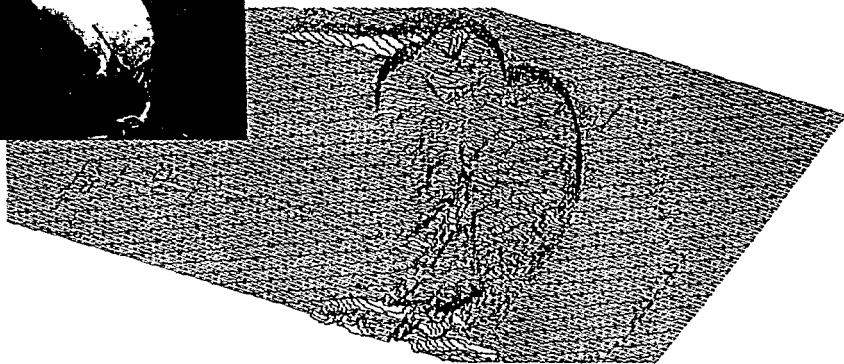
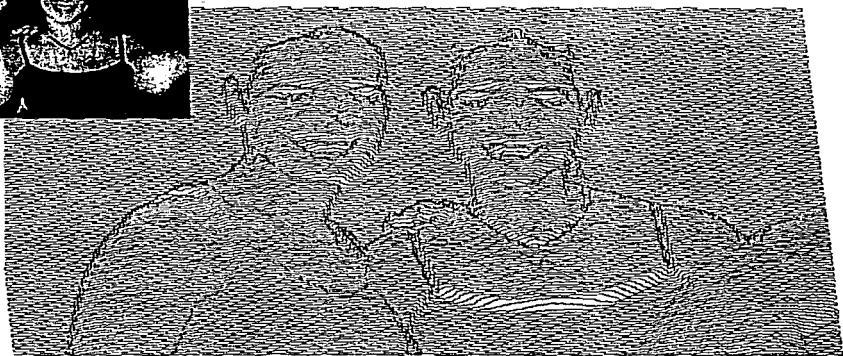
Cabe mencionar que la bibliografía presentada podría servir como base a un estudiante que desea iniciarse en el campo del procesamiento de imágenes así como también en el área de visión por computadora y en particular en el estudio de las formas.

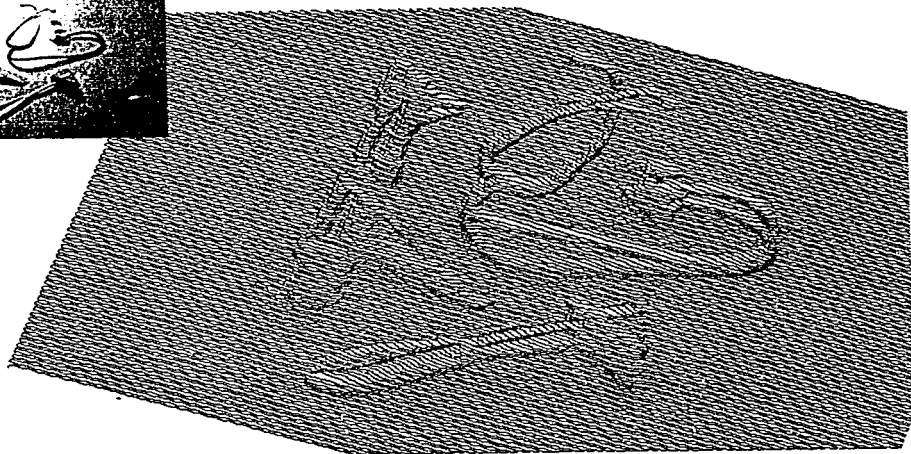
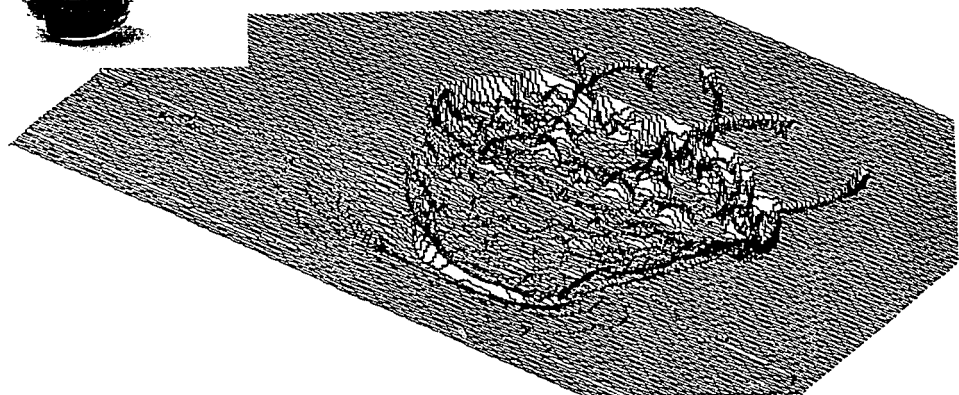
La representación unidimensional de imágenes digitales como aquí se ha hecho da una pauta para el desarrollo de nuevas investigaciones y sirve de apoyo para algunas otras que se están llevando a cabo en el Instituto de Investigación en Matemáticas Aplicadas y Sistemas (IIMAS), en el Instituto de Biotecnología (IBT) y en el Centro de Instrumentos de la UNAM.

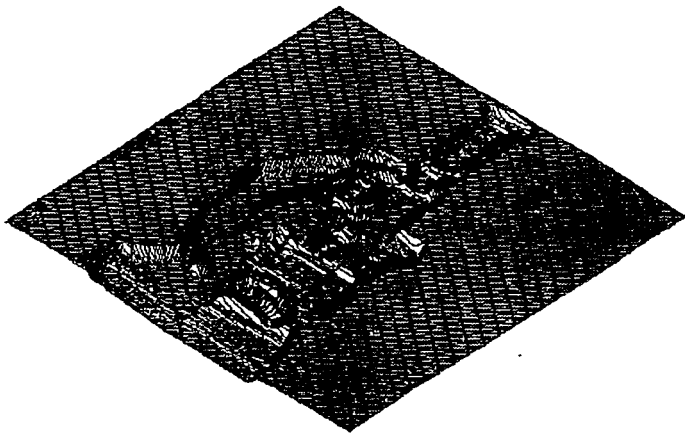
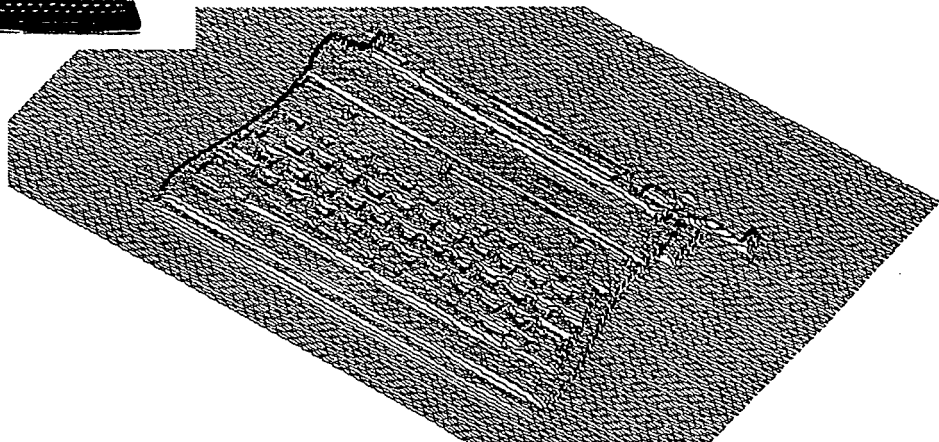
En las siguientes páginas se muestran ejemplos de imágenes digitales y su correspondiente cadena representada por su forma discreta usando el método presentado en esta tesis.

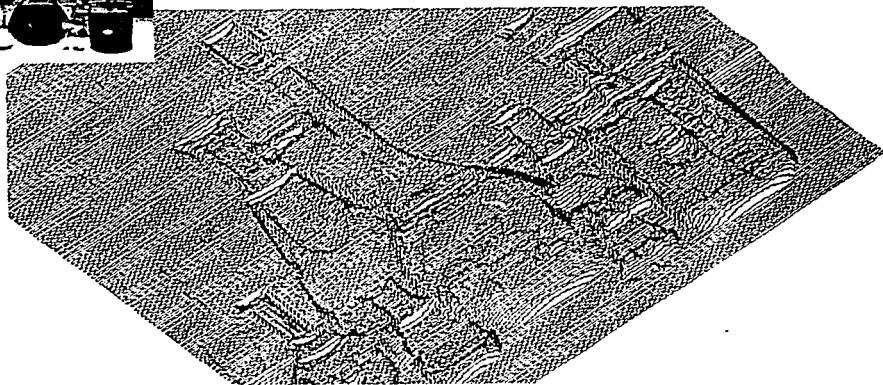
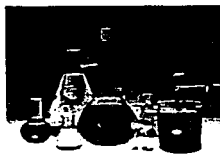
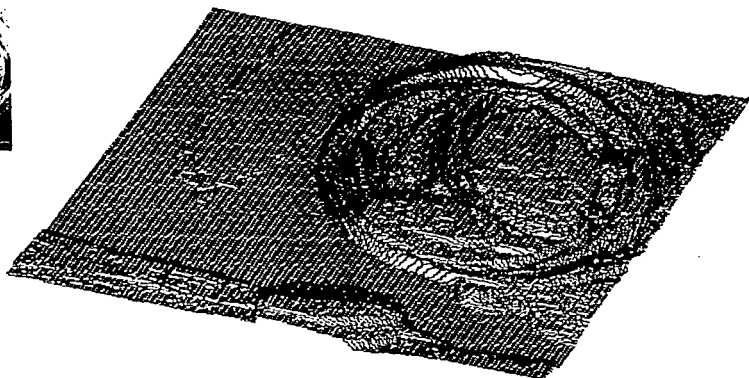


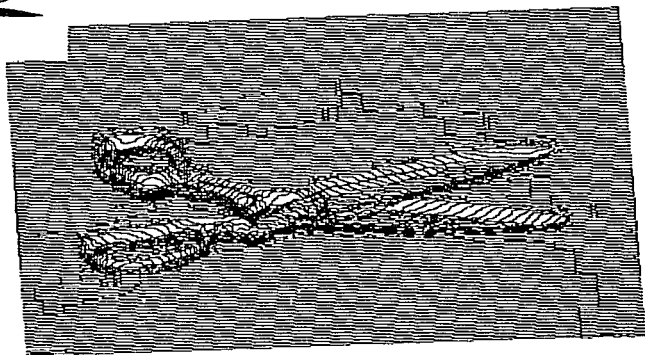
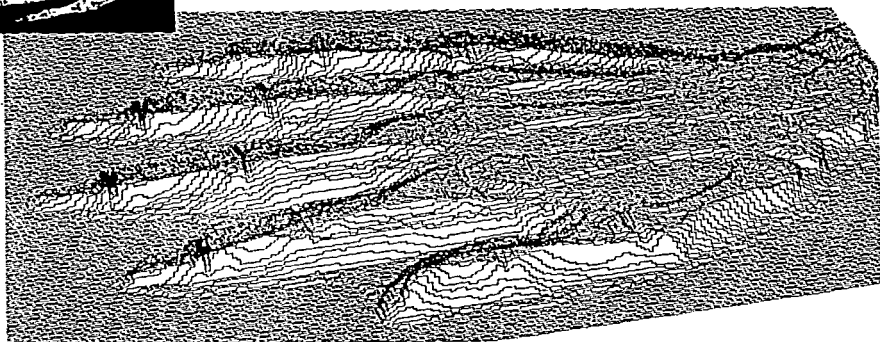


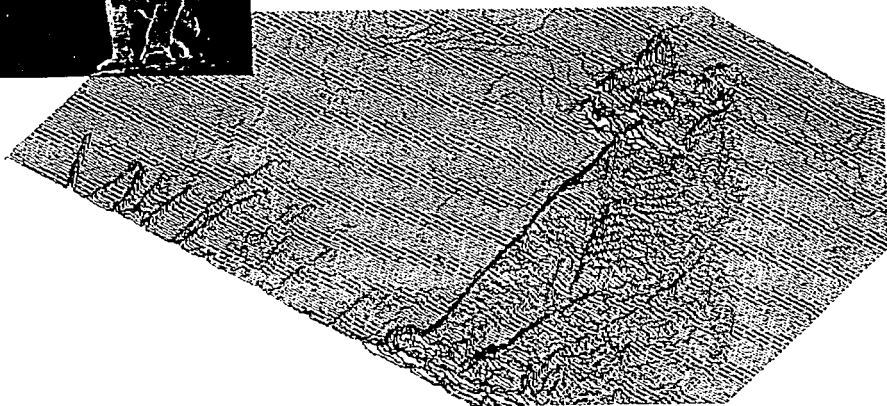
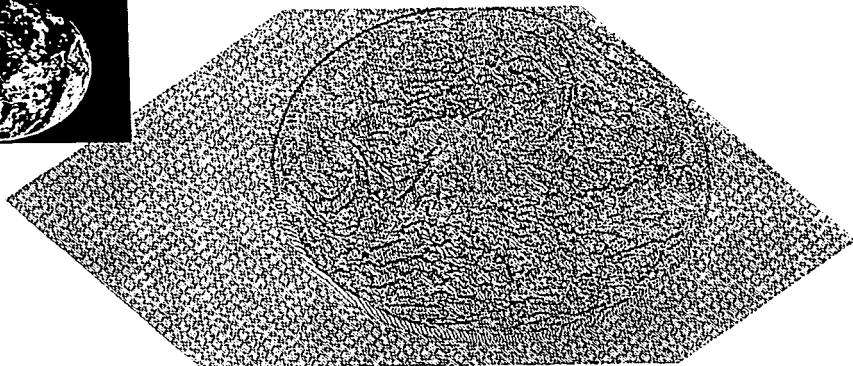












APENDICE

Un *arco* es una porción de una curva entre dos de sus puntos.

El *área* A de una región R está definida por $A = [\#R]*s$, donde s es el factor de escala que especifica el área del pixel.

El *borde o la frontera* de una región conectada R consiste de su contorno y el conjunto (posiblemente vacío) de fronteras simples, cada una de las cuales rodea todos los pixeles pertenecientes a algún hoyo de R .

Una *celda de resolución* es el componente de área más pequeño y elemental que tiene una intensidad de imagen asociada en una imagen digital. Una celda de resolución es referenciada por sus coordenadas espaciales, las cuales son las coordenadas centrales de su área. La celda de resolución constituye la unidad básica del proceso de bajo nivel de imágenes digitales de datos. Las celdas de resolución usualmente tienen áreas cuadradas, rectangulares o hexagonales.

Una *celda regular* es una figura geométrica cerrada, en donde cada uno de sus lados tiene la misma longitud, por ejemplo: el triángulo equilátero, un cuadrado, un pentágono, hexágono, etc.

Compactación es el proceso que se sigue para reducir del tamaño de archivos, ya sean éstos imágenes, videos, texto, etc.

El *contorno* de una región conectada R consiste de su frontera y el conjunto (posiblemente vacío) de fronteras simples que rodean pixeles de R .

Una **curva** es una configuración unidimensional, o una figura, obtenida al extender, alargar, contraer, doblar, etc., un segmento de línea o una línea ("curva" incluye línea recta o segmento de línea).

Una **curva algebraica** es aquella que corresponde a una ecuación algebraica.

Una **curva cerrada** es aquella cuyos puntos inicial y final coinciden.

Una **curva en el espacio** es una curva en el espacio tridimensional, tal como una curva en una esfera. Una curva en el espacio puede ser representada mediante dos ecuaciones simultáneas con variables de tres coordenadas x , y y z ; cada ecuación representa una superficie y el par representa la curva de intersección. También, una curva en el espacio puede ser descrita mediante tres ecuaciones, expresando x , y y z separadamente como funciones de un parámetro común.

Una **curva plana** es la que se encuentra en un plano y es representada frecuentemente como la gráfica de una ecuación $y=f(x)$, o bien $F(x,y)=0$, en coordenadas rectangulares x , y . Una manera para representar una curva en el plano es mediante ecuaciones paramétricas $x=g(t)$, $y=h(t)$.

Una **curva trascendental** es descrita por una ecuación que involucra funciones trascendentales, por ejemplo $y=\sin(x)$.

La **curvatura** de un círculo o un arco circular es el recíproco $1/r$ del radio r ; la curvatura mide la rotación angular de la línea tangente por unidad de longitud de arco (círculos pequeños tiene curvatura alta, y círculos grandes tiene curvatura baja). En general, la curvatura de una curva plana es un número definido en cada punto P sobre la curva y puede variar de punto a punto. La noción de curvatura puede ser extendida a curvas y superficies en el espacio. Un concepto importante es el de curvatura de una superficie en cualquier punto de ella. Por ejemplo, la curvatura de una esfera de radio r , es $1/r^2$ en cualquiera de sus puntos.

Detección de borde se refiere a cualquier proceso que determina una cadena de píxeles que separan una región de una imagen de una región vecina de la misma imagen.

Extracción de rasgos es el proceso mediante el cual un patrón de medida inicial o algún subconjunto de patrones de medida es transformado en un nuevo rasgo. Algunas veces a este proceso se le llama *propiedad de extracción*.

Una **figura** F o una **subimagen** F en una imagen continua o digital I es cualquier función F cuyo dominio es algún conjunto A del conjunto de coordenadas espaciales o celdas de resolución, cuyo rango es el conjunto G de intensidades de la imagen, y que es definida por $F(r, c) = I(r, c)$ para cualquier (r, c) perteneciente a A .

Una **forma** es una región sin tomar en cuenta su tamaño, posición y orientación en el plano. Esto es, dos regiones tienen la misma forma si se puede hacerlas coincidir exactamente realizando translaciones y rotaciones en el plano, además de un cambio de escala uniforme.

Una **frontera simple** es una curva cerrada orientada que no se toca o se cruza a sí misma. Los píxeles que están dentro de la frontera simple constituyen una región conectada que no tiene hoyos.

Un conjunto de píxeles H constituye un **hoyo** de una región R si H es un conjunto conectado de píxeles máximo que no pertenecen a R , pero están rodeados por R .

Una **imagen** es una representación espacial de un objeto, de una escena bidimensional, tridimensional o de otra imagen. Esta puede ser real o puede ser virtual, como en óptica. En visión por computadora, "imagen" usualmente significa *imagen grabada*, tal como, una imagen de video, una imagen digital, una fotografía o una pintura. Esta puede ser pensada como una función I de dos variables definida usualmente en una región rectangular cerrada de un plano. El valor de la imagen localizada en coordenadas espaciales (r, c) es denotado por $I(r, c)$. Para sensores ópticos o fotográficos $I(r, c)$ es

típicamente proporcional a la energía radiante recibida en la banda electromagnética para la cual el sensor o detector es sensible en un área pequeña alrededor de (r, c) . Para sensores de visor de rango, $I(r, c)$ da la distancia de (r, c) a un objeto en el espacio tridimensional. Para un sensor táctil, $I(r, c)$ es proporcional a la cantidad que el sensor detecta en el elemento de superficie alrededor de (r, c) . Cuando la imagen es un mapa, $I(r, c)$ es un índice o símbolo asociado con alguna categoría, tal como el color, uso de tierra, tipo de tierra, o tipo de piedra. Una imagen grabada puede estar en formato digital, fotográfico o de video.

Una **imagen binaria** es una imagen en la cual cada pixel toma el valor del 0 o de 1.

Una **imagen digital** es una imagen en un formato digital obtenido mediante la partición del área de la imagen en un arreglo finito de dos dimensiones de pequeñas, uniformemente formadas y mutuamente exclusivas regiones llamadas celdas de resolución, y asignando un valor de imagen representativo a cada región espacial. Una imagen digital puede ser pensada como una función cuyo dominio es el conjunto finito de dos dimensiones de celdas de resolución y cuyo rango es el conjunto de las posibles intensidades de la imagen.

Una **imagen en escala de gris** o **una imagen en nivel de gris** es una imagen en la cual cada pixel tiene un valor en un rango de 0 a N . Las imágenes en escala de grises típicamente tienen valores en el rango de 0 a 63, 0 a 255, o 0 a 1023, correspondientes a representación de los valores en 6, 8 o 10 bits.

Una **imagen espejo** es la que se obtiene al aplicar cierto operador a una imagen y el resultado será la imagen especular de la original.

Un **modelo de elevación digital** es una representación digital de una parte de la superficie de la tierra ([17]).

El **nivel de gris, intensidad de la imagen o valor de la imagen** es un número o valor asignado a una posición en la imagen. Para sensores ópticos o fotográficos, la intensidad de la imagen en (r, c) es proporcional a la salida integrada, reflexión, o transmisión de una pequeña área, usualmente llamada celda de resolución o pixel, centrado en la posición (r,c) . Su valor puede ser relacionado a la transmisión, reflexión, brillantez, densidad, etc.

Un **operador de imagen** es una función que toma una imagen como su entrada y produce una imagen como salida.

El **perímetro** de una región conectada R es la longitud del contorno de R .

Un **pixel** es una pareja cuyo primer miembro es una celda de resolución o posición espacial (renglón, columna) y cuyo segundo miembro es el valor de la intensidad de la imagen o el vector de los valores de la imagen asociados con la posición espacial.

Plantilla se refiere a un rasgo o patrón de un objeto.

Preprocesamiento es una operación aplicada antes de que la identificación de patrones sea llevada a cabo. El preprocesamiento produce, para las categorías de interés, rasgos de patrones que tienden a ser invariantes bajo translación, rotación, escalamiento, nivel de iluminación, y ruido. En esencia, el preprocesamiento convierte los patrones de medida a una forma que permite una simplificación en la regla de decisión.

Procesamiento de imágenes abarca todas las operaciones que pueden ser aplicadas a una imagen de datos. Estas incluyen, pero no están limitadas a, compresión de imágenes, restauración de imágenes, mejoramiento de imágenes, preprocesamiento, cuantificación, filtrado espacial, igualación y técnicas de reconocimiento.

Las técnicas de **reconocimiento de patrones** pueden ser usadas para construir reglas de decisión que permiten identificar unidades en base a sus patrones de medida. Un **patrón de medida o patrón** es la estructura de datos que resulta de hacer notar una

unidad. Las técnicas de reconocimiento de patrones pueden también ser empleadas para agrupar unidades teniendo patrones de medida suficientemente similares. En *reconocimiento estadístico de patrones*, los patrones de medida tienen la forma de N-tuplas o vectores. En *reconocimiento sintáctico de patrones*, los patrones de medida tienen la forma de sentencias del lenguaje de una gramática. En *reconocimiento estructural de patrones*, las medidas no tienen la forma de una N-tupla o un vector, sino que la unidad que está siendo medida es codificada en términos de sus partes, sus relaciones y propiedades.

El **rectángulo frontera** de una región R es el rectángulo más chico que circunscribe R , de lados paralelos a las direcciones renglón columna.

Una **región** de una imagen es un conjunto de celdas de resolución en el dominio espacial de la imagen.

Una **región R es conectada** si existe un camino entre dos celdas de resolución cualesquiera contenidas en R . De una manera más precisa, R es 4-conectada (8-conectada) si para cada par de celdas de resolución (r, c) y (u, v) pertenecientes a R existe una secuencia $\langle (a_1, b_1), (a_2, b_2), \dots, (a_m, b_m) \rangle$ de celdas de resolución pertenecientes a R tal que $(r, c) = (a_1, b_1)$, $(u, v) = (a_m, b_m)$; y (a_i, b_i) es 4-conectada (8-conectada) con (a_{i+1}, b_{i+1}) , $i = 1, 2, \dots, m-1$.

Una **región R es convexa** si para cada par de celdas de resolución en R , R contiene el segmento de línea recta digital que une el par de celdas de resolución.

La **representación de la frontera de un objeto tridimensional o volumen** es una representación que contiene cada una de las fronteras de la superficie del volumen. Cada frontera de la superficie es representada en términos de piezas simplemente descritas, cada una de las cuales tiene su propia frontera en forma de arco. Cada arco en sí mismo es representado en términos de piezas simplemente descritas que empiezan y terminan en puntos finales o vértices.

Resolución es un término genérico que describe qué tan bien un sistema, proceso, componente, material o imagen puede reproducir un objeto aislado consistente de líneas u objetos cercanos separado por un espacio. El *límite de resolución* o *resolución espacial* es descrito en términos de la más pequeña dimensión del objeto que puede ser discriminado u observado.

El concepto de **ruido** se le da a efectos no deseables introducidos en una imagen de muestra.

Segmentación de imágenes es un proceso que típicamente particiona el dominio espacial de una imagen en subconjuntos mutuamente exclusivos llamados regiones. Cada región es uniforme y homogénea con respecto a alguna propiedad, tal como el tono o la textura, y el valor de su propiedad difiere en alguna manera significativa de cada región vecina. Un proceso de segmentación de imágenes que usa la intensidad de la imagen como un valor de la propiedad produce regiones que son llamadas rasgos tonales discretos.

Un **segmento de línea recta digital** entre celdas de resolución (r_1, c_1) y (r_2, c_2) es el conjunto de todos los píxeles que se intersectan con el segmento de línea que une (r_1, c_1) y (r_2, c_2) .

Suavización de imagen se refiere a cualquier filtro espacial que produce una imagen de salida que espacialmente simplifica y aproxima la imagen de entrada. El proceso de suavizar una imagen suprime pequeños detalles de la imagen y realza estructuras de la imagen más grandes o más ásperas.

Una **superficie** es, informalmente, una porción continua de espacio de dos dimensiones, y se dice que es *cerrada* cuando delimita una porción de espacio finita en tres dimensiones a la que entonces se le llama sólido. El *área* de una superficie es una medida de su "extensión" en dos dimensiones; si es cerrada, el volumen es una medida de la extensión del sólido correspondiente.

La **textura** concierne a la distribución espacial de las intensidades de la imagen y sus rasgos tonales discretos. Cuando una pequeña área de la imagen tiene una variación muy baja de rasgos tonales discretos, la propiedad dominante de dicha área es el tono de gris. Cuando un área pequeña tiene una amplia variación de rasgos tonales discretos, la propiedad dominante del área es la textura. Tres cosas son cruciales en esta distinción: (1) el tamaño de pequeñas áreas, (2) los tamaños relativos de los rasgos tonales discretos, y (3) el número de rasgos tonales discretos distinguibles. La textura puede ser descrita en términos de dimensiones de uniformidad, densidad, aspereza, rugosidad, regularidad, intensidad y direccionalidad.

Visión por computadora es la combinación de las tecnologías de procesamiento de imágenes, reconocimiento de patrones e inteligencia artificial que se enfoca al análisis por computadora de una o más imágenes, tomadas con un sensor simple/multibanda o en una secuencia de tiempo. El análisis reconoce, proporciona y localiza la posición y orientación de una descripción o patrón simbólico y suficientemente detallado de los objetos en la imagen considerados de interés en el ambiente tridimensional. El proceso de visión por computadora frecuentemente usa modelado geométrico y representaciones de conocimiento complejo en modelado basado en "matching" o una metodología de búsqueda.

Un **voxel** es por un par ordenado cuyo primer componente es una localidad (renglón, columna, lado) correspondiente a un elemento de volumen, usualmente un paralelepípedo rectangular y cuyo segundo componente es el vector de propiedades de interés en dicho elemento.

Un par de pixeles vecinos es **4-conectado** si comparten un lado común. Un par de pixeles vecinos es **8-conectado** si comparten un lado común y una esquina común.

Nota: Definiciones tomadas de [25], [26] y [44].

REFERENCIAS

- [1] E. Bribiesca and A. Guzman, *How to describe pure form and how to measure differences in shapes using shape number*. Pattern Recognition, Vol. 12, pp. 101-112, 1980.
- [2] T. Pavlidis, *Algorithms for shape analysis of contours and waveforms*, Proc. 4th Int. Joint Conf. on Pattern Recognition, Kyoto, pp. 70-85, 1978.
- [3] R.F.A. Collard and H.F.J.M. Buffart, *Minimization of structural information: a set-theoretical approach*, Pattern Recognition, Vol. 16, pp. 231-242, 1983.
- [4] D.H. Ballard and C.M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [5] E. Persoon and K. S. Fu, *Shape discrimination using Fourier descriptors*, Proc. 2nd IJCPR, pp. 126-130, 1974.
- [6] H. Freeman, *On the encoding of arbitrary geometric configurations*, IRE Transactions on Electronic Computers. EC-10, pp. 260-268, 1961.
- [7] H. Freeman, *Techniques for the digital computer analysis of chain encoded arbitrary plane curves*, Proc. Natn. Electron. Conf. Vol. 18, pp. 312-324, 1961.
- [8] E. Bribiesca, *A geometric structure for two-dimensional shapes and three-dimensional surfaces*, Pattern Recognition, Vol. 25, No. 5, pp. 483-496, 1992.
- [9] E. Bribiesca, *Arithmetic operations among shapes using shape numbers*, Pattern Recognition, Vol. 12, No. 2, pp. 123-137. 1981.
- [10] E. Bribiesca, *A new chain code*, Pattern Recognition, Vol. 32, pp. 235-251, 1999.
- [11] E. Bribiesca, *Measuring 2D shape compactness using the contact perimeter*, Comp. Math. Appl. 33(11) (1997) 1-9.
- [12] Z. Kulpa, *More about areas and perimeters of quantized objects*, Comput. Vision Graphics Image Process., Vol. 22, pp. 268-276, 1983.
- [13] H. Freeman, *Computer processing of line-drawing images*, Comput. Surveys, Vol. 6, pp. 57-97, 1974.
- [14] Amnon Jonas and Nahum Kiryati, *Digital representation schemes for 3D curves*, Pattern Recognition, Vol. 30, No. 11, pp. 1803-1816, 1997.

- [15] E. Bribiesca, *Digital elevation model data analysis using the contact surface area*, Graphical Models and Image Processing, 60, 166-172, 1998.
- [16] G. Lohmann, *Volumetric Image Analysis*, Wiley Teubner, 1998.
- [17] D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice-Hall, 1982.
- [18] W. Karush, *Dictionary of Mathematics*, Webster's New World, 1989.
- [19] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision Vol. 1*, Addison-Wesley Publishing Company, 1993.
- [20] H. Freeman, *Computer processing of line drawing images*, ACM Comput. Surveys 6 (1974) 57-98.
- [21] S.L. Horowitz and T.P. Pavlidis, *Picture segmentation by a tree traversal algorithm*, J. ACM 23, 2, 368-388, 1976.
- [22] Y. Shirai, *Analyzing intensity arrays using knowledge about scenes*, In PCV, 1975.
- [23] I. Tomek, *Two algorithms for piecewise linear continuous approximation of functions of one variable*, IEEE Trans. Computers 23, 4, 445-448, 1974.
- [24] J. Sklansky and D.P. Kibler, *A theory of non-uniformly digitizing binary pictures*, IEEE Trans. SMC 6, 9, 637-647, 1976.
- [25] R.O. Duda and P.E. Hart, *Pattern Recognition and Scene Analysis*, New York:Wiley, 1973.
- [26] L.G. Roberts, *Machine perception of three-dimensional solids*, In Optical and Electro-optical Information Processing, J.P. Tippett et al. (Eds.). Cambridge, MA: MIT Press, 1965.
- [27] E. Persson and K.S. Fu, *Shape discrimination using Fourier descriptors*, Proc., 2nd IJ CPR, 126-130, 1974.
- [28] C. deBoor, *A practical Guide to Splines*, New York: Springer-Verlag, 1978.
- [29] W. L. Smallwood and E.R. Green, *Biologia*, Publicaciones Cultural, Primera edición, 1970.
- [30] **Diccionario Enciclopédico Quillet**, *Editorial Cumbre*, Treceava edición, 1985.
- [31] E. Bribiesca, *A chain code for representing 3D curves*, Pattern Recognition 33 (5), 755-765 MAY 2000.

[32] Hermilo Sánchez, *Optimización de una medida de semejanza para objetos tridimensionales a partir de invariantes y transformaciones*, Computación y Sistemas Vol. III, No. 4, 274-286, 2000.

[33] <http://www.albany.edu/~achm110/abzdna.html>