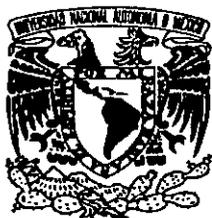


62



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

---

---

**FACULTAD DE INGENIERIA**

**RECONOCIMIENTO DE PATRONES EN IMAGENES  
CARACTERIZADAS EN LINEAS O BORDES**

**T E S I S   P R O F E S I O N A L  
QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
( AREA ELECTRICA ELECTRONICA )  
P R E S E N T A :  
JAIME HECTOR MURILLO QUINTERO**

**DIRECTOR DE TESIS  
M.I. LARRY H. ESCOBAR SALGUERO.**



284517

CIUDAD UNIVERSITARIA, MEXICO, D.F. , 2000



Universidad Nacional  
Autónoma de México



**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.

## *Dedicatoria*

*Dedico mi trabajo y esfuerzo a todos los que me tienen y han tenido, desde siempre, fe en mí, porque es lo que me impulsa a seguir adelante.*

## *Agradecimientos*

*A mi Madre, que guió mi mano ayudándome a completar mis primeros trazos en la escuela cuando yo era un niño.*

*A mis hermanas Alma, Amelia y Laura, por ser mi familia y haber estado en los tiempos buenos y malos que vivimos juntos.*

*A mi sobrina Ana Paola, quien es la más reciente y apreciada miembro en la familia y representa una nueva esperanza para todos.*

*A todos las personas que estuvieron cerca de mí cuando crecía y siempre mostraron un interés verdadero por ayudarme; familiares, maestros y amigos. Un agradecimiento muy especial a mi abuelo Ricardo, a quien extraño mucho.*

*A la Universidad y Facultad de Ingeniería, que me han dado mucho, una buena educación que va más allá de la formación académica.*

*Al Ingeniero Macías, por su amistad y por el apoyo que siempre me ha brindado desde que lo conozco. Además de darme la oportunidad de pertenecer a un buen lugar de trabajo.*

*A Larry, por su amistad, compañía diaria y por haber dedicado parte de su tiempo y esfuerzo a la dirección de esta tesis; por su ejemplo de trabajo y superación.*

*A todos los amigos del departamento de electrónica y compañeros de la división, a los que están hoy, agradezco su amistad que hace más disfrutable la rutina diaria; a los que se han ido por otros caminos en estos últimos años, les agradezco por los buenos momentos que tuvimos y por contar con su amistad por siempre.*

*A mi novia Belem, quien siempre tiene palabras de cariño y aliento para mí; quien en poco tiempo me ha dado muchas cosas que han llenado un espacio importante de mi vida. Te quiero mucho y gracias por estar conmigo.*

*A todos, Gracias.*

*Héctor*

# ◆ Índice General ◆

---

## CAPITULO I INTRODUCCION

---

1.1	INTRODUCCION.....	3
1.2	ESTADO DEL ARTE.....	5
1.3	CONTENIDO DE LOS CAPITULOS .....	6

---

## CAPITULO II IMAGEN DIGITAL

---

2.1	LA IMAGEN COMO MEDIO DE COMUNICACIÓN .....	11
2.2	VISION HUMANA .....	12
2.2.1	Origen de la información .....	13
2.2.2	Fuentes de energía .....	14
2.2.3	Estructura del ojo humano .....	15
2.2.4	Elementos de la percepción ocular .....	17
2.3	CONCEPTO DE IMAGEN ELECTRONICA .....	19
2.4	TECNOLOGIAS PARA EL TRATAMIENTO DE LA IMAGEN ELECTRONICA .....	19
2.4.1	La imagen analógica.....	19
2.4.2	La imagen digital .....	20
2.5	TECNICAS PARA LA REPRESENTACION DE LA IMAGEN DIGITAL .....	20
2.5.1	Bitmaps .....	20
2.5.1.1	Tipos de bitmaps .....	20
2.5.2	Imagen vectorial.....	21
2.5.3	Tecnología fractal .....	22
2.6	SISTEMAS DE CODIFICACION DEL COLOR .....	22
2.7	SISTEMAS DE CODIFICACION DE IMAGENES MONOCROMATICAS.....	23
2.8	FORMATOS DE ARCHIVOS GRAFICOS.....	24
2.9	CONSTRUCCION DE LA IMAGEN EN LA PANTALLA DE LA COMPUTADORA .....	24
2.9.1	Resolución del monitor .....	24

2.9.2	Resolución de la imagen.....	25
2.10	RELACIONES BASICAS ENTRE PÍXELES.....	25
2.10.1	Vecinos de un píxel .....	26
2.10.2	Conectividad .....	26
2.10.3	Relaciones de equivalencia .....	27
2.10.4	Medidas de Distancia.....	28
2.10.5	Operaciones Aritmético / Lógicas .....	29
	COMENTARIO AL CAPITULO .....	30

---

## CAPITULO III

### MEJORAMIENTO Y SEGMENTACION DE LA IMAGEN

---

3.1	METODOS DEL DOMINIO ESPACIAL.....	33
3.1.1	Métodos basados en los píxeles.....	34
3.1.2	Métodos basados en el histograma.....	37
3.1.3	Resta de imágenes .....	40
3.1.4	Mejora Local .....	40
3.1.5	Promedio de imágenes.....	41
3.2	FILTRADO ESPACIAL.....	42
3.2.1	Filtros suavizantes.....	44
3.2.2	Filtros de realce.....	46
3.2.3	Filtros de derivadas .....	48
3.3	MEJORA EN EL DOMINIO DE LA FRECUENCIA.....	51
3.4	METODOS EN EL DOMINIO DE LA FRECUENCIA.....	51
3.4.1	Filtrado paso Bajo .....	52
3.4.2	Filtrado paso Alto.....	54
3.5	SEGMENTACIÓN BASADA EN FRONTERAS.....	55
3.6	METODOS DE GRADIENTE .....	56
3.6.1	Operadores de Segunda Derivada .....	58
3.7	DETECTORES DE BORDES GAUSIANOS .....	62
3.8	TRANSFORMADA DE HOUGH .....	64
3.9	BUSQUEDA DE GRAFOS.....	66
3.10	UMBRALIZACION .....	68
	COMENTARIO AL CAPITULO .....	71

---

## CAPITULO IV

### REPRESENTACION Y DESCRIPCION

---

4.1	ESQUEMAS DE REPRESENTACION .....	75
4.1.1	Códigos de cadena .....	75
4.1.2	Curvas $\phi$ -s .....	77
4.1.3	Aproximaciones poligonales.....	77
4.1.4	Firma .....	79
4.1.5	Lados del contorno .....	79
4.1.6	El esqueleto de una región .....	80
4.2	DESCRIPTORES DE CONTORNO .....	80
4.2.1	Números de forma .....	80
4.2.2	Descriptores de Fourier .....	81
4.2.3	Momentos .....	84
4.2.4	El Histograma del código de cadena .....	87
4.2.5	El Histograma en parejas geométricas .....	88
4.2.6	Combinación de los descriptores simples. ....	89
4.3	DESCRIPTORES RELACIONALES .....	90
	COMENTARIO AL CAPITULO .....	93

---

## CAPITULO V

### RECONOCIMIENTO DE PATRONES

---

5.1	DESCRIPCION.....	97
5.1.1	Elementos del análisis de imágenes.....	99
5.1.2	Patrones y clases de patrones .....	99
5.1.3	Métodos de decisión teórica .....	101
5.2	METODOS SINTACTICOS.....	101
5.3	TEORIA DE LENGUAJES, GRAMATICAS Y AUTOMATAS.....	102
5.3.1	Expresiones regulares .....	107
5.3.2	Gramáticas.....	109
5.3.3	Función del análisis sintáctico.....	114
	5.3.3.1 Tipos de análisis sintáctico.....	116
5.4	RECONOCIMIENTO SINTACTICO DE CADENAS.....	120

5.5	AUTOMATAS PARA RECONOCIMIENTO .....	122
5.6	RECONOCIMIENTO SINTACTICO DE ARBOLES .....	124
5.7	APRENDIZAJE.....	127
	COMENTARIO AL CAPITULO .....	133

---

## CAPITULO VI

### IMPLEMENTACION DE EXPERIMENTOS Y EVALUACION DE RESULTADOS

---

6.1	EXPERIMENTO 1 .....	137
6.1.1	Definición de la estructura .....	137
6.1.2	Definición del código de freeman, y extracción de primitivas.....	138
6.1.3	Generación de las reglas que describen la secuencia de la estructura .....	142
6.1.4	Generación de una familia de Patrones .....	144
6.1.5	Diseño de un autómata finito para el reconocimiento de cadenas .....	145
6.1.6	Comentario al experimento.....	148
6.2	EXPERIMENTO 2 .....	149
6.2.1	Análisis de la escena .....	149
6.2.2	Mejora y Segmentación de la imagen.....	149
6.2.3	Descripción del objeto .....	150
6.2.4	Definición del árbol y generación de las reglas que lo describen .....	152
6.2.5	Comentario al experimento.....	157
	COMENTARIO AL CAPITULO .....	158

---

## CAPITULO VII

### CONCLUSIONES

---

	CONCLUSIONES.....	161
--	-------------------	-----

---

### ANEXO A (Nociones Básicas de Teoría de Conjuntos)

### BIBLIOGRAFIA

---

C · A · P · I · T · U · L · O

I

I N T R O D U C C I O N

**IV. Representación y Descripción.**

Se establecen los esquema adecuados para diferentes aplicaciones, ya sea para establecer sus características externas (su contorno) o para establecer sus características internas (los pixeles que comprenden la región), en seguida se hace descripción de la zona representada en la imagen por medio de características tales como la longitud, la orientación de una línea recta que une puntos externos, y el número de concavidades. Se exponen diversas técnicas para codificar la información extraída en el proceso de segmentación.

**V. Reconocimiento de Patrones.**

Estudia las imágenes digitales para desarrollar técnicas que hacen posible su reconocimiento, interpretación y lograr un método confiable que automatice este proceso. Este método de análisis debe de ser capaz de descubrir, identificar y comprender los patrones que son relevantes en el rendimiento de un trabajo basado en imágenes. Por la naturaleza de los patrones que se intenta clasificar se optó por exponer los métodos estructurales o sintácticos, se presenta la filosofía del método y la mecánica para su implementación.

**VI. Implementación de Experimentos y Evaluación de Resultados.**

Una vez establecido un método de reconocimiento se procede a diseñar experimentos que lo ejemplifiquen, se evalúa su confiabilidad, alcances y limitaciones.

**VII. Conclusiones.**

Se evalúan los resultados obtenidos con base a los objetivos iniciales y se presentan las conclusiones sobre el trabajo realizado.



## 1.1 INTRODUCCION

### RECONOCIMIENTO DE PATRONES EN IMAGENES CARACTERIZADAS EN LINEAS O BORDES

La percepción visual constituye una acción sumamente poderosa, las imágenes son tal vez, el medio de mayor penetración para el aprendizaje. La vista nos sitúa en el entorno donde participaremos con otros sentidos, las imágenes son una guía para nuestras acciones, son un mecanismo que agiliza nuestra memoria, un simple símbolo decodifica en nuestro cerebro una cadena de acciones a realizar, respondiendo a situaciones aprendidas y procesadas.

El hombre en su afán por resolver problemas ha recurrido a la imitación de su propia naturaleza y de su entorno como un método de aprendizaje, hoy en día la informática trata de ser un modelo de la mecánica del cerebro, en esta dinámica y conforme avanza el proceso informático, la percepción visual se ha convertido en un medio sumamente efectivo en la adquisición de datos y el manejo de información, acciones fundamentales para la toma de decisiones.

Los requerimientos técnicos para implementar estos ambientes son grandes, pero está comprobado que las personas prefieren ambientes visuales, cada vez más complejos y más interactivos para los usuarios. La tendencia actual se aboca a transportar la realidad tangible a una realidad virtual, creando imágenes que puedan tener una interpretación en el mundo digital, por ello la necesidad de implantar sistemas expertos que manejen gran cantidad de información gráfica en forma eficiente.

El *objetivo* de este trabajo de tesis es caracterizar una imagen en sus rasgos principales e introducirla en una base de imágenes patrón para su comparación y en este proceso llegar a su reconocimiento.

Esperando como resultado que al implementar un algoritmo de reconocimiento de patrones se pueda obtener un sistema que sea lo más confiable posible y susceptible a errores en el proceso de comparación, esperando que este método pueda generalizarse para formas similares, en este caso para patrones basados en líneas, en diferentes imágenes. Una buena estructuración nos garantizará el poder caracterizar rasgos eficientemente y obtener una identificación exitosa.

Los elementos de un sistema de propósito general; ver figura 1.1, capaz de realizar las operaciones de tratamiento digital de imágenes está conformado por las siguientes etapas fundamentales[1]:

- **Adquisición de imágenes.** Para la adquisición de imágenes se necesitan dos elementos. El primero es un dispositivo físico sensible a una determinada banda del espectro de energía electromagnética (como las bandas de rayos X, ultravioleta, visible o infrarrojo) y que produzca una señal eléctrica de salida proporcional al nivel de energía detectado. El segundo, denominado digitalizador, es un dispositivo diseñado para convertir la salida del sistema sensible a una forma digital.
- **Almacenamiento.** El almacenamiento digital para aplicaciones de PDI (Procesamiento Digital de Imágenes) cae en tres categorías básicas:
  1. Almacenamiento temporal a corto plazo, para ser empleado durante el procesamiento.
  2. Almacenamiento en línea, para una reutilización relativamente rápida, y
  3. Almacenamiento en archivo, caracterizado por un acceso poco frecuente. El tamaño de la información almacenada se mide en bytes (8bits). Un método para obtener almacenamiento a corto plazo consiste en emplear la memoria de una computadora, otro en tarjetas especializadas,

denominadas memorias temporales que almacenan una o más imágenes a las que puede accederse con rapidez. La cantidad de imágenes que puede manejar cada dispositivo dependerá de sus cualidades técnicas.

- **Procesamiento.** El tratamiento de imágenes digitales implica un procedimiento que normalmente se expresa en forma de algoritmos. Así, con la excepción de la adquisición de imágenes y su representación, la mayor parte de las funciones de tratamiento de la imagen pueden ser implementadas en software. La única razón de ser del hardware especializado en el procesamiento de imágenes es al necesidad de mayor velocidad en algunas aplicaciones para evitar limitaciones fundamentales de las computadoras. En la actualidad es posible obtener comercialmente una gran cantidad de software de tratamiento digital de imágenes. Cuando se combina con otro software de aplicación, como hojas de cálculo y gráficos, proporcionando un punto de partida excelente para la solución de problemas específicos de procesamiento de imágenes.
- **Comunicación.** La comunicación en el tratamiento digital de imágenes implica, principalmente, comunicaciones locales entre sistemas de procesamiento de imágenes y telecomunicaciones entre dos puntos, habitualmente para la transmisión de los datos de las imágenes técnicas como la compresión y descompresión de imágenes son fundamentales para la transmisión de imágenes a distancia.
- **Presentación.** Las señales de salida del módulo de visualización pueden emplearse como entrada de un módulo de grabación de imágenes que produzca una copia impresa (diapositivas, fotografías o transparencias) de la misma imagen que aparece en la pantalla del monitor. otros medios de visualización incluyen tubos de rayos catódicos (CRT) y dispositivos de impresión.

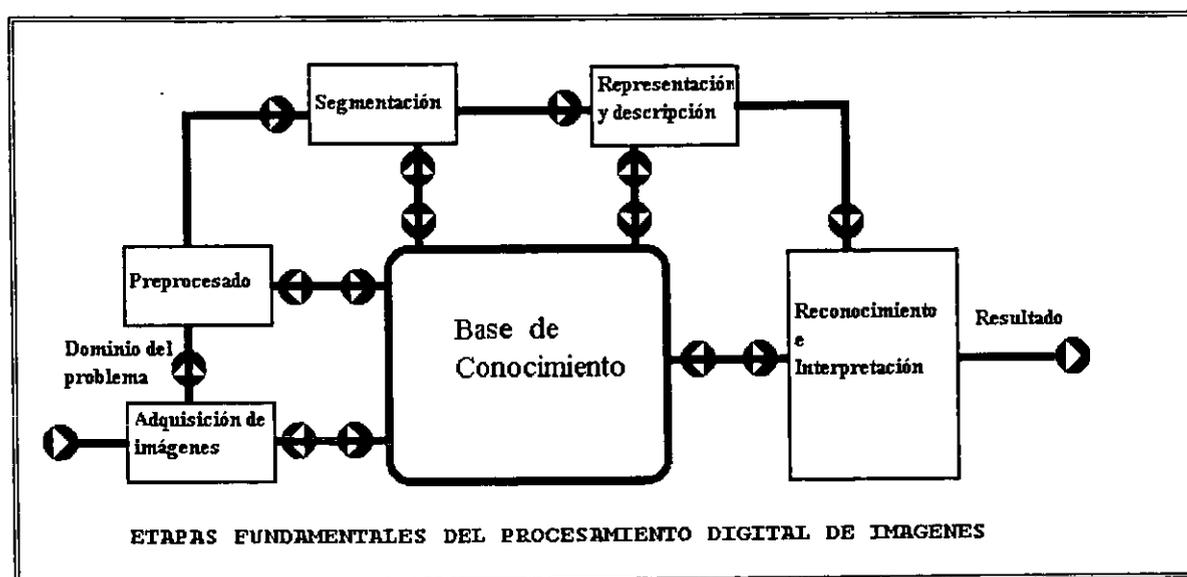


Figura 1.1 [B]

Con base en estos elementos se establece el siguiente procedimiento para llegar al objetivo de este trabajo. El primer paso consiste en crear un procedimiento para caracterizar las imágenes, estas imágenes tienen como principal característica que están formadas o constituidas por líneas o bordes, por ejemplo: huellas dactilares, firmas o símbolos sencillos. Una vez generada la base de imágenes patrones, se procede a generar el algoritmo de reconocimiento, basado en las características de la imagen. Al momento de tener la imagen como un archivo gráfico, esta imagen en primer lugar será caracterizada de acuerdo a la base de

conocimiento en que están formadas las imágenes patrón, después de introducir la imagen dato a la base de patrones, en un proceso comparativo el algoritmo de reconocimiento irá descartando posibilidades hasta encontrar el patrón coincidente. El criterio de decisión para aceptar o rechazar el patrón dependerá del peso que tenga la característica principal de la imagen o el conjunto de ellas, para automatizar el procedimiento se pueden establecer límites de porcentaje de igualdad entre las imágenes comparadas y una vez alcanzando este límite se aprobará la identificación, estableciendo un porcentaje de certeza.

## 1.2 ESTADO DEL ARTE

El procesamiento y análisis de imágenes digitales nace en el momento en que se dispone de recursos tecnológicos para captar y manipular grandes cantidades de información espacial en forma de matrices de valores. Esta distinción sitúa al procesamiento y análisis de imágenes digitales como una tecnología asociada a las ciencias de la computación y por tanto cabe pensar de ella como una proyección del término visión artificial dentro del ámbito de la inteligencia artificial. Una de las primeras técnicas de imágenes digitales surgió en los años veinte en la transmisión de imágenes de fotografías periodísticas a través de cable submarino entre Londres y New York. Esto supuso que el tiempo entre emisión y recepción de las imágenes pasaría de una semana (por barco) a tres horas (por cable). Técnica que era realmente muy precaria, en comparación a las tecnologías que se utilizadas en la actualidad.

El verdadero despegue del tratamiento digital de imágenes se tiene hasta los años 50's y 60's con la aparición de los primeros sistemas de computo, fomentado en parte por al carrera espacial y las investigaciones militares, las primeras imágenes que tuvieron un interés para ser sometidas a técnicas de interpretación digital fueron las obtenida por medio de satélites, todo esto en los EE.UU. Dentro del contexto europeo tan solo cabe destacar la investigación desarrollada, durante los años setenta, por los profesores G. Matheron y J. Serra de la École National Supérieure des Mines de París (Francia) con base en resultados de la geometría integral y el cálculo de probabilidades para la caracterización y medición de formas. Esta metodología se denomina Morfología Matemática y ha tenido sus principales áreas de aplicación en los campos de la geología, la biología y la medicina. En los años ochenta se puede observar el comienzo de cierta especialización y catalogación de las técnicas existentes en el PDI. Así, aparecen los conceptos de técnicas para el procesamiento de imágenes digitales como el conjunto de todas aquellas técnicas asociadas a la captura, codificación y representación de las imágenes que no introducen sobre las mismas ningún tipo de interpretación, y técnicas para el análisis de imágenes digitales, técnicas de visión por computadora o visión mediante robot, en otro sentido se refieren a aquellas técnicas que tratan de extraer la información presente en la imagen con el fin último de hacer una interpretación de las escena representada por dicha imagen.

Durante los años ochenta las técnicas de análisis de imágenes se desarrollan de forma vertiginosa como consecuencia de la gran cantidad de aplicaciones que aparecen y la madurez alcanzada en el diseño de arquitecturas de computadoras. Los desarrollos más teóricos han seguido en gran medida las pautas marcadas por Marr , siendo la línea marcada por el MIT (Massachusetts Institute of Technology) la que más influencia ha tenido. Las mayores contribuciones se han centrado en el desarrollo de algoritmos para la detección de características ( bordes, líneas, texturas) que ayudan a definir lo que Marr llamó el esbozo primitivo, así como en el desarrollo de técnicas globales de segmentación de una imagen en regiones. A este último respecto cabe destacar aquellas aproximaciones que introdujeron la información de contexto en los procesos de clasificación y segmentación.

Hoy en día aún es complicado extraer información de una imagen que ofrezca múltiples características, sin embargo, existen imágenes que son fácilmente caracterizables por las formas con que se constituyen, formas de rasgos simples y que ofrecen un buen panorama de posibilidades. En el procesamiento digital se han desarrollado técnicas eficientes que caracterizan la imagen y ofrecen alternativas para su reconocimiento. Algunas aplicaciones que se han implementado para una identificación personalizada de individuos, son: la identificación por medio de huellas dactilares, reconocimiento del iris del ojo, identificación a través de su firma e identificación en rasgos de escritura.

Este trabajo de tesis esta abocado a trabajar con imágenes simples que se caracterizaran en líneas o formas que pueden ser caracterizadas como una serie de cadenas, arboles y descriptores de secuencias de puntos y crear una estructura simple, para poder aplicar métodos estructurales en su reconocimiento, estos métodos tienen como fundamento algoritmos basados en áreas de inteligencia artificial.

### 1.3 CONTENIDO DE LOS CAPITULOS

#### I. Introducción.

Se hace un esbozo de la importancia que tienen las imágenes como un medio fundamental para transmitir información en un sistema de toma de decisiones. Se establece el objetivo y los resultados esperados del trabajo de tesis. Se comentan los elementos básicos que constituyen el procesamiento digital de imágenes: desde la adquisición de las imágenes generadas en la realidad hasta su interpretación en el mundo digital.

Se hace una descripción general del estado del arte que guarda el reconocimientos de patrones en imágenes digitales, fundamentalmente en los nuevos enfoques y tendencias que se están generando para diversas aplicaciones.

#### II. Imagen Digital.

Se hace una referencia histórica de la evolución en la percepción de las imagen a través del tiempo, desde los fundamentos de la visión humana hasta el enfoque tecnológico de la actualidad. En este capítulo se explica el contexto y el enfoque que en la actualidad tiene las imágenes digitales, desde su constitución a base de píxeles hasta su manipulación en diferentes niveles, se establecen las características principales de una imagen digital; los formatos que se emplean en el almacenamiento y procesamiento de imágenes digitales, se exponen los principales rasgos que caracterizan a las imágenes en el dominio digital, estableciendo los rasgos que se generan al momento de captar y transformar la imagen real a la imagen digital.

#### III. Mejoramiento y Segmentación de la Imagen.

Se estudian las técnicas que sirven para adecuar una imagen digital original, en una imagen que sirva para una aplicación específica, las aproximaciones se presentan en dos categorías básicas, el dominio espacial y en el dominio de la frecuencia. En seguida se establece en qué consiste el primer paso del análisis de imágenes, la segmentación, que adecua la información de la imagen para resaltar su estructura u objetos principales que se emplearan en alguna aplicación específica, basando la segmentación en tres principales categorías: discontinuidad, similaridad y umbralización.

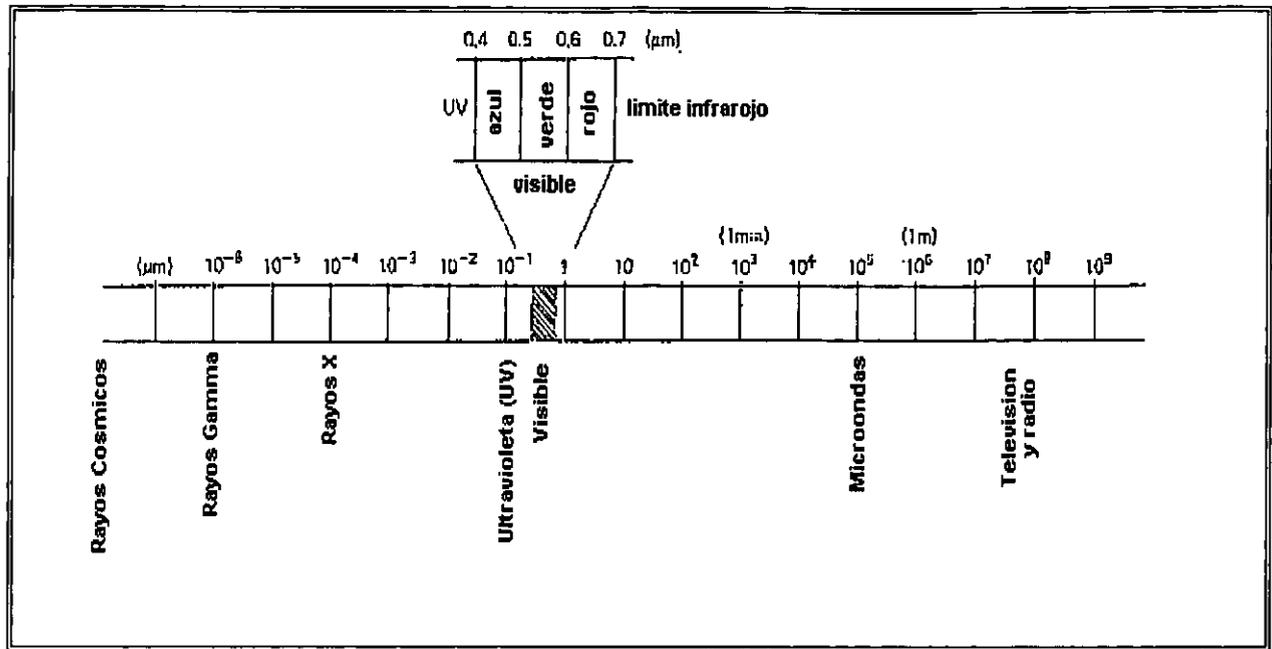


Figura 2.3 [C]

Podemos ver que la región asociada a la parte visible del espectro es una región muy pequeña del espectro que va de  $0.4 \mu\text{m}$  a  $0.7 \mu\text{m}$ , donde  $\mu\text{m}$  denota un micrómetro o  $10^{-6}$  metros. La región del azul se extiende de  $0.4\mu\text{m}$  a  $0.5\mu\text{m}$ , la del verde de  $0.5\mu\text{m}$  a  $0.6\mu\text{m}$  y la del rojo de  $0.6\mu\text{m}$  a  $0.7\mu\text{m}$  [3]. Por esto, la cantidad de información con la que se cuenta para poder caracterizar un fenómeno usando solamente su color es insuficiente y por tanto, es necesario tener observaciones más detalladas de su espectro para poder cuantificar la información de ese fenómeno.

### 2.2.3 Estructura del ojo humano

El ojo es casi una esfera de unos  $20[\text{mm}]$  de diámetro, formada por un conjunto de membranas denominadas córnea, esclerótica, corioide y retina. La córnea y la esclerótica constituyen las envolturas externas anterior y posterior del ojo respectivamente. La capa coroidal además de alimentar el ojo a través de sus vasos sanguíneos, tiene la misión de absorber las luces extrañas que entran al ojo, así como de amortiguar el efecto de dispersión de la luz dentro del globo ocular. El iris o diafragma está situado en la parte anterior del corioide, y tiene como misión controlar la cantidad de luz que entra en el ojo; ver figura 2.4. Para ello, la pupila o parte central del iris puede cambiar de tamaño en función de la luminosidad incidente desde  $2\text{mm}$  a  $8\text{mm}$  de diámetro. El lente del ojo está formado por capas concéntricas de células fibrosas y esta sujeta al corioide a través de fibras. La lente está compuesta principalmente por agua (60%-70%), 6% de grasa y proteínas. En el lente se absorbe cerca de un 8% del espectro de luz visible así como una gran porción de luz infrarroja y ultravioleta.

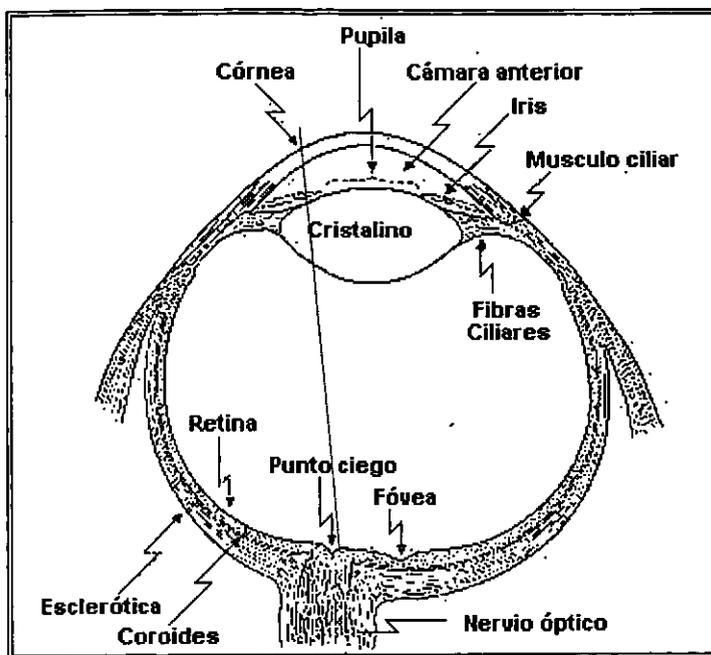


Figura 2.4 Diagrama simplificado de una sección transversal del ojo humano. [B]

La membrana más interna del ojo es la retina que cubre toda la pared interna del ojo. Cuando la luz llega al ojo, la imagen que transporta se forma en la retina por la sensibilización de dos clases de receptores: los bastones y los conos. El número de conos existentes en un ojo está entre 6 y 7 millones y su situación dentro del ojo se concentra alrededor de un punto llamado fóvea. La misión de los conos dentro del ojo es doble. Por un lado son responsables de la detección del color y por otro ayudan a resolver los detalles finos de la imagen. Cuando una persona quiere obtener detalles finos en una imagen, ésta se forma en su retina alrededor de la fóvea, consiguiendo por tanto, que los conos sean mayoritariamente los receptores de la luz. La visión a través de los conos se denomina visión fotópica o de luz brillante. [1]

Por otro lado, el número de bastones existentes en un ojo es muy superior al de conos y está entre 75 y 150 millones. Los bastones se distribuyen sobre toda la retina y al igual que los conos tienen una doble misión. Por un lado son responsables de dar una impresión general del campo de visión y por otro son responsables de la sensibilidad a niveles bajos de iluminación. Los bastones no son sensibles al color. Un objeto que a la luz del día tiene colores vivos, observado a la luz de la luna aparece sin colores, esto es debido a que sólo los bastones están estimulados. La visión a través de bastones se le denomina visión escotópica o de luz tenue. La figura 2.5 muestra la densidad de conos y bastones del ojo derecho.

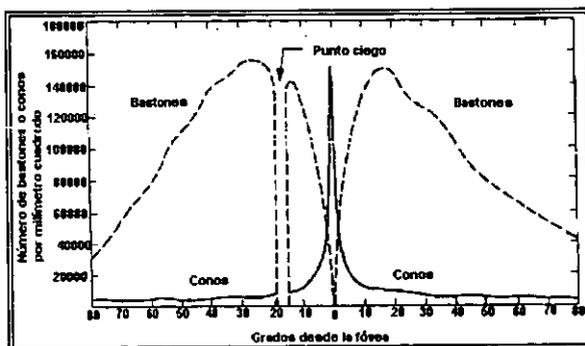


Figura 2.5 Distribución de bastones y conos en la retina. [B]

Puede verse como la densidad de conos toma su valor máximo alrededor de la fovea y en cambio los bastones tienen su valor máximo a  $20^\circ$  de la fovea y luego decrece. También puede observarse como existe una región ciega que coincide con la unión del nervio óptico con el ojo.

Con objeto de comparar el ojo con un sensor electrónico, consideremos el tamaño de la fovea, un círculo de aproximadamente unos  $1.5[\text{mm}]$  de diámetro, y la densidad de conos en esa región, aproximadamente unos  $150,000$  por  $[\text{mm}^2]$ . Puede por tanto, estimarse que el número de elementos receptores en la zona de mayor precisión del ojo puede llegar a ser de unos  $265,000$ . Si comparamos este número de receptores con los que la tecnología actual, pueden integrarse en un sensor CCD (dispositivos de carga acoplada) de unos  $7[\text{mm}^2]$ . Los sensores CCD son dispositivos electrónicos fotosensibles que poseen una estructura de células sensibles a la luz en forma de mosaico, cada una de esas células es lo que se denomina pixel, cada pixel es una estructura detectora que es capaz de almacenar fotones. Puede decirse que al menos en número de receptores la tecnología actual puede dar la misma resolución que el ojo humano.

#### 2.2.4 Elementos de la percepción ocular

El ojo humano es receptivo a la radiación electromagnética que denominamos luz visible y que notaremos como distribución espectral por  $L(\lambda)$ , siendo  $\lambda$  la longitud de onda. Al intervalo de  $\lambda$  valores que va de  $350[\text{nm}]$  a  $780[\text{nm}]$  (nanómetros) lo llamaremos luz visible. La distribución de la sensibilidad del ojo a las distintas longitudes de onda tiene forma de campana con un valor máximo para los conos en torno a los  $600[\text{nm}]$  y un máximo para los bastones en torno a los  $500[\text{nm}]$ . La radiación emitida por un objeto puede expresarse como:

$$I(\lambda) = \rho(\lambda)L(\lambda)$$

[Ecu. 2. 3]

donde  $0 < \rho(\lambda) < 1$  representa el coeficiente de reflexión o de transmisividad del objeto y la distribución de energía incidente. Ya que los valores de  $\rho(\lambda)$  además de depender de la longitud de onda, dependen de la materia o sustancia de la que se trate, las medidas de  $I(\lambda)$  nos permitirán discriminar entre distintas sustancias.

El ojo humano tiene la capacidad de adaptarse en un intervalo de valores de iluminación del orden de  $10^{10}$ . El punto más importante a la hora de interpretar este enorme intervalo es el hecho de que el ojo no opera de forma simultánea sobre todo el intervalo, si no que en cada caso y en función de la luminosidad media existente se hace sensible a un intervalo alrededor de dicho valor medio. Dicho intervalo de sensibilidad está definido de manera que el ojo considera como negro todos aquellos valores que están por debajo del límite inferior del intervalo. El límite superior no se satura como el inferior, pero si queremos que el ojo sea sensible a un determinado intervalo de luminosidad, habremos de impedir que el ojo reciba valores de intensidad luminosa muy superiores, ya que en ese caso trataría de adaptarse a los valores más intensos a costa de perder sensibilidad en los menos intensos. Este efecto de adaptación se denomina adaptación al brillo. Numerosos experimentos han mostrado que el brillo subjetivo percibido es una función logarítmica de la intensidad de luz incidente en el ojo. La figura 2.6 muestra una gráfica con la variación del brillo subjetivo frente al logaritmo de la intensidad medida en miliLambert [mL].

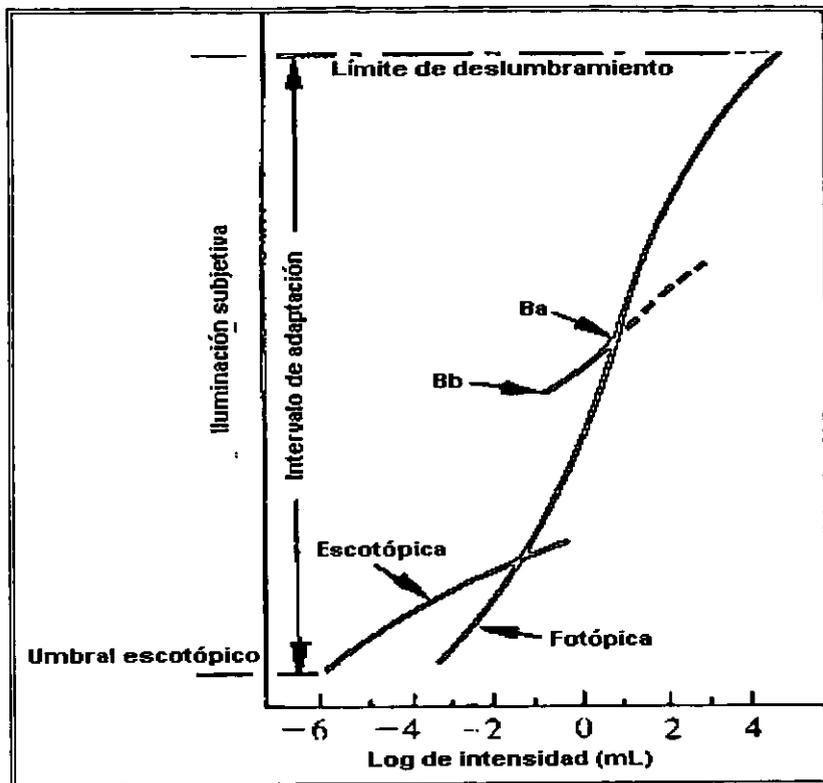


Figura 2. 6 Intervalo de sensaciones subjetivas de iluminación mostrando un nivel de adaptación particular [B]

Además puede observarse donde se sitúan tanto la visión fotópica como la escotópica así como la posible adaptación permitida para un brillo medio fijo  $B_x$ .

La habilidad del ojo humano para discriminar los distintos cambios de brillo en distintos niveles de luminosidad ha sido objeto de numerosos experimentos científicos. Un experimento clásico, en este sentido, es exponer a un observador frente a un panel plano iluminado de forma uniforme con intensidad  $I$ , y de tal manera que no perciba ninguna otra intensidad luminosa. Entonces en el centro de dicho panel vamos incrementando la luminosidad  $\delta I$ . El resultado de dicho experimento mostrará que mientras la variación de la luminosidad  $\delta I$  sea pequeña en comparación con  $I$ , el observador no notará variación alguna, pero cuando  $\delta I$  supera un determinado umbral entonces notará la diferencia. A la cantidad  $\delta I/I$  que hace que para un determinado valor de  $I$ , el 50% de las veces se perciba una variación denomina razón de Weber.

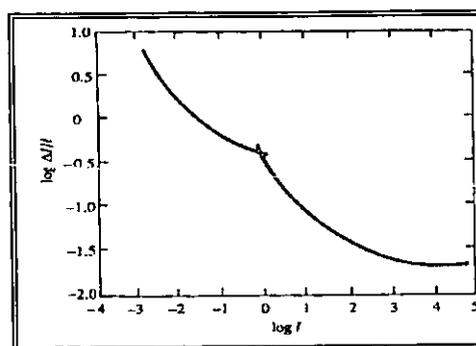


Figura 2.7 Cociente de Weber típico en función de la intensidad [B]

La figura 2.7 muestra la gráfica de  $\log(\delta I/I)$  frente a  $\log I$ . Esta curva que muestra la habilidad de discriminación del ojo es pobre para valores bajos de iluminación (razón de Weber grande) y se incrementa substancialmente cuando la iluminación del fondo aumenta (razón de Weber pequeña). Esta conducta está asociada a las propiedades de la visión a través de los bastones y conos respectivamente. De lo anterior se pone de manifiesto el interés de estudiar el brillo percibido en función de la iluminación.

En dos fenómenos experimentales se observa que esta función no es elemental, las bandas de Mach y el contraste simultáneo. Las bandas de Mach es el fenómeno por el cual si nosotros miramos una imagen de niveles de gris que está formada por una sucesión de escalones de intensidad decreciente, aunque constante en cada escalón, nuestra percepción de la misma es una imagen en donde el nivel de gris de un escalón varía de forma continua entre los valores de su escalón anterior y el posterior.

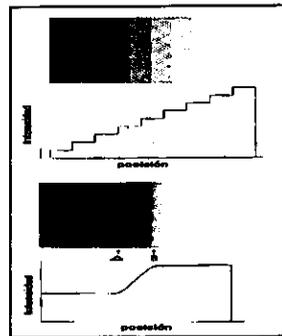


Figura 2. 8 Dos ejemplos que muestran que la iluminación percibida no es una simple función de la intensidad. [B]

## 2.3 CONCEPTO DE IMAGEN ELECTRONICA

Se denomina así a las imágenes capturadas, almacenadas y visualizadas a través de medios electrónicos. Este tipo de imágenes pueden ser "creaciones" o imágenes reales digitalizadas. Las primeras se realizan a través de técnicas de diseño por computadora. Las segundas son imágenes ya existentes que son transformadas a un formato comprensible por la computadora o dispositivo electrónico. El proceso de pasar una imagen real a un formato comprensible para el dispositivo se denomina captura.

Las imágenes electrónicas tienen que ser representadas de forma que el dispositivo eléctrico utilizado para su tratamiento pueda entenderlas y manejarlas, por lo que deben encontrarse en formatos digitales o analógicos.

Infografía es la técnica del grafismo por computadora. Los campos de aplicación de esta técnica son múltiples, podemos citar como ejemplo: teledetección, cartografía, imágenes médicas, bancos de imágenes periódicas, restauración documental, etc.

## 2.4 TECNOLOGIAS PARA EL TRATAMIENTO DE LA IMAGEN ELECTRONICA

### 2.4.1 La imagen analógica

Esta técnica consiste en representar la imagen a través de una onda senoidal. Las variaciones en la amplitud o en la frecuencia de esta onda determinan la información a partir de la cual se reconstruye la imagen. [6]

### 2.4.2 La imagen digital

La imagen digital, bien sea generada por la computadora o bien creada a través de algún instrumento de captura, tal como una cámara o un escáner, supone la traducción de los valores de luminosidad y color a un lenguaje que pueda entender la computadora y los periféricos con él relacionados, esto es, un lenguaje digital. La principal ventaja aportada por este lenguaje es la estabilidad: mientras que la emulsión de una imagen fotográfica clásica sufre una degradación química con el paso del tiempo, que repercute en la calidad de dicha reproducción, los ceros y unos que componen una imagen digital permanecen estables, con lo que la imagen no variará a lo largo del tiempo. Ahora bien, la calidad ofrecida por los procedimientos analógicos actuales siguen siendo superiores, en la mayoría de los casos, a la que se puede obtener con la tecnología actual en los equipos digitales accesibles al gran público.

El interés principal que puede suscitar la imagen digital, proviene de la posibilidad de construir y distribuir mensajes en los que la incorporación de imágenes puede enriquecer el contenido de la información.

## 2.5 TECNICAS PARA LA REPRESENTACION DE LA IMAGEN DIGITAL

Las imágenes pueden ser representadas de diferentes formas. En la actualidad contamos con tres técnicas : bitmaps, vectores, geometría fractal.

### 2.5.1 Bitmaps

Esta técnica consiste en representar las imágenes a través de múltiples puntos denominados píxeles. que al visualizarse todos juntos dan la sensación de mostrar una imagen continua. Un pixel es la información relativa a la posición y valor de luminosidad de un punto en la pantalla de la computadora. Una imagen digital consiste, en un conjunto discreto de posiciones y valores de luminosidad.

Un concepto importante a tener en cuenta es el de resolución. Tenemos dos formas de medirla: en números absolutos (tamaño), viene a ser el número de puntos de largo por el número de puntos de alto que definen una imagen, en número de píxeles por unidad de medida, pulgadas o milímetros. Lo normal es utilizar pulgadas (dpi= densidad de puntos por pulgada o ppp = puntos por pulgada). La segunda magnitud está relacionada con las posibilidades del dispositivo de salida (monitor o impresora), hablándose de baja resolución (200 dpi), media resolución (240-600 dpi) y alta resolución (por encima de 600 dpi). A mayor resolución mayor tamaño de archivos de la imagen pero también mayor calidad y mayores posibilidades de ampliación de la imagen con calidades aceptables.

#### 2.5.1.1 Tipos de bitmaps

- a) Imágenes bitonales (blanco y negro): se representan mediante una matriz de bits, los cuales sólo se pueden encontrar en dos estados, activados o desactivados. Su ventaja principal es el poco espacio de almacenamiento que requieren, siendo su mayor desventaja la imposibilidad de poder representar imágenes en color y la baja resolución que presentan. Estas imágenes pueden mejorarse a través de la técnica del halftoning. Esta técnica consiste en crear una ilusión de variedades de gris mediante la utilización de patrones binarios de blanco y negro.
- b) Paleta de colores o niveles de grises: se utiliza un byte para representar cada pixel, con lo que es posible representar hasta 256 distintas tonalidades de grises o colores. En el caso de imágenes en color, cada byte

representa un número que a modo de índice es utilizado para buscar el color real de ese pixel en una tabla, en la que están representados todos los colores disponibles para esa imagen. Esta tabla se denomina paleta de colores y cada imagen lleva asociada la suya propia.

- c) Color real: Son las de mayor calidad y las que más espacio ocupan. Cada pixel contiene información completa sobre su color, que se expresa como una combinación de diferentes intensidades de rojo, verde y azul. El número de colores posible depende del número de bits que se usen para representar cada pixel. Como las imágenes se representan con los tres colores básicos o primarios (verde, rojo y azul), el color de cada pixel está compuesto por la mezcla de estos tres, por lo que el número de bits de almacenamiento para un pixel en color real será generalmente múltiplo de tres. Así un color de 16 bits es realmente un color de 15 bits, teniendo uno sobrante al que se le da otro uso. Se suelen usar tres tipos de archivos de imagen de color real:
- c.1) Color real de 16 bits: Permiten 32 tonos distintos de cada color primario que combinados proporcionan 32,768 posibles colores.
  - c.2) Color real de 24 bits: Se suelen almacenar por planos de color. Cada plano contiene la imagen en un color: una de color rojo, otra de color verde y otra de color azul. La imagen final se obtiene como combinación de esos tres planos. Cada plano se representa mediante pixeles de 8 bits, lo que da 256 colores por plano, con un total de algo más de 16 millones de colores.
  - c.3) Color real de 32 bits: Es posible representar cada color primario mediante 10 bits lo que proporciona más de 1000 millones de colores. [6]

### 2.5.2 Imagen vectorial

Las imágenes se describen mediante objetos. Cada parte de la imagen es un objeto que puede ser tratado de forma independiente, pudiendo estar a su vez constituido por otros objetos que también son independientes. Es decir, cuando se define cada uno de los objetos que componen la imagen y se integra en ella no pierde sus características de objeto, pudiendo ser editado y manipulado en cualquier momento, sin que por ello los demás componentes del dibujo se vean modificados. Los objetos están definidos mediante vectores. Los archivos en los que se almacenan este tipo de imágenes están formados por las coordenadas de los vectores que las describen. Estos archivos en lugar de almacenar pixeles almacenan secuencias de comandos.

Un vector es una línea que se describe con la localización de los puntos de sus extremos. Un rectángulo podría definirse de la siguiente manera: RECT 0,0,200,200. Empleando coordenadas cartesianas, el programa dibujará ese rectángulo comenzando en el extremo superior izquierdo de la pantalla, alcanzando 200 pixeles a la derecha horizontalmente y 200 pixeles verticalmente hacia abajo.

El problema de estas imágenes es la lentitud a la hora de redibujar una imagen ya definida, acentuada mientras más compleja sea ésta, ya que se tienen que ir haciendo cálculos para cada objeto y posteriormente dibujarla. Para resolver en parte este problema algunos programas de tratamiento de este tipo de imágenes mezclan esta técnica con la de los bitmaps, tratando a los pixeles como objetos.[6]

Hay programas que permiten la conversión de las imágenes vectoriales en bitmaps y viceversa. El primer caso es posible a través de cualquier programa de captura de pantallas. El segundo caso es más complicado, sin embargo algunos programas son capaces de calcular los bordes de una imagen de mapa de bits, o las formas de las áreas coloreadas en la imagen, e inferir un polígono que la describa. Este procedimiento es conocido como autotracing.

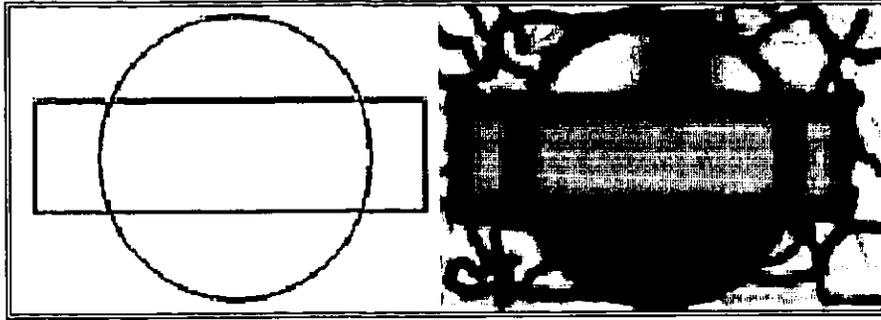


Figura 2. 9 A la izquierda: una imagen vectorial, a la derecha una imagen con formato bitmap [C]

### 2.5.3 Tecnología fractal

Esta tecnología está basada en la representación y diseño de imágenes a partir de ecuaciones matemáticas complejas. La palabra fractal tiene sus orígenes en el vocablo latino fractus, que significa irregular o roto. Este término fue introducido en la década de los 70 por el matemático Benoit B. Mandelbrot [6]. Un fractal es un objeto geométrico que posee autosemejanzas, es decir, las partes más pequeñas del dibujo son altamente similares a las partes más grandes. Un ejemplo típico es el de una costa. En este ejemplo cualquier nivel de detalle que tomemos, observamos siempre similitudes con el nivel de detalle interior.

La geometría fractal se aplica a múltiples campos de la ciencia y de la técnica (matemáticas, química, física, biología, medicina, astronomía, economía, etc.), últimamente se viene aplicando al mundo de la infografía donde ya están saliendo al mercado aplicaciones comerciales que utilizan esta técnica para el tratamiento y compresión de imágenes. Un campo donde está teniendo gran éxito es el de la representación de formas de la naturaleza, especialmente en la creación de paisajes artificiales de planetas imaginarios en películas de ciencia-ficción.

Por ejemplo, para dibujar una montaña se suele seguir el siguiente proceso iterativo: se toma una figura geométrica (un triángulo), se calcula el punto medio de cada arista y los puntos resultantes se desplazan un factor aleatorio para unirse después entre sí, obteniendo nuevos triángulos irregulares, este proceso repetido el número de veces necesario da como resultado una imagen muy real de una montaña.

## 2.6 SISTEMAS DE CODIFICACION DEL COLOR

La percepción del color es un acto subjetivo realizado en el cerebro humano tras procesar la información que le llega de ambas retinas. Los dos modelos matemáticos más usados para representar el color en las imágenes generadas por aparatos construidos por el hombre son:

- RGB. Es un sistema aditivo de colores generados por una fuente de luz que se emplea en las pantallas de TV. Los colores básicos son rojo, verde y azul (RGB = Red, Green y Blue), el resto de los colores se obtiene por combinación de los básicos; la suma de luces de color rojo y verde da el amarillo.
- CMYK. Es un sistema sustractivo, el mismo que empleábamos en la escuela cuando pintábamos con lápices de colores. Los distintos pigmentos aplicados a un papel blanco hacen de filtro impidiendo que éste pueda reflejar al color complementario. El término CMYK viene por las iniciales de los colores

básicos de este sistema: Cian (azul cielo) Magenta (color carmín de los pintores) Yellow (amarillo) y Negro (Black, la K en vez de la B es para no confundirlo con Blue). Este es el sistema de color que utilizan las impresoras de color y las imprentas. Existen otros sistemas de color como el Indexado, el HSB o el LAB pero su uso está restringido a ambientes muy profesionales. [4]

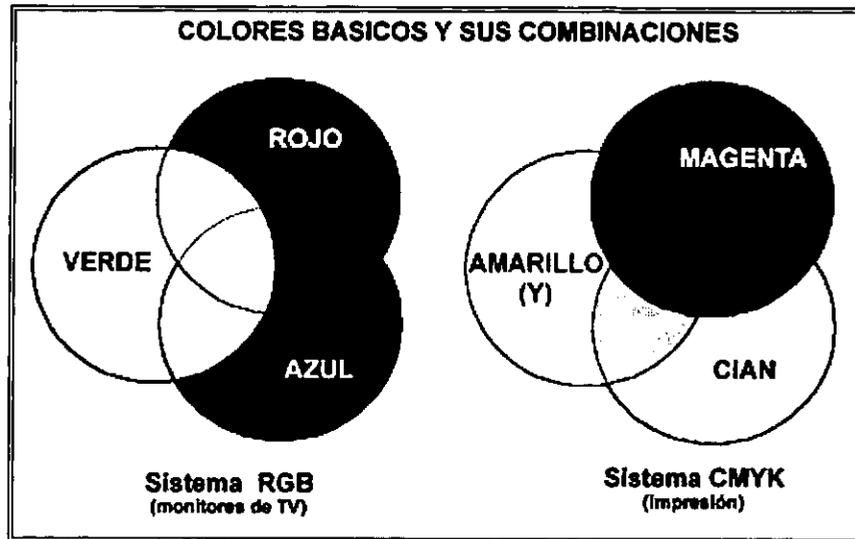


Figura 2. 10 [E]

## 2.7 SISTEMAS DE CODIFICACION DE IMAGENES MONOCROMATICAS

- a) Monocromo de mapa de bits (Líneas): Es el equivalente digital a los dibujos realizados con lápiz o bolígrafo: solo hay dos tonos posibles; blanco y negro. Son adecuados para digitalizar textos y gráficos generados por impresoras. Ocupan muy poco espacio y a la hora de imprimir podemos aprovechar al máximo la resolución de la impresora. Las fotografías pierden mucha calidad al pasarlas a este formato.
- b) Monocromo de escala de grises. Es el equivalente a los dibujos de sombreados realizados con lápiz o carboncillo. Se utiliza una gradación de 256 niveles de gris. Se usa, lógicamente, para representar fotografías en blanco y negro.



Figura 2. 11 A la izquierda: imagen a base de dos tonos: blanco y negro, a la derecha: la misma en tonos de gris. [E]

## 2.8 FORMATOS DE ARCHIVOS GRAFICOS

Una vez generadas las imágenes se almacenan en archivos con formatos especialmente diseñados para el almacenamiento de la información que permite reconstruir la imagen digital. Los más usuales hoy día son: BMP, PCX, TIF, GIF, TARGA, EPS, JPEG, PCPAINT, RAW, PICT Y PICT2.

BMP	Bit Map (mapa de bits), este tipo de archivos se utilizan como formato gráfico estándar en Microsoft Windows.
PCX	Archivos creados por Zsoft Corporation para ser utilizados en programas de diseño gráfico como Paintbrush, y PhotoFinish. Es uno de los formatos más estandarizados y puede ser utilizado prácticamente por todos los programas de diseño gráfico del mercado para PC's. Es el formato más utilizado para imágenes de 256 colores.
TIF	Iniciales de Tagged Interchange File Format, diseñado para ser un formato universal de imagen. Se usa tanto en PC's como en Macintosh. Es el formato más utilizado por usuarios de ordenadores, sobre todo en publicidad y en el mundo de la autoedición.
GIF	Creado por Compuserve como formato de imagen independiente de la plataforma en que se utilice. Es el formato más utilizado para escanear o digitalizar imágenes de 8 bits.
JPG	Diseñado por un grupo de expertos en fotografía denominado Joint Photographics Expert Group para poder trabajar en fotografía digital, este formato sufre ciertas pérdidas de calidad al comprimirse.

[E]

## 2.9 CONSTRUCCION DE LA IMAGEN EN LA PANTALLA DE LA COMPUTADORA

Podríamos considerar la pantalla de la computadora como un panel formado por celdas cuadradas. Cada una de esas celdas recibirá una determinada estimulación eléctrica de las partículas de fósforo que la componen que hará que presente un color u otro.

### 2.9.1 Resolución del monitor

A grandes rasgos la resolución del monitor viene determinada por dos variables: El tamaño de la "celda", al que se denomina "pitch dot" y que en los monitores actuales es, habitualmente, de 0.28[mm]. Dado que la imagen que vemos está compuesta por puntos adyacentes de diferentes colores, cuanto más pequeños sean dichos puntos obtendremos una mayor sensación de continuidad.

El número de puntos que es capaz de enviar a la pantalla la tarjeta gráfica de la computadora. Este parámetro depende de la cantidad de memoria de la que disponga la tarjeta para hacer los cálculos necesarios. Actualmente la resolución más habitual es de 800 puntos horizontales por 600 puntos verticales, mientras que en los equipos precedentes se situaba en (640x480) puntos. La variación del número de puntos influirá en el tamaño relativo de las imágenes dentro de la pantalla, ya que al ser fijo, la imagen ocupará un fragmento menor o mayor de la superficie de la misma según la cantidad total de puntos que se hayan representado en ella. [4]

### 2.9.2 Resolución de la imagen

Es un concepto relacionado con el anterior y recurriremos a un ejemplo práctico para explicarlo. Cuando captamos una imagen con la cámara y la imprimimos comprobamos que tiene unas dimensiones físicas de una imagen a una resolución de 72 puntos por pulgada.

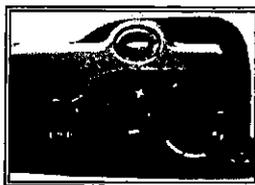


Figura 2.12 Imagen a 72 ppp [E]

Si a esta imagen le aumentamos la resolución a 150 ppp conseguiremos varias cosas:

- Su tamaño en pantalla se habrá multiplicado por cuatro al haber aumentado el número al doble el número de puntos necesarios para representar cada pulgada de anchura y de altura. (Recordemos que el tamaño del punto es una característica física del monitor).
- El programa de editor de imágenes habrá tenido que generar los puntos necesarios para llegar a la nueva resolución. Para ello habrá realizado una interpolación que consiste en el cálculo de cual sería el valor más probable para intercalar entre dos puntos que en la imagen original eran adyacentes. Lo más probable es que esa interpolación haya reducido la calidad de la imagen ya que habrá sido errónea en muchas ocasiones.
- Habremos aumentado el tamaño del archivo que guarda la imagen, ya que ahora tiene que almacenar mucha más información. Sin embargo, si imprimimos la nueva imagen observaremos que su tamaño en el papel sigue siendo idéntico al de la imagen original.

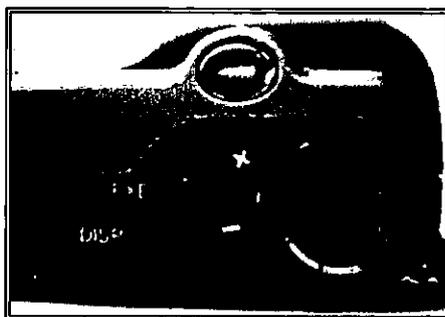


Figura 2.13 La misma imagen con una resolución de 150 ppp [E]

## 2.10 RELACIONES BASICAS ENTRE PÍXELES

Entre los píxeles existen varias relaciones simples aunque importantes de una imagen digital, una imagen se nota mediante  $f(x,y)$ , para píxeles particulares usaremos letras minúsculas como  $p$  y  $q$ .

Un subconjunto de píxeles de  $f(x,y)$  se nota  $S$ .

### 2.10.1 Vecinos de un pixel.

Un pixel  $p$  de coordenadas tiene cuatro vecinos horizontales y verticales cuyas coordenadas son:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1).$$

[Ecu. 2.4]

Este conjunto de pixeles, que recibe el nombre de 4-vecinos de  $p$ , se denota  $N_4(p)$ . Cada pixel está a distancia unitaria de  $(x, y)$ .

Los pixeles diagonales vecinos de  $p$  tienen coordenadas

$$(x+1, y+1), (x-1, y-1), (x-1, y+1), (x+1, y-1).$$

[Ecu. 2.5]

y se denotan  $N_D(p)$ . Estos puntos, junto con los cuatro vecinos [Ecuación 2.4], se llaman 8-vecinos de  $p$  y se notan  $N_8(p)$ .

### 2.10.2 Conectividad

La conectividad entre pixeles es un concepto importante usado para establecer las fronteras de objetos y las regiones componentes de una imagen. Para establecer si dos pixeles están conectados hemos de establecer si son adyacentes en algún sentido (por ejemplo si son 4-vecinos y si sus niveles de gris cumplen algún criterio de similitud, por ejemplo ser iguales). Así o en una imagen binaria con valores 0 y 1 dos pixeles pueden ser 4-vecinos y no estar conectados salvo que tengan el mismo valor. [1]

Sea  $V$  el conjunto de valores de niveles de gris que se usa para definir la conectividad; por ejemplo en las imágenes binarias,  $V = \{1\}$  o intervalo como  $V = \{33, 34, \dots, 50\}$  en las imágenes de niveles de gris.

Consideramos tres tipos de conectividad:

- A) 4-conectividad. Dos pixeles  $p$  y  $q$  con valores en  $V$  se dicen 4-conectados si  $q$  pertenece a  $N_4(p)$ .
- B) 8-conectividad. Dos pixeles  $p$  y  $q$  con valores en  $V$  se dicen 8-conectados si  $q$  pertenece a  $N_8(p)$ .
- C)  $m$ -conectividad. Dos pixeles  $p$  y  $q$  con valores en  $V$  se dicen  $m$ -conectados si:
  - a)  $q \in N_4(p)$  o
  - b)  $q \in N_D(p)$  y además el conjunto  $N_4(p) \cap N_4(q)$  es vacío. (Este es el conjunto de pixeles que son 4-vecinos de  $p$  y  $q$  cuyos valores están en  $V$ ).

Es importante notar que la  $m$ -conectividad se introduce para eliminar la ambigüedad en los posibles caminos que unen dos pixeles. Diremos que un pixel  $p$  es adyacente a  $q$  si están conectados. Es obvio que este concepto depende del tipo de conectividad que se use. Dos subconjuntos de una imagen  $S$ ,  $S_1$  y  $S_2$ , diremos que son adyacentes si algún pixel en  $S_1$  es adyacente a alguno en  $S_2$ . Un camino del pixel  $p$  con coordenadas  $(x, y)$  a  $q(x, y)$  con coordenadas es una sucesión distinta de pixeles con coordenadas:

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

[Ecu. 2. 6]

donde:  $(x_0, y_0) = (x, y)$  y  $(x_n, y_n) = (s, t)$ , siendo  $(x_i, y_i)$  adyacente a  $(x_{i-1}, y_{i-1})$

Para  $1 \leq i \leq n$ , siendo entonces  $n$  la longitud del camino. Es obvio que el tipo de camino depende del tipo de adyacencia utilizado.

Si  $p$  y  $q$  son pixeles en un subconjunto  $S$  de una imagen, entonces  $p$  está conectado a  $q$  en  $S$  si existe un camino de  $p$  a  $q$  que esté contenido en  $S$ .

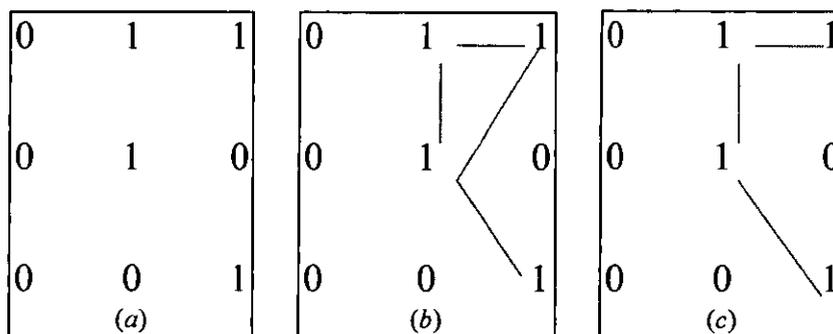


Figura 2. 14 (a) Distribución de pixeles; (b) 8-vecinos del pixel central; (c)  $m$ -vecinos del mismo pixel. Las líneas a trazos son caminos entre ese pixel. [B]

### 2.10.3 Relaciones de equivalencia

Una relación binaria  $R$  es un conjunto  $A$  de pares de elementos de  $A$ . Si el par  $(a, b)$  pertenece a  $R$ , diremos que "a está relacionado con b" y normalmente se presentará como  $aRb$ , tómesese, por ejemplo, el conjunto de puntos  $A = \{p_1, p_2, p_3, p_4\}$  ordenados de la forma:

$$\begin{array}{c} p_1 \quad p_2 \\ p_3 \\ p_4 \end{array}$$

Definiendo la relación de 4-conexión. En este caso  $R$  es un conjunto de pares de puntos de  $A$  que están 4-conectados; es decir,  $R = \{(p_1, p_2), (p_2, p_1), (p_1, p_3), (p_3, p_1)\}$ . Así,  $p_1$  está relacionado con  $p_2$ , con  $p_3$  y viceversa, pero  $p_4$  no está relacionado con ningún otro punto mediante la relación de 4-conexión.

Se dice que la relación binaria  $R$  sobre un conjunto  $A$  es:

- reflexiva si para cada  $a$  de  $A$ ,  $aRa$ ;
- simétrica si para cada  $a$  y  $b$  de  $A$ ,  $aRb$  implica  $bRa$ ; y
- transitiva si para  $a$ ,  $b$  y  $c$  de  $A$ ,  $aRb$  y  $bRc$  implica  $aRc$

Una relación que satisfaga estas tres propiedades se denomina relación de equivalencia. Una propiedad importante de las relaciones de equivalencia es que si  $R$  es una relación de equivalencia sobre un conjunto  $A$ , entonces  $A$  puede ser dividido en  $k$  subconjuntos disjuntos, denominados clases de equivalencias, donde  $k$

puede valer entre 1 e infinito, inclusive, de forma que  $aRb$  si y sólo si  $a$  y  $b$  están en el mismo subconjunto. [1]

### 2.10.4 Medidas de Distancia

Para  $p, q$  y  $z$  con coordenadas respectivamente.  $D$  es una función de distancia o métrica si cumple

- A)  $D(p, q) \geq 0$  ( $D(p, q)$  si y sólo si  $p = q$ )
- B)  $D(p, q) = D(q, p)$  y
- C)  $D(p, z) \leq D(p, q) + D(q, z)$

La distancia euclídea [1], entre  $p$  y  $q$  se define como:

$$D_c(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$$

[Ecu. 2.7]

Para esta medida, los pixeles que están a una distancia menor o igual que algún valor  $r$  ( $x, y$ ) son los puntos contenidos en un círculo de radio  $r$  con centro en  $(x, y)$ .

La distancia  $D_4$  (también llamada de bloques por la estructura que genera) se define mediante:

$$D_4(p, q) = |x-s| + |y-t|$$

[Ecu. 2.8]

Por ejemplo, los pixeles a distancia menor o igual que 2 del punto central forman contornos de distancia constante:

$$\begin{array}{ccccc} & & 2 & & \\ & & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ & & 2 & 1 & 2 \\ & & 2 & & \end{array}$$

La distancia  $D_8$  (también llamada de ajedrez entre  $p$  y  $q$ ) se define mediante:

$$D_8(p, q) = \max(|x-s|, |y-t|)$$

[Ecu. 2.9]

por ejemplo los pixeles con distancia  $D_8$  menor o igual que 2 son:

2 2 2 2 2  
 2 1 1 1 2  
 2 1 0 1 2  
 2 1 1 1 2  
 2 2 2 2 2

### 2.10.5 Operaciones Aritmético / Lógicas

Las operaciones aritmético/lógicas entre pixeles se usan muy frecuentemente en el procesamiento de imágenes [1]. Las operaciones aritméticas entre dos pixeles  $p$  y  $q$  se notan como sigue

- ◇ Suma:  $p + q$ ,
- ◇ Diferencia:  $p - q$ ,
- ◇ Multiplicación:  $pq$  o  $p \cdot q$
- ◇ División:  $p \div q$

Las principales operaciones lógicas entre pixeles son

- Y:  $p \text{ AND } q$  o también,  $p \cdot q$
- O:  $p \text{ OR } q$  o también,  $p + q$
- COMPLEMENTO: NOT  $q$  o también  $\bar{q}$

Es importante notar que las operaciones aritméticas se aplican a imágenes multivaluadas, es decir, a imágenes cuyos pixeles pueden tomar diferentes valores, mientras que las lógicas se refieren a imágenes bivaluadas tomando los valores 0 o 1.

Además de las operaciones aritmético/lógicas, es posible ampliar estos conceptos, a operaciones de máscara en la que se tenga en cuenta los valores de los pixeles vecinos, tanto para imágenes multivaluadas como binarias. En las figuras 2.15, se dan ejemplos de estas operaciones.

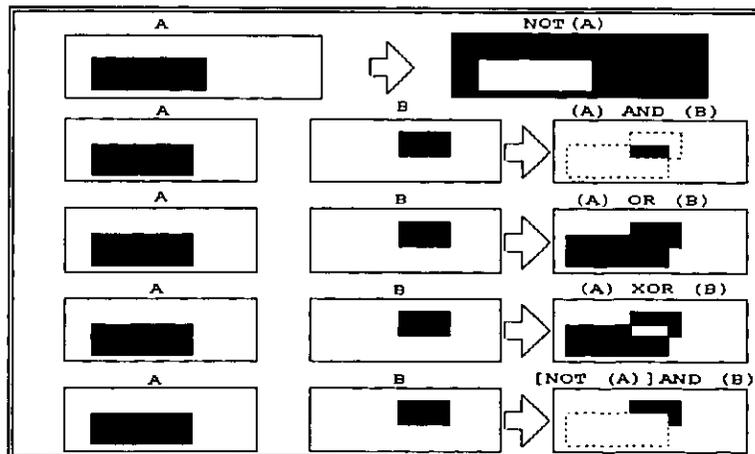


Figura 2. 15 Ejemplos de operaciones Lógicas sobre imágenes binarias. [B]

*II COMENTARIO AL CAPITULO II*

Las posibilidades para representar imágenes en el mundo digital parecen infinitas, existe una gran cantidad de dispositivos que captura de la realidad una imagen de alta calidad en su formato digital, existen un gran numero de formatos y su manipulación solo esta limitada por los recursos del *software* con que se cuente, en una imagen se puede manipular el color, la forma de la escena, la definición de la imagen, etc. El tamaño de información que la representa puede ser comprimida para que ocupe un menor espacio en los dispositivos de almacenamiento, esto escogiendo el formato más adecuado. Desde este punto ya es posible manipular las características de la imagen para adecuarla a una aplicación deseada, no siempre es conveniente trabajar con una gran cantidad de información que aporta la imagen al ser convertida a su formato digital, el trabajar con imágenes en color implica ocupar gran cantidad de memoria de máquina para representarla, para nuestros propósitos será necesario reducir lo más posible el rango de información, tratando de llevar los objetos de la imagen a una definición clara de los umbrales que definen sus límites y contornos, y así poder extraer solo esos contornos en procesos consecuentes.

En el siguiente capítulo se aboca a estudiar técnicas más específicas de manipulación de la información de la imagen, que tienen como fin eliminar la información que no es conveniente para alguna aplicación, dejando solo la una representación concreta de los objetos que interesan en la imagen.

· C · A · P · I · T · U · L · O ·

II

I M A G E N D I G I T A L

La percepción visual siempre ha llamado la atención del hombre, en la historia de la humanidad se han generado un sin número de expresiones visuales que intentan utilizar la fuerza de las imágenes para expresarse, en este capítulo se hace un pequeño esbozo de las distintas filosofías que se han formulado para describir la información que proyecta una imagen y como es llevada a diferentes medios de representación. En la actualidad la generación de información se ha vuelto gigantesca y las imágenes forman una parte esencial en estos procesos, en el área visual se ha creado el termino “digitalización”, que define la codificación y generación de cualquier imagen de la realidad, como un mapa finito de unos y ceros, y al aplicarle un adecuado tratamiento representará una imagen digital de muy alta calidad. En este capítulo se enumeran y describen estas nuevas técnicas de representación de la imagen en función de una herramienta sumamente poderosa que es la computadora y todos los medios que se ligan con ella.

## 2.1 LA IMAGEN COMO MEDIO DE COMUNICACION

En el estudio de los mecanismos de procesamiento y representación de la información visual que percibe un ser vivo se encuentran los orígenes de la ciencia y la filosofía. Desde la época de los clásicos griegos (Platón y Aristóteles) hasta nuestros días se han formulado teorías sobre como percibe el ser humano su realidad exterior y de qué manera usa y representa la información captada por sus sentidos. Más cercanos en la historia están los estudios de Kepler sobre el análisis geométrico de la formación de la imagen en el ojo, los de Newton sobre visión en color y dentro de este siglo los de Helmholtz (1910) sobre óptica fisiológica y Wertheimer (1912) sobre el movimiento aparente de agrupaciones de puntos o campos. Estos trabajos, entre otros, han establecido las bases de las actuales teorías de percepción visual.

Sin embargo, el primer gran paso se dio a principio de este siglo con la aparición de la escuela de psicología de la Gestalt a partir de los trabajos de Wertheimer. Esta escuela estableció que los criterios de tipo generalizados tales como similaridad, homogeneidad, cercanía, etc., eran suficientes para tratar de explicar las propiedades de los mecanismos psicofísicos de la visión humana. Pero, esta aproximación con criterios tan globales se abandonó años más tarde al no poder dar respuesta a nuevas interrogantes dentro del proceso de percepción visual.

La reacción que surgió, hacia los años 50, después del fracaso de las teorías de la Gestalt se orientó en la búsqueda de explicaciones al más bajo nivel celular lo que originó una nueva corriente de ideas y teorías que con base en las potencialidades de las neuronas como células individuales trataron de explicar los mecanismos y propiedades de la visión humana, [Hubel & Weisel (1962, 1968) y Barlow (1972)]. Sin embargo, esta aproximación tampoco pudo dar respuesta a los problemas más profundos, tales como nuestro mecanismo visual codifica, representa y reconoce distintos tipos de informaciones espaciales.

Desde la aparición de las primeras computadoras digitales hacia los años sesenta, se puso claramente de manifiesto la gran potencialidad de éstas para el tratamiento de información espacial en campos de aplicación, directamente relacionados con el estudio de propiedades del sistema de visión humana. El primer problema importante que se trató de resolver fue la determinación de los saltos o discontinuidades presentes en una imagen.

Pero dada la enorme complejidad mostrada por el sistema de visión humana y como consecuencia del lento avance de la teorías y algoritmos que explicaban su funcionamiento y propiedades, se intentó abordar estos problemas de formas más directas a partir de tres enfoques.

- *Un primer enfoque*, comandado por las ideas de Azriel Rosenfeld, comenzó el desarrollo de técnicas empíricas basadas en criterios de tipo matemático que aproximaban estos problemas de estimación de

bordes y líneas usando distintos criterios. De esta primera aproximación nacieron algunas ideas interesantes como el uso simultáneo de operadores de distinto tamaño, pero tuvo el gran inconveniente de no poder proponer métodos para la evaluación de los distintos algoritmos. De cualquier manera esta aproximación ha seguido su desarrollo hasta nuestros días con notable éxito en algunos casos.

- *El segundo enfoque*, consistió en profundizar en el problema reduciendo su alcance a un mundo de juguete, es decir, un mundo de bloques, blancos mate, iluminados sobre fondo negro. Los bloques podían tener cualquier forma, siempre que todas sus superficies fueran planas y todos sus bordes rectos. Este modelo funcionó razonablemente bien permitiendo que trabajos como los de Waltz (1975) y Mackworth (1973) permitieran resolver la interpretación de dibujos lineales calculados a partir de imágenes de prismas sólidos. Por supuesto que la simplificación impuesta en el modelo se había hecho con la esperanza de que un conocimiento profundo del problema en un mundo simplificado permitiera más tarde extrapolar dichos conocimientos a mundos más complejos. Pero desafortunadamente esto no ocurrió así y hubo que estudiar un nuevo enfoque.
- *El tercer enfoque*, dentro de este enfoque, se destacan los trabajos pioneros de Horn (1975, 1977) sobre la formación de la imagen. En ellos Horn establece modelos de cálculo que expresan la formación de la imagen a través de ecuaciones diferenciales que relacionan los valores de intensidad de la imagen con la geometría de la superficie, tras estudiar con gran detalle el modo en que la iluminación, la geometría, la reflectancia de la superficie y el punto de vista del observador actuaban de forma conjunta para crear los valores de intensidad medidos en la imagen. Las ideas anteriores a esta tercera aproximación eran que: Debe existir un nivel adicional de comprensión, donde el carácter de las tareas de procesamiento de la información llevadas a cabo durante la percepción, se analicen y comprendan de modo independiente a los mecanismos y estructuras particulares que los implementan en nuestros cerebros.

## 2.2 VISION HUMANA

El sistema de visión humana como otras partes de la anatomía del cuerpo, es fruto de la evolución y los retos a que se ha visto sometido el ser humano en su supervivencia. Como consecuencia de esto, el sistema de visión humano responde mejor a unos patrones que otros y, además puede autoengañarse al aplicar pautas de interpretación en situaciones ambiguas por la existencia de ilusiones visuales, ambigüedades e inconsistencias. Uno de los ejemplos de ilusión es la figura 2.1 de ilusiones ópticas. Por ejemplo en la ilusión de Hering y Wundt las líneas rectas horizontales parecen estar curvas por el efecto de las otras líneas convergentes; ver figura 2.1.

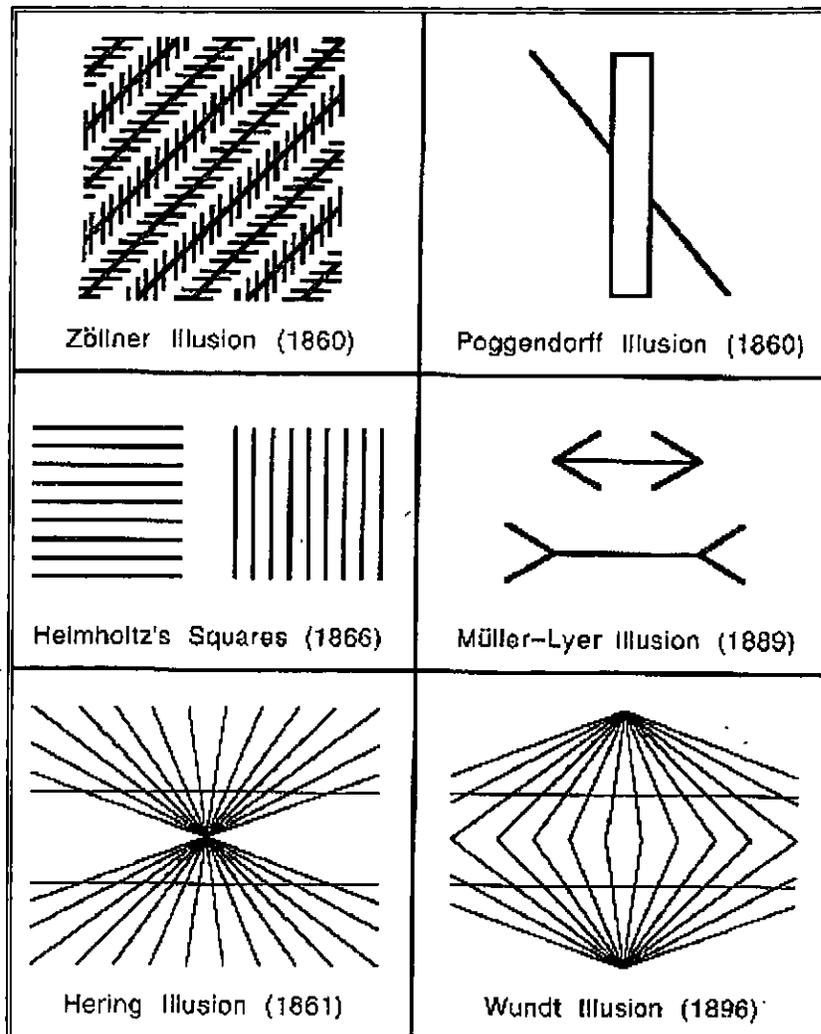


Figura 2.1 [H]

### 2.2.1 Origen de la información.

En forma genérica podemos asociar el concepto de imagen al de un "mapa espacial" o espacio-temporal, esta información produce una estimulación sensorial en el cerebro. Ejemplos de este tipo de "mapas" son nuestras percepciones de una determinada situación o escena a partir de nuestros sistemas sensoriales (vista, oído, tacto, gusto, y olfato). Si nos centramos en el modelo visual humano, asociaremos el concepto de ver con el de percibir una señal luminosa con una intensidad mínima y en un intervalo de frecuencia espectral dado. Sin embargo, hoy en día son bien conocidas las posibilidades de obtener imágenes a partir de sensores que trabajen en condiciones muy distintas de iluminación a las que es sensible el ojo humano, por ejemplo, el infrarrojo, rayos X, etc.

Una pregunta importante es ¿Cuánta información podemos obtener sobre una determinada escena cuando usamos un sistema sensorial?. Para contestar adecuadamente debemos hacer uso de las propiedades físicas de la interrelación de la materia con las ondas electromagnéticas como vehículo de transmisión. Un hecho conocido de la física básica es que el espectro electromagnético de una sustancia es lo que podríamos llamar la "huella digital" de la misma. Si observamos una escena usando un sensor que es sensible a unas

determinadas bandas de frecuencia, la información que obtendremos será la gráfica de valores de la respuesta espectral de dicha escena en las frecuencias del espectro, a las que es sensible el sensor. Por ejemplo, en el caso de humanos, los ojos distinguen solamente tres bandas del espectro electromagnético asociadas a los colores primarios: rojo, azul y verde.

### 2.2.2 Fuentes de energía

La luz visible es una de las muchas fuentes de energía electromagnética a partir de la cual podemos captar imágenes. Las ondas de radio, los rayos ultravioleta, los rayos infrarrojos, los rayos X, etc., son otras fuentes de energía para las cuales hoy en día ya existen sensores especializados para ellas. Aunque aparentemente distintas, estas formas de energía se conducen de acuerdo con la teoría básica de las ondas. Como se muestra en la figura 2.2:

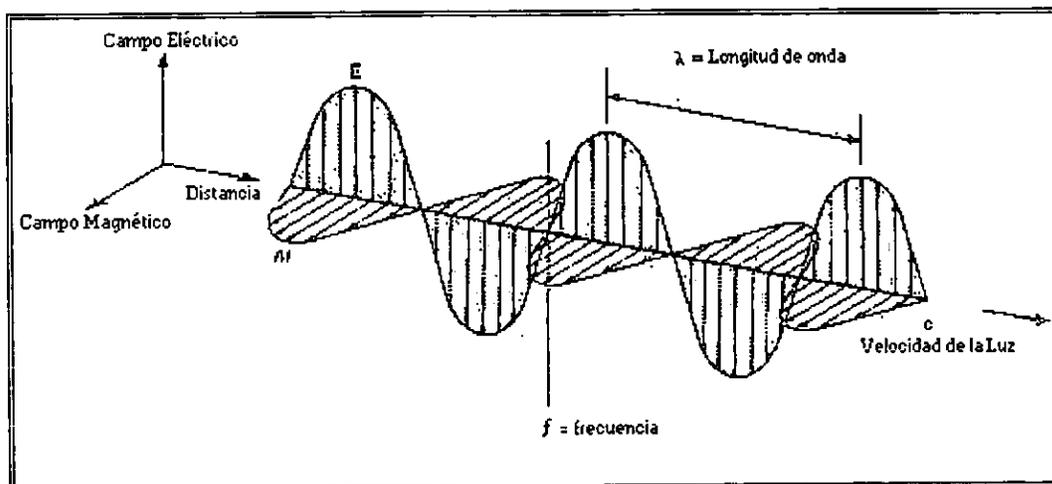


Figura 2.2 [C]

Esta teoría describe la energía electromagnética como ondas sinusoidales que viajan a la velocidad de la luz

$$c = 3 \times 10^8 \text{ [m / s]}$$

[Ecu. 2. 1]

La distancia entre dos picos consecutivos de dichas ondas que denotaremos por  $\lambda$ , se denomina longitud de onda, y el número de picos que pasan por un punto del espacio en cada unidad de tiempo  $f$ , se denomina frecuencia de la onda. La relación entre estos parámetros viene dada por la ecuación

$$c = \lambda \cdot f$$

[Ecu. 2. 2]

Es evidente pues, que frecuencia y longitud de onda son magnitudes inversamente proporcionales ya que  $c$  es esencialmente una constante. Si representamos el espectro en función del logaritmo de la longitud de onda.

· C · A · P · I · T · U · L · O ·

III

MEJORAMIENTO Y  
SEGMENTACION DE LA  
IMAGEN

Dentro de las técnicas de procesamiento de imágenes digitales, las técnicas de mejora y realce tienen como objetivo que la imagen pueda ser visualizada de manera adecuada a través de la pantalla de una computadora y hacer la mejor interpretación posible de la imagen por medio del sistema visual humano. Estas técnicas se centran en la eliminación de ruido en la imagen, realce de bordes, elección de los mejores valores de brillo y contraste, eliminación de los efectos de distorsión introducidos por el mecanismo de captura de la imagen, etc. Una imagen puede representarse como una señal en el dominio del espacio o en el dominio de las frecuencias, por esta razón existen técnicas que permiten llevar a cabo operaciones de realce y/o mejora de la imagen desde ambos dominios de definición.

El primer paso del análisis de imágenes consiste generalmente en segmentar la imagen. La segmentación subdivide una imagen en sus partes constituyentes u objetos, el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. En general, la segmentación autónoma es una de las tareas más difíciles del procesamiento de imágenes, esta etapa del proceso determina el eventual éxito o fracaso del análisis. De hecho, la segmentación rara vez llega a alcanzar una solución satisfactoria. Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similitud. En la primera categoría, el método consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris. Las principales áreas de interés de esta categoría son la detección de puntos aislados la detección de líneas y bordes de una imagen. Los bordes corresponden a cambios locales significativos en una imagen y son probablemente, los rasgos más importantes para el análisis de la misma. Estos se encuentran en la frontera entre dos regiones diferentes, siendo su detección, un paso importante para la recuperación de información. Simplificando la definición podríamos decir que un borde es una discontinuidad de algún tipo en la función de intensidad de la imagen. Existen diferentes tipos de bordes, algunos son debidos a las sombras que se producen sobre los objetos, otros a la variación del reflejo de los objetos o incluso por la textura de los mismos.

### 3.1 METODOS DEL DOMINIO ESPACIAL

El dominio espacial se refiere al propio plano de la imagen, y las técnicas de esta categoría se basan en la manipulación directa de los píxeles de la imagen, los métodos del dominio espacial tienen como base usar los valores de nivel de luminosidad de los píxeles para conseguir mejorar la imagen [1]. Desde un punto de vista matemático lo que queremos saber es, qué tipo de transformaciones podemos aplicar a una imagen para que el resultado sea otra imagen pero con mejoras en sus características visuales, es decir, las transformaciones  $T$  que aplicadas a la imagen  $f(x,y)$  genera otra imagen  $g(x,y)$  con mejores propiedades visuales.

$$g(x,y) = T(f(x,y))$$

[Ecu. 3.1]

Cuando expresamos que  $g(x,y)$  depende de  $f(x,y)$  a través de  $T$ , queremos establecer que el cálculo del valor de  $g$  en el punto  $(x,y)$  depende en principio de todos los valores de  $f(x,y)$ .

En la realidad y usando la hipótesis de que el valor en un punto de imagen está relacionado con unos pocos píxeles que lo rodean, se consideran transformaciones  $T$  que están definidas sobre el píxel  $(x,y)$ , en el entorno  $3 \times 3$  del píxel  $(x,y)$  o en su caso en un entorno  $5 \times 5$  de dicho píxel. Por entorno  $3 \times 3$  de un píxel se

consideran los pixeles que se encuentran directamente cerca de dicho pixel, es decir, los pixeles marcados con una \* de la forma:

*	*	*
*	(x,y)	*
*	*	*

Una de las aproximaciones principales en este tipo de formulación se basa en el empleo de las denominadas máscaras (también llamadas plantillas, ventanas o filtros). El caso más simple se presenta cuando  $T$  depende de un solo pixel. La figura 3.1 muestra gráficas que definen transformaciones  $T$  definidas para cada pixel, donde, la figura 3.1(a) es la transformación para producir una imagen de mayor contraste, oscurece los niveles de gris por debajo de  $m$  e ilumina los niveles por encima de  $m$ ; y la figura 3.1(b) produce una imagen de dos niveles (binaria).

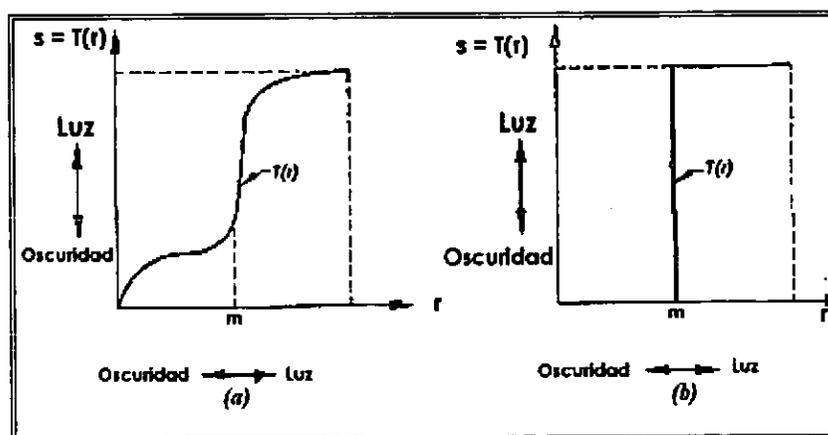


Figura 3. 1 Funciones de transformación del nivel de gris para mejora del contraste. [B]

La elección de estas curvas de transformación afecta fundamentalmente a la mejora de la imagen por la adecuación de sus niveles de gris, en aquellos intervalos que son mejor percibidos por el sistema visual humano.

### 3.1.1 Métodos basados en los pixeles

Son aquellas transformaciones simples que con base en un solo pixel, permiten obtener mejoras en la visualización de la imagen.

◊ **Imágenes de negativos.** Dadas las especiales características no-lineales del sistema visual humano a veces es interesante calcular la imagen invertida de una imagen dada ya que aspectos que resultaban difíciles de percibir pueden estar más claros en la imagen invertida. Casos típicos de imágenes de negativos son las radiografías. La expresión que calcula la inversa de una imagen de niveles de grises (256 o 8 bits) es la siguiente:

$$g(x, y) = 255 - f(x, y)$$

[Ecu. 3. 2]

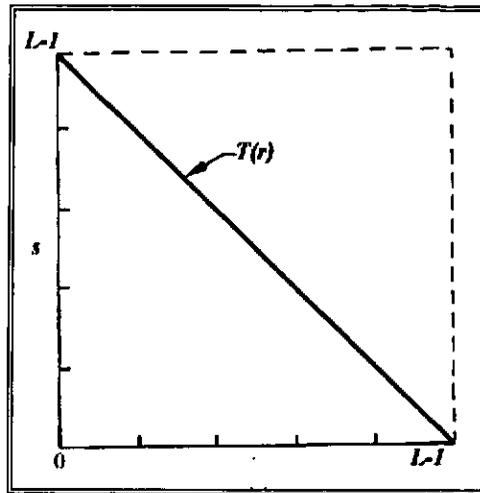


Figura 3. 2 Función de transformación de los niveles de gris para la obtención de un negativo [B]

Donde  $L$  es el número de niveles de gris. En la figura 3.3 se aplicó la función de transformación de niveles de gris representada en la figura 3.2.



Figura 3.3 A la izquierda imagen original, a la derecha su negativo. [A]

◊ **Extracción de Bits.** Supongamos que cada pixel en una imagen es cuantificado uniformemente en  $B$  bits. Deseamos extraer el  $n$ -ésimo bit más significativo y mostrarlo. Sea:

$$u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_n 2^{B-n} + \dots + k_{B-1} 2^1 + k_B$$

[Ecu. 3. 3]

Queremos que el resultado sea:

$$u = \begin{cases} L & \text{si } k_n = 1 \\ 0 & \text{en caso contrario} \end{cases}$$

[Ecu. 3. 4]

donde  $L$  es el número de niveles de gris.

Este tipo de transformación es útil para determinar el número de bits que son visualmente significativos, ya que al mostrar determinados planos puede probarse que ellos no son significativos.

◊ **Mejora del contraste** Diversos factores pueden influir en la ausencia de contraste en una imagen, los más relevantes pueden ser ausencia de iluminación, sensores con un intervalo de valores de iluminación bajo o cualquier otro factor equivalente. La idea fundamental en esta transformación es modificar los valores de los píxeles de manera que se produzca un aumento en el intervalo dinámico de los valores de niveles de gris de la imagen, esto quiere decir que en términos generales debemos de elegir dos valores de nivel de gris uno por la parte alta de los blancos  $r_1$ , y otro por la parte baja de los negros  $s_1$ , de manera que los píxeles con valor igual o superior a  $r_1$  se transformen en niveles más iluminados y aquellos con valor igual o inferior a  $s_1$  se transformen en niveles más oscuros, es decir,

si  $r_2 = T(r_1)$  entonces  $r_2 > r_1$ , y si  $s_2 = T(s_1)$ ,  $s_2 < s_1$ ,

y los píxeles con valores entre  $r_1$  y  $s_1$  se convierten de acuerdo a una transformación lineal a los valores entre  $r_2$  y  $s_2$ . Ya que  $s_1$  es mayor que  $s_2$  y  $r_1$  es menor que  $r_2$ , es evidente que hemos aumentado el intervalo dinámico de los valores centrales del histograma. Una forma de expresar matemáticamente esta transformación es:

$$g(x, y) = \begin{cases} a \cdot f(x, y) & \text{para } f(x, y) < s_1 & \text{siendo } a < 1/2 \\ b \cdot f(x, y) & \text{para } f(x, y) < r_1 & \text{siendo } b < 1/2 \\ c \cdot f(x, y) & \text{para } s_1 < f(x, y) < r_1 & \text{siendo } c < 1/2 \end{cases}$$

[Ecu. 3. 5]

◊ **Compresión del intervalo dinámico.** Existen detectores cuyo intervalo dinámico excede con creces las posibilidades de representación de la pantalla de la computadora. Ejemplo de estos casos son los sensores usados en aplicaciones astronómicas cuyo intervalo dinámico puede ser del orden de miles (1000-30000)[1] o en aplicaciones de tipo médico donde la película de las radiografías y angiografías tienen un intervalo dinámico de más de 1000 niveles de gris distintos. En estos casos es necesario comprimir el intervalo dinámico de los valores de nivel de gris con objeto de poder representar las imágenes en la tecnología actual, que salvo contadas excepciones tan solo puede representar 256 niveles de gris.

La transformación más adecuada para estos casos, consiste en usar una función monótona que permite comprimir los valores, ejemplos de estas transformaciones pueden ser la raíz cuadrada o el logaritmo sobre los niveles de gris. Una expresión matemática genérica de esta transformación para el caso de intervalos muy grandes es:

$$s = c \cdot \log(1 + |r|)$$

[Ecu. 3. 6]

siendo  $c$  una constante de escala,  $r$  y  $s$  indican respectivamente los niveles de gris de la entrada y la salida.

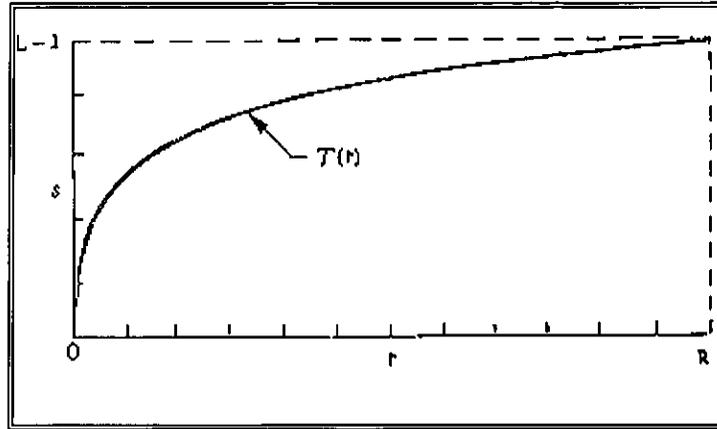


Figura 3. 4 Función de transformación logarítmica [B]

◇ **Realce de intervalos de intensidad.** A veces estamos interesados en resaltar determinados intervalos de niveles de gris a costa de ocultar el resto de valores en la imagen. Este interés puede venir de resaltar una característica de la imagen que conocemos estando asociada a un intervalo concreto de niveles de gris.

### 3.1.2 Métodos basados en el histograma

El histograma de una imagen de niveles de gris se define como una gráfica que a cada valor posible de nivel de gris entre  $[0,255]$  le asigna la cantidad de píxeles en la imagen con dicho valor. Es decir, si la imagen tiene  $N$  píxeles en total y hay  $p_r$  con valor  $r$ , el valor del histograma en  $r$  será:

$$h(r) = p_r / N$$

[Ecu. 3. 7]

Los métodos de realce basados en el histograma tiene como fin modificar la densidad de puntos asignados a los distintos valores de niveles de gris de forma que la imagen mejore en su visualización, es evidente que para conseguir esto de una forma coherente deberíamos definir lo que se entiende por el histograma de una imagen con buenas cualidades visuales. Si bien esta definición no existe como tal, si existe la experiencia de que cuando el intervalo de valores del histograma es el máximo posible y cuando la densidad de puntos está repartida entre los posibles valores  $[0,255]$ , de forma que el histograma se asemeje a la forma de una campana entonces el sistema visual humano consigue una máxima respuesta en su apreciación de la imagen. [1]

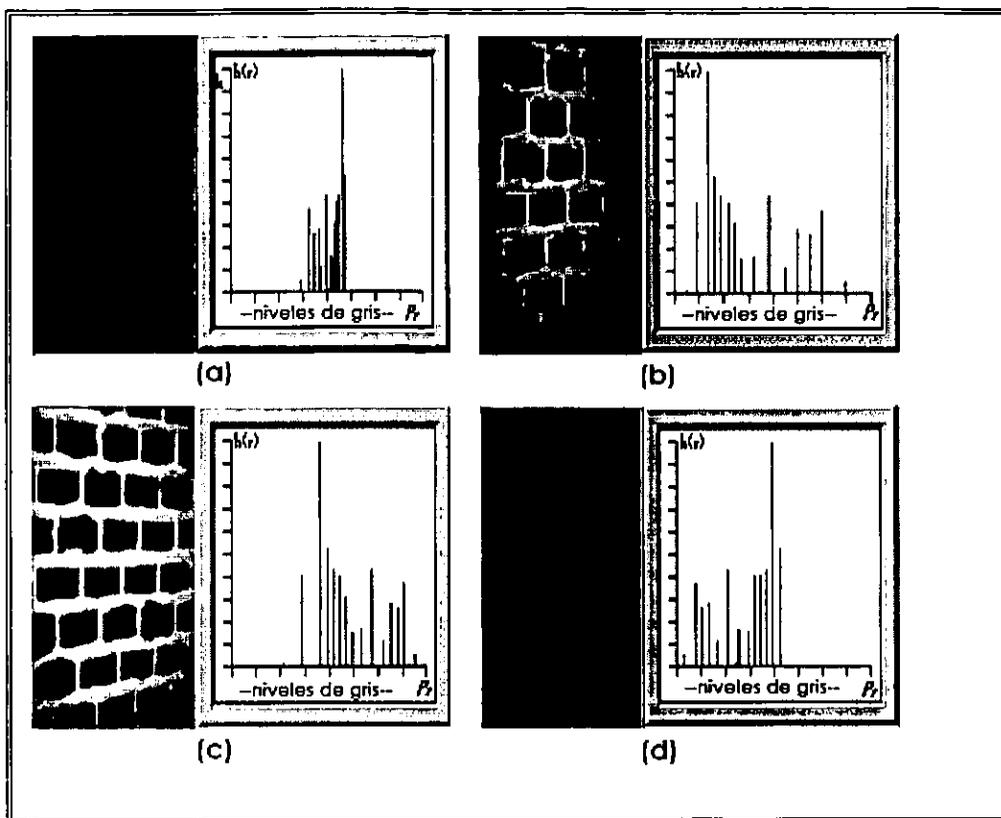


Figura 3.5 Cuatro tipo de imágenes con su respectivo histograma. (a) Imagen de bajo contraste, (b) Imagen de alto contraste, (c) Imagen brillante y (d) Imagen oscura. [A]

◇ **Ecuación del Histograma.** Supongamos que la transformación que estamos estudiando es  $s = T(r)$  siendo  $r$  un valor de nivel de gris en la imagen original y  $s$  su valor en la imagen ecualizada. Entonces el primer objetivo será que el intervalo dinámico de los valores de  $s$  sea al máximo posible, lo que en condiciones de pantallas normales significa  $[0,255]$ . Para que la función  $T$  además de conseguir el objetivo de alargar el intervalo dinámico de los valores de niveles de gris y no introducir perturbaciones sobre la imagen debemos imponer algunas condiciones sobre dicha función, por ahora impondremos tan solo dos condiciones:

- En el intervalo de valores de  $r$ ,  $T(r)$  debe ser una función monótona creciente.
- $0 \leq T(r) \leq 255$ , para  $0 \leq r \leq 255$

La primera condición preserva el orden en los píxeles desde el negro al blanco y la segunda condición garantiza que la función  $T$  conserva sus valores dentro del intervalo permitido, existen infinitas curvas que verifican estas condiciones, de hecho cualquier curva plana que una el origen de coordenadas con el punto  $[255,255]$  de manera que no tenga ni máximos ni mínimos y cuyo grafo se mantenga dentro del cuadrado definido por los puntos  $[0,0]$ ,  $[255,0]$ ,  $[0,255]$  y  $[255,255]$  verifica las condiciones establecidas.

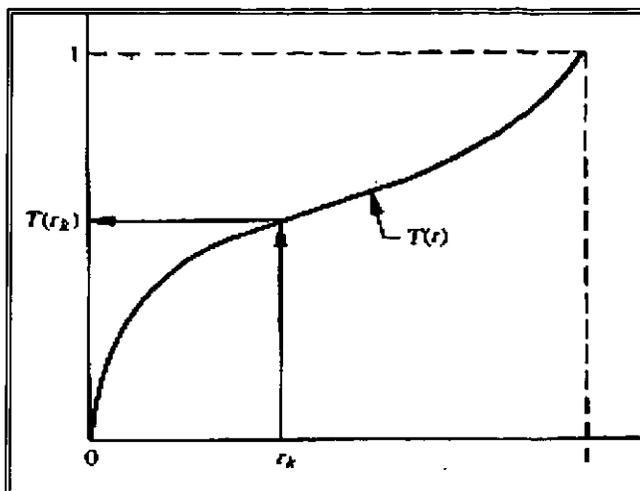


Figura 3. 6 Una función de transformación de los niveles de gris. [B]

Para seleccionar una curva con un comportamiento adecuado a nuestros deseos debemos y podemos considerar que estas funciones son distribuciones acumuladas de probabilidad, aunque para ello ahora supongamos que el intervalo de definición de los valores  $r$  y  $s$  están en  $[0,1]$  en lugar de  $[0,255]$ . Esta hipótesis no resta ninguna generalidad al análisis.

Un resultado elemental de la teoría de la probabilidad establece que si tenemos una variable aleatoria  $R$ , que en nuestro caso sería la función que asigna valor de nivel de gris a cada pixel, la transformación de dicha variable aleatoria por la función acumulada de probabilidad de dicha variable aleatoria,  $P_R(R)$ , define una distribución uniforme de valores, es decir, que si consideramos  $T=P_R$ , el conjunto de valores de  $s$  resultante de la transformación  $s = T(r)$ , sigue una distribución uniforme [1]. Ya que una distribución uniforme supone que todos los puntos tienen la misma densidad de probabilidad, es de interés que usemos esta función  $T$  como medio para redistribuir la densidad de puntos asignada a los distintos niveles de gris, en un caso particular se debe de construir la función  $T$  a partir de la imagen de la siguiente manera:

$$T(r_k) = 255 \sum_{j=0}^k \frac{n_j}{N}$$

[Ecu. 3. 8]

para  $k=0,1,\dots,255$ . Donde  $N$  representa el número total de pixeles en la imagen, y  $n_j$  el número de pixeles con nivel de gris  $j$ .

Una generalización de la técnica anterior nos permitirá transformar una imagen cuyo histograma sea una distribución uniforme, a otra imagen cuya forma de histograma esté definido por nosotros, puede ser de interés transformar el histograma de la imagen original de manera que la imagen resultante tenga un histograma definido por una función en forma de campana (gaussiana). Para llevar a cabo esta transformación habrá que hacer uso de del resultado de la teoría de la probabilidad. Ahora tendremos que hacer dos transformaciones para conseguir el objetivo.

- Transformado la imagen original a una imagen con un histograma definido por la distribución uniforme:  $s = T(r)$  siendo  $T$  la función definida antes.

- Calcular la función inversa del histograma acumulado que queremos que tenga la imagen final:  $G^{-1}$ . Si queremos una imagen con un histograma definido por una función gaussiana tendremos que calcular la función inversa de la función acumulada gaussiana.
- Transformar de nuevo la imagen haciendo uso de la expresión  $z = G^{-1} [T(r)]$ .

La primera transformación nos permite pasar de una imagen con unas características dadas por la forma de su histograma a una imagen con un histograma que aproxima una distribución uniforme, y la segunda transformación nos permite pasar de dicha imagen "neutra" a una nueva imagen cuyo histograma tiene la forma dada por la función  $G$ . En resumen, hemos calculado una nueva imagen a partir de la original usando la transformación.

En la práctica la función puede ser calculada de diversas maneras. Si conocemos su expresión analítica entonces tan solo tenemos que calcular su valor, pero si no es así, podemos usar una versión digitalizada de la gráfica de función  $G$  (histograma acumulado) y a partir de dicha versión digitalizada aproximar el valor que estamos calculando. Si observamos como son las dos transformaciones antes comentadas podemos ver que la primera transforma valores del eje  $x$  en valores del eje  $y$  a través de  $T$ , y la segunda transforma los valores del eje  $y$  en valores del eje  $x$  a través de  $G$ .

### 3.1.3 Resta de imágenes

El procedimiento de mejorar una imagen dada a través de la diferencia con otra imagen en unas condiciones dadas suele ser de mucha utilidad cuando el proceso de captura de la imagen no puede garantizar que todas las imágenes se tomen con la misma intensidad de luz; ejemplos típicos son las imágenes de microscopía óptica, las imágenes astronómicas, etc. Este procedimiento de resta de una imagen puede considerarse como una imagen del fondo de la observación que corrige las posibles variaciones que puedan ocurrir en el transcurso de las observaciones, otro valor añadido que tienen estas operaciones de resta es la supresión de posibles artefactos que pueden aparecer en una imagen como consecuencia de errores o desperfectos en el aparato de medida.

En el caso de aplicaciones médicas, esta técnica puede usarse para resaltar aquellas zonas por las que va pasando una determinada sustancia inyectada por el paciente. La definición matemática de la resta de dos imágenes es otra imagen definida por:

$$g(x, y) = f(x, y) - h(x, y)$$

[Ecu. 3.9]

### 3.1.4 Mejora Local

Los métodos anteriores pueden clasificarse como globales ya que su repercusión es sobre todos los píxeles de la imagen. Sin embargo, existen casos en que la distribución de niveles de gris en la imagen es tal que determinadas transformaciones de tipo global pueden ser adecuadas para unas zonas de la imagen pero no adecuadas para otra. En estos casos es posible que se utilicen las técnicas antes estudiadas pero a nivel de regiones concretas de la imagen, una a una, y no a nivel global de la misma.

La forma más común de llevar a cabo un procedimiento de mejora local, es tomar un rectángulo de un tamaño determinado e ir recorriendo la imagen con dicho rectángulo, centrándolo cada vez sobre un pixel, entonces se actúa sobre los pixeles dentro del rectángulo usando los métodos antes descritos y asignando valor al pixel del centro, usando como única información los pixeles encerrados en el rectángulo. En lugar de usar histogramas, la mejora local puede basarse en otras propiedades de la imagen, estas propiedades pueden ser la media de los niveles de gris y la desviación estándar de los mismos. Estas medidas tienen sentido si pensamos que la media del brillo de la imagen y la desviación típica una medida del contraste de la misma. Una transformación típica basada en estos conceptos es la transformada de una imagen  $f(x,y)$  en otra  $g(x,y)$  de la siguiente manera:

$$g(x,y) = k \frac{[f(x,y) - m(x,y)]}{\sigma(x,y)} + m(x,y)$$

[Ecu. 3. 10]

donde  $k$  es una constante de valor menor o igual al valor medio global de la imagen,  $m(x,y)$  el valor medio en el entorno del pixel  $(x,y)$ , y  $\sigma(x,y)$  la desviación estándar en dicho entorno. De esta expresión se deduce que áreas con poco contraste recibirán una mayor contribución, que áreas con uno mayor contraste.

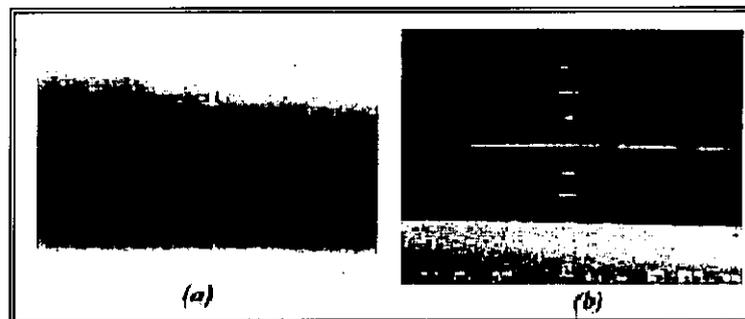


Figura 3.7 Imagen antes (a) y después de la mejora local (b) [B]

### 3.1.5 Promedio de imágenes

La técnica de promediar imágenes tiene como propósito reducir el ruido en ella utilizando elementos conocidos de la estadística. Sabemos de la estadística que si tenemos  $n$  observaciones  $x_1, x_2, \dots, x_n$  extraídas de una distribución con media  $\mu$  y desviación típica  $\sigma$ .

$$\sum_{i=1}^n x_i / n$$

[Ecu. 3. 11]

Entonces la media de dichos datos sigue una distribución con media, la media de la población  $\mu$ , y desviación estándar dada por:

$$\sigma_{\bar{x}} = \sigma / \sqrt{n}$$

[Ecu. 3. 12]

Aplicando este resultado al caso de imágenes vemos que si podemos obtener una imagen como promedio de un conjunto de imágenes tomadas en las mismas condiciones, entonces el nivel de error en los valores de los pixeles decrecerá con el tamaño de la muestra en el factor raíz de  $n$ , un punto importante es que las imágenes deben estar todas centradas sobre la misma región de una forma muy escrupulosa, ya que de otra manera el procedimiento de promedio conduciría a opacar más que a su mejora y realce. Este procedimiento de promedio es el que usan muchas tarjetas de adquisición de imágenes fijas, ya que integran la señal durante un corto periodo de tiempo.

### 3.2 FILTRADO ESPACIAL

El concepto de filtrado de una imagen esta asociado a la representación de una imagen en el dominio de frecuencias, cualquier filtro que se diseñe tendrá como objetivo modificar la contribución de determinados intervalos de frecuencias a la formación de la imagen, por ejemplo si queremos filtrar una imagen de una contaminación de ruido aleatorio lo que tendremos que hacer es diseñar un filtro que reduzca lo más posible la contribución de las altas frecuencias en la formación de la imagen, en cambio si lo que queremos es realzar cualquier patrón de comportamiento presente y sabemos que dicho patrón está asociado a un determinado intervalo de frecuencias, lo que haremos será construir un filtro que disminuya la contribución de cualquier frecuencia que no este dentro del intervalo deseado.

Los filtros más comunes son: filtros paso bajo que suprimen las frecuencias altas, filtros paso alto que suprimen las frecuencias bajas y filtros paso banda que suprimen determinados intervalos de frecuencias. Aunque la terminología de la teoría del filtrado lineal es esencialmente del dominio de frecuencias, el teorema de convolución nos permitirá hablar también de filtros en el dominio espacial aunque el sentido intuitivo de dichos filtros no sea tan claro como en el dominio de las frecuencias. Si notamos por  $G(u,v)$  la TF (Transformada de Fourier) de la imagen original y por  $H(u,v)$  la función de transferencia del filtro que queremos aplicar, entonces sabemos que:

$$\mathfrak{F}^{-1}[F(u,v)H(u,v)] = f(x,y) * h(x,y)$$

[Ecu. 3. 13]

Por tanto, conocer saber qué formas adoptaran las funciones  $h(x,y)$  asociadas a determinados filtros de frecuencias, a estas funciones  $h(x,y)$  se les denominan filtros espaciales.

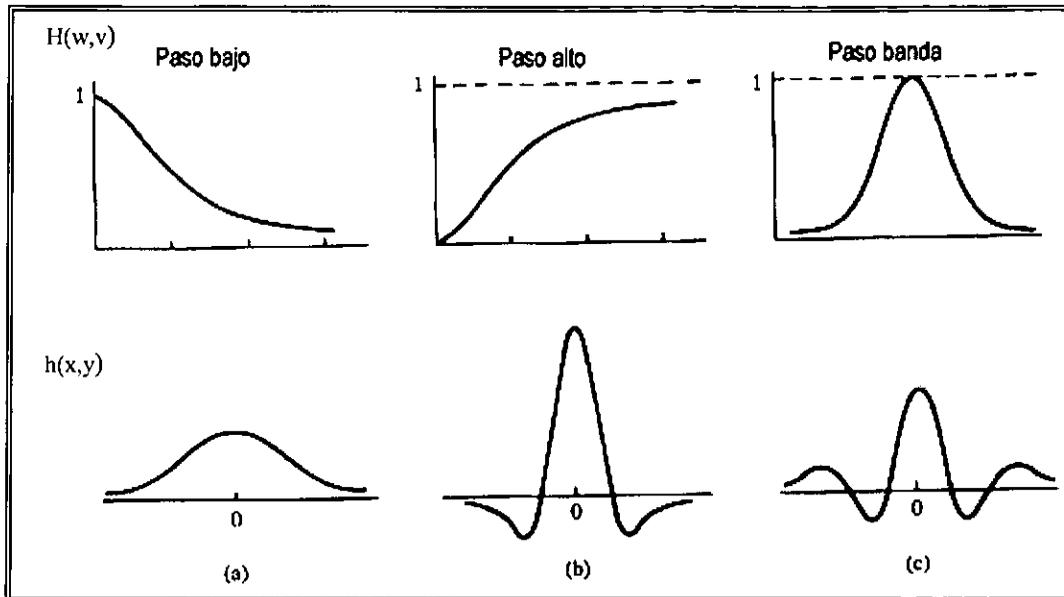


Figura 3. 8 Formas asociadas a los tres tipos básicos de filtros. El caso (a) representa a los filtros paso bajo, el caso (b) a los paso alto y el caso (c) a los paso banda. [B]

Para diseñar un filtro espacial  $h(x,y)$  en teoría tendríamos que calcular los valores de la función sobre una retícula de igual tamaño que la definida por la imagen definida por  $f(x,y)$ , pero en la práctica puede considerarse que los valores de  $h(x,y)$  decrecen muy rápidamente cuando nos vamos alejando del origen. Es práctica habitual discretizar dicha función para construir aproximaciones usando máscaras de distintos tamaños. Las más comunes serán las de tamaños  $3 \times 3$ ,  $5 \times 5$  o  $7 \times 7$ . Un ejemplo de máscara  $3 \times 3$  es de la forma:

$w_{-1,-1}$	$w_{-1,0}$	$w_{1,0}$
$w_{0,1}$	$w_{0,0}$	$w_{0,-1}$
$w_{1,-1}$	$w_{1,0}$	$w_{1,1}$

donde los  $w_{ij}$  son los valores discretizados en las posiciones de sus subíndices de la función, en el caso de funciones simétricas, los valores de la máscara a igual distancia del origen tendrán el mismo valor. El diseño de filtros espaciales se resume en calcular un conjunto de pesos que definan una máscara cuya TF tenga el comportamiento de uno de los tres filtros básicos.

Existen filtros de tipo no-lineal, en estos casos tanto la convolución como herramienta base del filtrado y la suma como operación matemática que la define dejan de tener sentido, y habrá que calcular usando funciones de tipo no lineal. Los casos más sencillos son aquellos en que se substituye la suma por la operación de máximo o mínimo. La teoría de la Morfología Matemática, entre otras aproximaciones, es una alternativa para el filtrado no-lineal representado la aproximación más sólida en este camino.

3.2.1 Filtros suavizantes.

Los filtros suavizantes sirven para que la imagen aparezca suavizada o para reducir su ruido. Los procesos para suavizar la imagen son fundamentales en muchos procesos, como la eliminación de los pequeños detalles de una imagen antes de la extracción de un objeto, y el relleno de pequeños espacios entre líneas y curvas.

◊ **Filtro espacial paso bajo.** La forma de la respuesta impulso (ver figura 3.8 ) necesaria para implementar un filtro paso bajo implica que los coeficientes  $w_{i,j}$  deben de ser positivos. Cualquier función positiva con un solo máximo puede ser usada para definir un filtro paso bajo, un ejemplo típico de este tipo de funciones lo representa la función gaussiana, ya que el cálculo de una convolución por una máscara de tamaño  $n \times n$ , representa por  $n^2$  multiplicaciones de valores pixeles por coeficientes  $w_{i,j}$  y sumar su resultado, no está garantizado que el resultado que obtengamos sea un valor posible para un pixel, es decir, encontrará un valor en el intervalo  $[0,255]$ . Para poder conseguir esto hay que imponer que la suma de los  $w_{i,j}$  sea igual a 1.

$$\sum_{i=1}^n \cdot \sum_{j=1}^n w_{i,j} = 1$$

[Ecu. 3. 14]

Para ello, lo que haremos será dividir cada valor , por la suma de todos ellos

$$w'_{i,j} = \frac{w_{i,j}}{\sum_{i=1}^n \cdot \sum_{j=1}^n w_{i,j}}$$

[Ecu. 3. 15]

Es evidente que cuanto más estrecha sea la respuesta impulso que estemos usando menos coeficientes serán necesarios para obtener una adecuada versión discretizada de la misma. Usualmente el tamaño de la máscara se toma en función de la cantidad de suavizamiento que queramos aplicar en cada momento, la visualización del resultado es el único medio para saber si el tamaño elegido fue el adecuado.

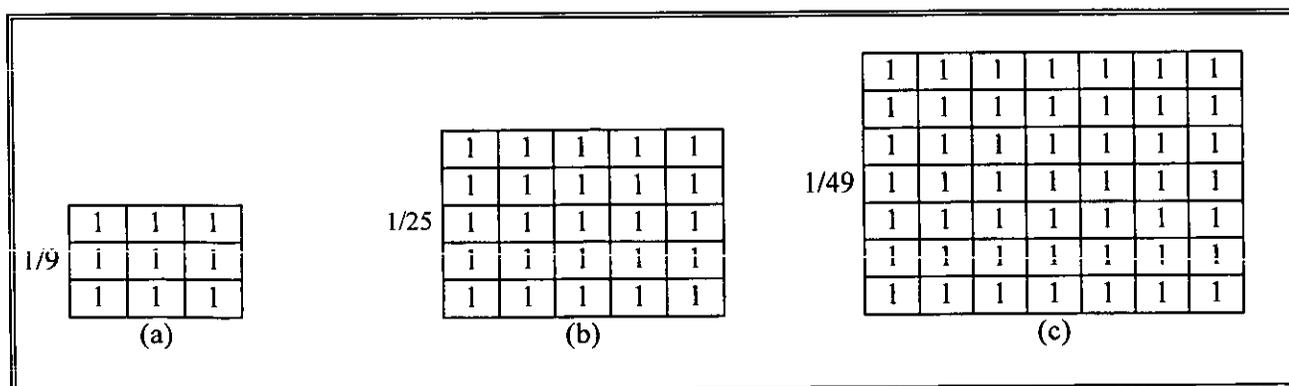


Figura 3.9 Filtros espaciales paso bajo de diferentes tamaños; (a) 3x3, (b) 5x5 y (c) 7x7 [B]

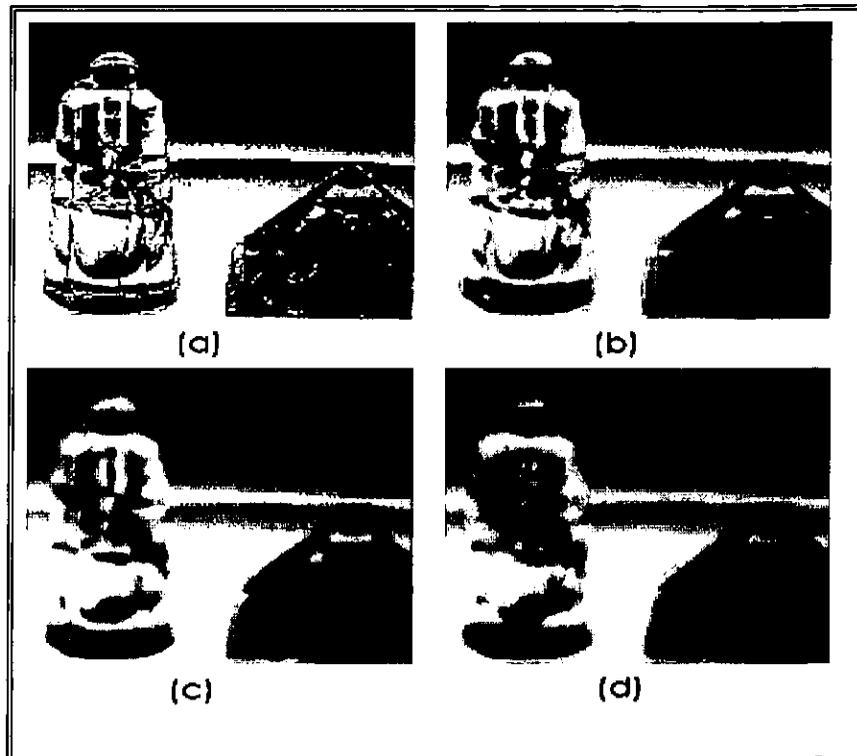


Figura 3. 10(a) imagen original; (c)-(d) el resultado del filtrado espacial paso bajo empleando máscaras de tamaño  $n \times n$ , con  $n=3,5$  y  $7$ . Ver las máscaras en la figura 3.9. [A]

◊ **Filtro por la mediana.** Uno de los principales inconvenientes que presenta el suavizamiento a través de la operación de convolución es la paulatina desaparición de los saltos entre los niveles de gris de pixeles vecinos, dichos saltos son los que definen las fronteras presentes en la imagen, por tanto, la convolución hace decrecer la fuerza con que las fronteras se presentan en la imagen. Cuando el objetivo es reducir el ruido sin borrar las fronteras, una alternativa es el uso de filtros mediana.

La definición de estos filtros es : El nivel de gris de cada pixel se reemplaza por la mediana de los niveles de gris en un entorno de este pixel, en lugar de por la mediana.

$$\text{mediana} \left\{ \begin{array}{l} f(x-1, y-1), f(x-1, y), f(x-1, y+1), f(x, y-1), \\ f(x, y), f(x, y+1), f(x+1, y-1), f(x+1, y), f(x+1, y+1) \end{array} \right\}$$

[Ecu. 3. 16]

Estos filtros son adecuados para suprimir ruidos aleatorios y picos sin significado, ya que logran que un pixel sea lo más parecido posible a los pixeles de su entorno. El tamaño de la máscara fijará que clase de picos estamos considerando, así como los no deseables dentro de la imagen.

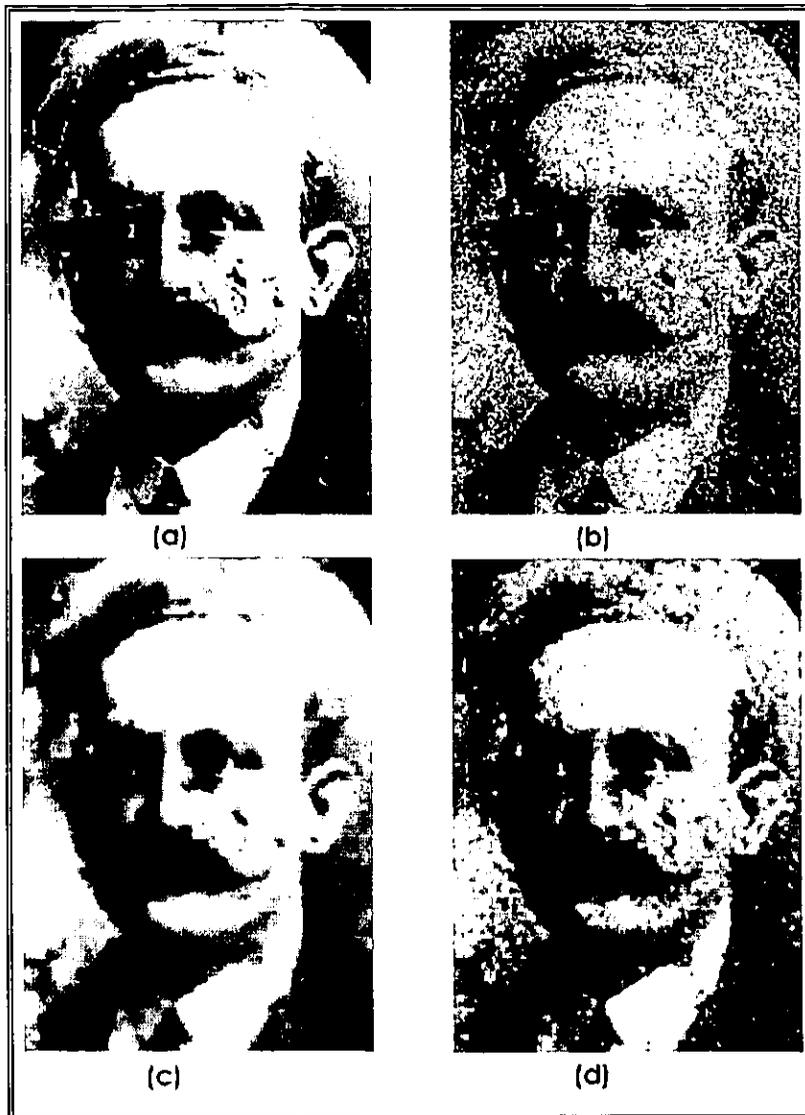


Figura 3. 11 (a) Imagen original; (b) imagen corrompida por ruido;(c) resultados del promedio en un entorno  $5 \times 5$ ; (d) resultado producido por un filtro de mediana. [A]

### 3.2.2 Filtros de realce.

El principal objetivo de los filtros de realce es el de resaltar aquellas características de la imagen que por causa del mecanismo de captación o defectos en la imagen, estos filtros son usados como un método directo para mejorar una imagen y presentarla a un observador humano. Con mucha frecuencia la característica más importante a realzar son las fronteras que definen los objetos presentes en la imagen.

◇ **Filtros paso alto.** La forma de la función respuesta impulso (ver figura 3.8) asociada a un filtro paso alto establece que debemos de tomar valores positivos en el centro de la función y valores negativos a su alrededor, si consideramos construir un máscara  $3 \times 3$  que implemente un filtro paso alto deberemos de tomar el valor central de la máscara positivo y el conjunto de valores que le rodean negativos; Otra condición importante de estas máscaras es que la suma de sus valores sea cero.

$$\sum_{i=1}^n \cdot \sum_{j=1}^n w_{i,j} = 0$$

[Ecu. 3. 17]

La definición anterior es importante ya que ello garantiza que en una zona donde la imagen tenga variaciones pequeñas, o sea, plana el resultado del filtrado será cero o un valor pequeño, esto implica que este tipo de filtro hace desaparecer las bajas frecuencias definidas por dicho tipo de zonas, y como consecuencia de esto disminuye drásticamente el contraste en la imagen, un ejemplo típico de este tipo de máscaras viene dado por la siguiente configuración:

$$1/9 \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Donde el factor 1/9 garantiza que el valor resultante este siempre entre [-255,255]. Ya que estos filtros pueden producir valores negativos que son causa de problemas por su difícil interpretación y representación como imágenes de niveles de gris, se han desarrollado otras formas de filtrado paso alto exentos de estos problemas.

◇ **Filtros de énfasis de las frecuencias (high-boost).** La idea fundamental en este tipo de filtros es el realce de las altas frecuencias presentes en la imagen pero sin perder la contribución de las bajas frecuencias.

Si partimos del hecho que:

$$\begin{aligned} \text{De énfasis} &= (A)(\text{Imagen Original}) - (\text{Imagen Paso Bajo}) \\ &= (A-1)(\text{Imagen Original}) + (\text{Imagen Original}) - (\text{Imagen Paso Bajo}) \\ &= (A-1)(\text{Imagen Original}) + (\text{Imagen Paso Alto}) \end{aligned}$$

donde A es el factor de amplificación.

Esto muestra que cuando A=1 este filtro coincide con el filtro paso alto, pero cuando A>1 se enfatiza la imagen original que se añade al resultado del filtro paso alto dando lugar a que no se pierdan las contribuciones de las bajas frecuencias, el resultado final es una imagen más parecida a la imagen original pero con las fronteras enfatizadas en función del valor de A.

Una forma de implementar en una máscara este filtro paso alto es usar la máscara anteriormente definiendo el valor central de la máscara de acuerdo a  $w = 9A - 1$ .

$$1/9 \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & w & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Al igual que en el caso de filtros paso bajo es posible aplicar cualquier tamaño de máscara, aunque en la práctica tamaños superiores a tres no son frecuentes.

En la figura 3.12 se observa que conforme aumenta el valor de A, el fondo de la imagen se hace más brillante, preservándola; el caso contrario ocurre cuando el filtro es del tipo paso alto dejando pasar solo los bordes en la imagen y cancelando todo lo demás, logrando el realce.

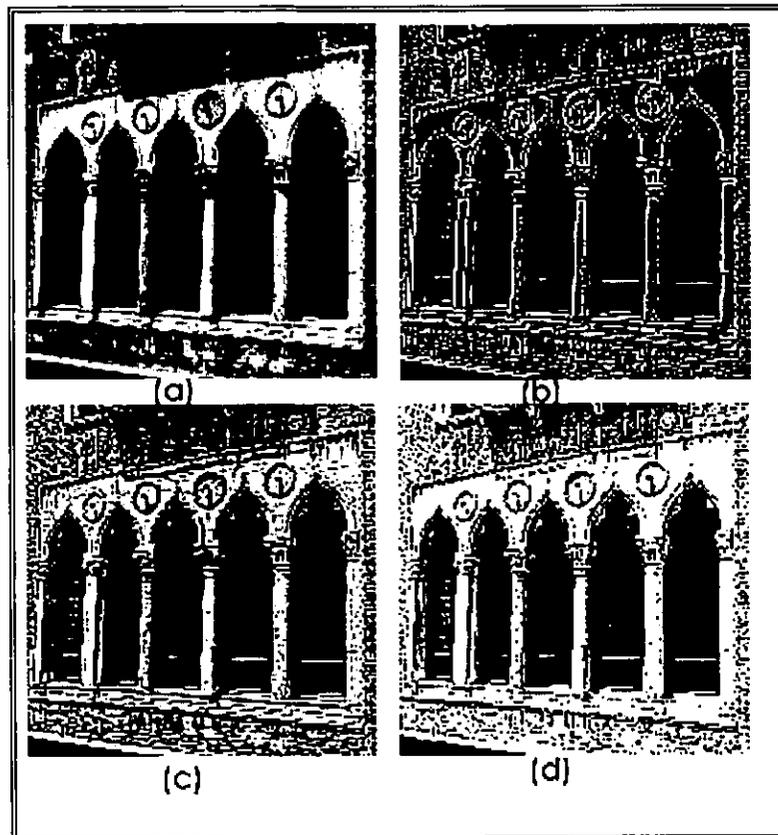


Figura 3. 12(a) imagen original; (b) resultado del filtrado de énfasis empleado  $A=1$  (es decir, un filtro paso alto); (c) filtrado de énfasis empleado  $A=1.11$  y (d) filtrado de énfasis empleado  $A=1.33$  [A]

### 3.2.3 Filtros de derivadas.

De forma paralela pero a la inversa de como actúan los filtros suavizantes basados en integración, los filtros basados en la derivación de la imagen resaltan las altas frecuencias. El método más común de cálculo de la derivada en un punto de la imagen es el cálculo de su gradiente, este se define el gradiente en cada punto de la imagen como un vector cuyas componentes son la derivada en el sentido del eje x y la derivada en el sentido del eje y. En notación matemática esto es:

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

[Ecu. 3. 18]

El valor del módulo de este vector se calcula como la longitud de un vector en una geometría de tipo euclídeo, es decir :

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

[Ecu. 3. 19]

Por cuestiones de velocidad de cálculo el módulo del gradiente también se aproxima usando otro tipo de función distancia, distinta de la euclídea. La cuestión más importante que plantean estos filtros es la estimación de los valores de  $\nabla f$  y del módulo del vector a partir de los valores discretos de la retícula de la imagen. Este tipo de filtros dan como salida una nueva imagen cuyos valores son, en cada pixel, el modulo del gradiente en dicho punto de la imagen de entrada. Las distintas aproximaciones dadas para la estimación de estos valores así como para el cálculo del módulo del gradiente, han servido como base para la definición de distintos tipos de máscaras de realce. Notemos los valores de una máscara 3x3 representada en términos de las variables z, que son las que indican el nivel de gris de la siguiente forma:

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

◊ **Operadores de Robert.** Los operadores de Robert del gradiente cruzado aparecen de considerar máscaras de tamaño 2x2 para estimar la siguiente expresión para el modulo del gradiente:

$$\nabla f \approx (|z_5 - z_9| + |z_6 - z_8|)$$

[Ecu. 3. 20]

El cálculo del valor de las siguientes máscaras y su suma implementaría la expresión anterior, las máscaras con un número par de emplazamientos son más complicadas de implementar.

1	0	0	1
0	-1	-1	0

◊ **Operadores de Prewitt.** Los operadores de Prewitt están basados en la estimación del módulo del gradiente usando máscaras 3x3. Los dos operadores, en la dirección del eje "x" y en la dirección del eje "y" son:

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

La suma de los resultados a partir de estas máscaras dan la siguiente estimación del modulo del gradiente:

$$|\nabla|f \approx \left| (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \right| + \left| (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \right|$$

[Ecu. 3. 27]

◊ **Operadores de Sobel.** Un tipo de máscaras más complejas esta dada por los operadores de Sobel [1]. Estos operadores están dados por las siguientes máscaras:

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

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

La ventaja adicional que presentan estas máscaras sobre las anteriores es que además de estimar el valor del módulo del gradiente, producen un suavizamiento en la imagen que es benéfico, dado el comportamiento ruidoso que presentan las estimaciones basadas en derivadas, para ver que estas máscaras tienen este comportamiento basta fijarse en que la aplicación de estas máscaras es equivalente a la aplicación sucesiva de dos máscaras sobre la imagen, una sobre filas y la otra sobre las columnas. La primera máscara será la máscara diferencial (-1,0,1) que aplicaremos sobre cada columna/fila y después al resultado le aplicamos la máscara de suavizamiento (1,2,1) sobre cada fila/columna.

El resultado de la figura 3.13 nos pone de manifiesto otro aspecto importante del cálculo con máscaras, y es el hecho de poder calcular máscaras en dos dimensiones a partir de la composición de máscaras en una dimensión. En aquellos casos en que esto pueda realizarse, es obvio que la eficiencia del cálculo aumentará considerablemente, ya que el número de operaciones a realizar disminuye drásticamente.

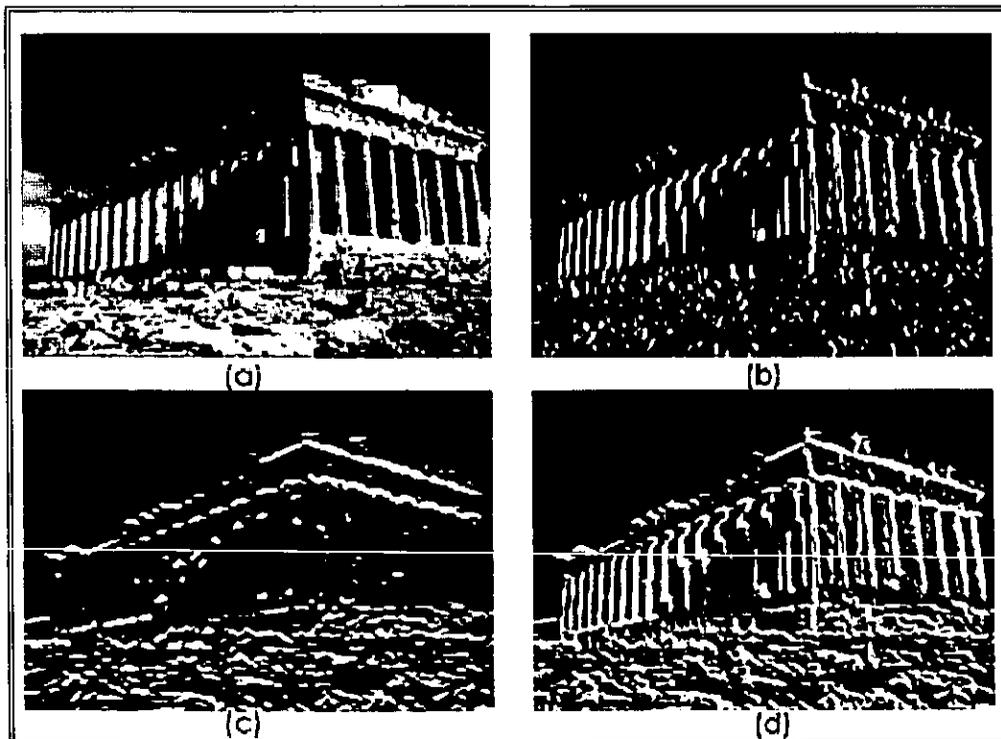


Figura 3.13 (a) imagen original; (b) resultado de aplicar el filtro Sobel para realce de bordes con orientación horizontal y (d) la suma de los efectos después de aplicar los dos tipos de filtros de Sobel[(b) y (c)][A]

### 3.3 MEJORA EN EL DOMINIO DE LA FRECUENCIA

Los principios de mejora en el dominio de las frecuencias son claros. Simplemente se trata de calcular la transformada de Fourier de la imagen a intensificar, multiplicar el resultado por la función de transferencia de un filtro y finalmente, tomar la transformada de Fourier inversa para llegar a una imagen mejorada. La idea general del filtrado lineal es bastante más atractiva e intuitiva en el dominio de la frecuencia. En la Práctica, las pequeñas máscaras espaciales son mucho más empleadas que las transformadas de Fourier debido a su facilidad de implementación y a su velocidad de operación.

### 4.4 METODOS EN EL DOMINIO DE LA FRECUENCIA.

El fundamento de las técnicas basadas en el dominio de las frecuencias es el teorema de convolución, este teorema dice que si  $g(x,y)$  es la imagen formada por la convolución de la imagen  $f(x,y)$  y un operador lineal invariante  $h(x,y)$ , es decir,

$$g(x, y) = h(x, y) * f(x, y)$$

[Ecu. 3. 22]

entonces por el teorema de convolución, se verifica en el dominio de las frecuencias la siguiente ecuación,

$$G(u, v) = H(u, v)F(u, v)$$

[Ecu. 3. 23]

donde  $G, H$  y  $F$  son las transformadas de Fourier de  $g, h$  y  $f$  respectivamente.

$$g(x, y) = \mathfrak{F}^{-1} [H(u, v)F(u, v)]$$

[Ecu. 3. 24]

Muchos problemas de mejora de imágenes pueden ser formulados de acuerdo con los siguientes planteamientos. Dado  $f(x,y)$ , una vez calculado  $F(u,v)$  el objetivo es seleccionar  $H(u,v)$ , de forma que la imagen deseada muestra algunos rasgos de interés de  $f(x,y)$ , A continuación se plantean dos ejemplos con sus posibles soluciones:

- A. Los bordes en la imagen  $f(x,y)$  pueden ser acentuados usando una función que enfatice las frecuencias altas de  $F(u,v)$ . Existen dos posibles soluciones:
- a)  $h(x,y)$  caracteriza un sistema cuyo objetivo es producir una imagen de salida  $g(x,y)$  a partir de una imagen de entrada  $f(x,y)$ . El sistema realiza la convolución de  $h(x,y)$  con la imagen de entrada y produce el resultado.
  - b) El teorema de convolución permite dar una nueva interpretación al proceso, en el sentido de que se puede alcanzar el mismo resultado si multiplicamos  $F(u,v)$  por  $H(u,v)$  para producir  $G(u,v)$ . Calculando la transformada de Fourier inversa de la salida produce la imagen deseada.

B. Supongamos por un momento que  $h(x,y)$  es desconocido y que aplicamos una función de impulso unitario al sistema.

- a) La transformada de Fourier del impulso unitario es simplemente uno y, por tanto, tenemos  $G(u,v) = H(u,v)$  por tanto, la transformada inversa de  $G(u,v)$  es  $h(x,y)$ . Este resultado es bien conocido en la teoría de los sistemas lineales: Un sistema lineal invariante a traslaciones se especifica completamente por su respuesta a un impulso. [1]

Desde el punto de vista del dominio de las frecuencias es importante notar que la mejora de imágenes no coincide con el problema de la restauración de las mismas, aunque su formulación podría resultar muy similar. La mejora en el dominio de las frecuencias es en principio directa, simplemente calculamos la transformada de Fourier de la imagen que queremos mejorar, multiplicamos el resultado por una función de transferencia (un filtro), y calculamos la transformada de Fourier inversa para producir la imagen mejorada.

Las ideas para suavizar una imagen mediante la reducción del contenido de las frecuencias altas o de realzar aumentando el contenido (la magnitud) de las frecuencias altas en relación a las bajas vienen obviamente del dominio de Fourier, de hecho la idea de filtrado lineal se considera más atrayente e intuitiva en el dominio de las frecuencias. Sin embargo, en la práctica, las máscaras espaciales pequeñas se usan más que la transformada de Fourier por su simplicidad de implementación y velocidad en las operaciones, sin embargo, la comprensión de los conceptos en el dominio de las frecuencias es esencial para la solución de problemas de difícil comprensión en el dominio espacial.

### 3.4.1 Filtrado Paso Bajo

Las fronteras y otras transiciones rápidas como el ruido en los niveles de gris de una imagen contribuyen significativamente al contenido de las frecuencias altas en el dominio de Fourier, por tanto, la suavización de la imagen se alcanza en el dominio de Fourier atenuando un intervalo especificado de componentes de altas frecuencias en la transformada de una imagen dada, es decir:

$$G(u, v) = H(u, v)F(u, v)$$

[Ecu. 3. 25]

Donde  $F(u,v)$  es la transformada de Fourier de la imagen a suavizar, el problema es seleccionar una función de transferencia  $H(u,v)$  que produzca  $G(u,v)$ , atenuando las componentes de alta frecuencia de  $F(u,v)$ . La transformada inversa producirá entonces la imagen que queremos, consideramos las funciones de transferencia que afectan la parte real e imaginaria de la misma forma [1].

◇ **Filtros Ideales.** Un filtro ideal bidimensional es uno cuya función de transferencia cumple la relación:

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \leq D_0 \\ 0 & \text{en caso contrario} \end{cases}$$

[Ecu. 3. 26]

donde  $D_0$  es una cantidad no negativa y  $D(u,v)$  es la distancia del punto al origen en el plano de las frecuencias, es decir,

$$D(u, v) = (u^2 + v^2)^{1/2}$$

[Ecu. 3. 27]

El nombre ideal proviene de que todas las frecuencias dentro de un círculo de radio  $D_0$  se pasan sin atenuación, mientras que todas las frecuencias fuera de ese círculo son atenuadas completamente. Los filtros paso bajo son simétricos alrededor del origen, todas las frecuencias fuera de ese círculo son atenuadas. Para un filtro ideal paso bajo radialmente simétrico, el punto de transición entre  $H(u,v) = 1$  y  $H(u,v) = 0$  recibe usualmente el nombre de frecuencia de corte.

El funcionamiento de todos los filtros paso bajo que se introducen a través del dominio de Fourier se puede comparar examinando las frecuencias de corte. Una forma de establecer un conjunto de frecuencias de corte estándar es calcular círculos que incluyen varias cantidades de la potencia de la señal  $P_T$ . Esta cantidad se obtiene como sigue:

$$P_T = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} P(u, v)$$

[Ecu. 3. 28]

donde  $P(u,v)$  denota el espectro de potencias. Si la transformación ha sido centrada, en un círculo de radio  $r$  con el origen en el centro de las frecuencias contiene  $\beta$  por ciento de la potencia,

$$\beta = 100 \left[ \sum_u \sum_v P(u, v) / P_T \right]$$

[Ecu. 3. 29]

donde la suma se toma en los valores que están dentro del círculo o su frontera.

La suavización y lo que se ha dado en llamar "ringing" de este tipo de filtros puede explicarse fácilmente de acuerdo con el teorema de convolución. Puesto que la transformada de Fourier de la imagen original y la borrosa se relacionan en el dominio de Fourier mediante,

$$G(u, v) = H(u, v)F(u, v)$$

[Ecu. 3. 30]

El uso de este teorema conduce a la siguiente expresión en el dominio espacial

$$g(x, y) = h(x, y) * f(x, y)$$

[Ecu. 3. 31]

donde  $h(x,y)$  es la transformada de Fourier inversa del filtro de transferencia  $H(u,v)$ .

El paso para entender el proceso de suavización como un proceso de convolución en el dominio espacial está en la forma de  $h(x,y)$ .

◊ **Filtro de Butterworth.** La función de transferencia del filtro de Butterworth de orden  $n$  y frecuencia de corte a distancia  $D_0$  tiene la forma

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$

[Ecu. 3.32]

Donde  $D(u,v)$  es la distancia euclídea al centro de la imagen.

Al contrario de los filtros ideales el filtro de Butterworth no tiene una discontinuidad muy pronunciada que establece un corte claro entre las frecuencias que pasan y las que se discriminan.

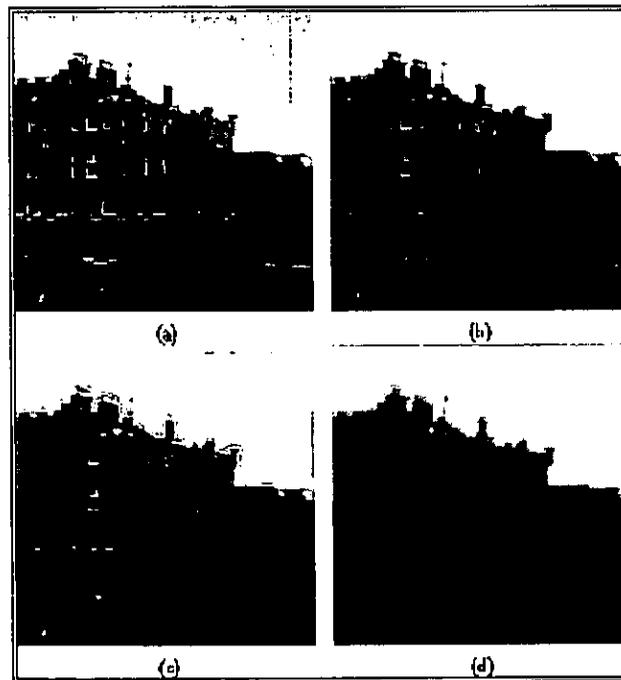


Figura 3.14 (a) Imagen original, (b) Butterworth de orden 1 y corte 0.5; (c) Butterworth de orden 6 y corte 0.5; (d) Butterworth de orden 1 y corte 0.1.

### 3.4.2 Filtrado Paso Alto

Como ya sabemos, una imagen puede ser suavizada atenuando las componentes de frecuencia alta en su transformada de Fourier, ya que los bordes y otros cambios abruptos en los niveles de gris se asocian con frecuencias altas. El realce de imágenes puede alcanzarse mediante el filtrado paso alto que atenúa las frecuencias bajas sin distorsionar la información en las frecuencias altas en el dominio de Fourier.

- ◇ **Filtro Ideal.** Un filtro bidimensional ideal paso alto es uno cuya función de transferencia cumple la relación,

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \geq D_0 \\ 0 & \text{en caso contrario} \end{cases}$$

[Ecu. 3.33]

donde  $D_0$  es la distancia de corte medida desde el origen del plano de frecuencias y  $D(u, v)$  ya ha sido definida.

- ◇ **Filtro de Butterworth.** La función de transferencia de un filtro de Butterworth paso alto viene definida por

$$H(u, v) = 1 - \frac{1}{1 + [D(u, v) / D_0]^{2n}} = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

[Ecu. 3.34]

donde las definiciones de los elementos que aparecen en él son las mismas que para el filtro de Butterworth paso bajo.

### 3.5 SEGMENTACION BASADA EN FRONTERAS

Podemos definir un borde como un cambio local significativo en la intensidad de la imagen que usualmente está asociado con una discontinuidad de la intensidad o su derivada [11]. Las discontinuidades pueden ser de dos tipos:

1. Salto donde la intensidad cambia abruptamente de un valor en un lado de la discontinuidad a otro valor diferente en el lado opuesto.
2. Discontinuidades de línea, donde las intensidades cambian abruptamente, regresando en poco espacio a su valor inicial.

Las siguientes definiciones caracterizan un detector de bordes:

- Un punto de borde es un punto geométrico con coordenadas  $[i, j]$  en la posición de un cambio de intensidad local significativo en la imagen.
- Un fragmento de borde es la coordenada  $i$  y  $j$  de un borde y la orientación  $\theta$  que puede ser el ángulo del gradiente.
- Un detector de bordes es un algoritmo que produce un conjunto de bordes a partir de la imagen.
- Un contorno es una lista de bordes, o la curva matemática que modela la lista de bordes.

- La unión de bordes es el proceso de formar una lista ordenada de bordes a partir segmentos dispersos.
- El seguimiento de bordes es el proceso de búsqueda en la imagen (filtrada) para determinar los contornos.

Las coordenadas de un punto borde pueden ser los índices de la fila y la columna donde el borde fue detectado, o bien las coordenadas a otra resolución. Las coordenadas pueden venir dadas en el sistema de coordenadas de la imagen, pero más probablemente vendrán dadas en el sistema de coordenadas producido por el filtro de detección. El término borde se usa tanto para puntos como para fragmentos.

El conjunto de bordes que produce un detector de fronteras puede dividirse en dos clases: los correctos y los falsos. Un tercer conjunto corresponde a los bordes que deberían haber sido detectados. Este tipo de bordes reciben el nombre de falsos negativos mientras que los falsos corresponden a falsos positivos, La diferencia entre la unión de fronteras y el seguimiento de fronteras es que la unión tiene como entrada un conjunto desordenado de bordes producido por un detector de bordes y el resultado es una lista ordenada. El seguimiento de bordes toma como entrada una imagen y produce una lista ordenada de bordes, este proceso puede utilizar información global.

#### ◊ Pasos en la detección de Bordes

Los algoritmos para la detección de bordes tienen tres pasos:

1. Filtrado. Puesto que el cálculo del gradiente basado en los valores de intensidad de un número finito de puntos es muy susceptible al ruido, normalmente se filtra la imagen inicialmente para mejorar el rendimiento del detector con respecto al ruido. Es obvio que este proceso ha de aplicarse con cuidado, puesto que el suavizamiento oculta los bordes.
2. Realce. Este paso es esencial para determinar cambios en la intensidad de los vecinos de un punto. El proceso es básicamente el cálculo del gradiente.
3. Detección. Muchos puntos en una imagen tienen un gradiente no nulo. Obviamente no todos ellos son bordes para todas las aplicaciones. Es necesario algún método para determinar qué puntos son bordes. Un criterio ampliamente usado es la umbralización.
4. Estimación. Esta deberá llevarse a cabo en la resolución apropiada y en algunos casos será necesario también estimar la dirección del gradiente.

### 3.6 METODOS DE GRADIENTE

La detección de bordes es esencialmente una operación de detección de cambios de intensidad locales significativos. En una dimensión estos cambios, los de salto, están asociados con un pico local de la primera derivada. Su equivalente en el caso bidimensional son los Operadores Isotrópicos. [11]

◊ **Operadores Isotrópicos.** Un operador isotrópico tienen como objetivo determinar la intensidad de un borde con independencia de su dirección en la imagen. Es obvio que esta propiedad es deseable entre los detectores de bordes ya que no sería bueno que un detector sólo fuera sensible a bordes en determinadas

direcciones. Este tipo de detectores es aplicado a cada punto de la imagen para calcular la intensidad del borde independientemente de su orientación.

Consideremos la imagen de intensidades  $f(x,y)$  con coordenadas  $x$ - $y$ . Supongamos que la rotamos, un punto  $(x,y)$  y su rotación por  $\theta$  que nos produce,

$$\begin{aligned}x' &= x \cos \theta + y \operatorname{sen} \theta \\y' &= -x \operatorname{sen} \theta + y \cos \theta\end{aligned}$$

[Ecu. 3. 35]

las derivadas de los dos sistemas se relacionan mediante,

$$\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x} \cos \theta + \frac{\partial f}{\partial y} \operatorname{sen} \theta$$

[Ecu. 3. 36]

y

$$\frac{\partial f}{\partial y'} = -\frac{\partial f}{\partial x} \operatorname{sen} \theta + \frac{\partial f}{\partial y} \cos \theta$$

[Ecu. 3. 37]

Si uno está interesado en operadores isotrópicos, entonces un análisis matemático simple prueba que derivadas de orden par o de orden impar elevadas a potencias pares pueden dar lugar a operadores isotrópicos. Puede probarse que un operador isotrópico es la suma de los cuadrados de las derivadas parciales en las direcciones  $x$ ,  $y$ ; ya que,

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

[Ecu. 3. 38]

Para encontrar la dirección en la que se produce el cambio, la dirección del borde, consideremos la ecuación para la derivada parcial en la dirección  $x'$  [Ecu. 3.39]. Podemos entonces obtener  $\theta$  para el que esta derivada parcial obtiene el máximo cambio. Para ello tendremos que resolver ,

$$\frac{d\left(\frac{\partial f}{\partial x'}\right)}{d\theta} = 0$$

[Ecu. 3. 39]

de donde,

$$\theta = \operatorname{arctag} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$$

[Ecu. 3. 40]

Sustituyendo  $\theta$  en la fórmula de la derivada parcial tendremos que el máximo cambio vale,

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

[Ecu. 3. 41]

Las expresiones anteriores tienen operaciones de raíces cuadradas y potencias cuadradas que pueden ser computacionalmente lentos. Por ello, a veces se realizan aproximaciones de estas funciones que son menos tediosas.

### 3.6.1 Operadores de Segunda Derivada.

Los detectores de fronteras que hemos discutido calculan el gradiente y si su valor está por encima de un umbral, se supone la existencia de un punto de borde, pero también implica tener como resultado demasiados puntos de borde falsos. Una mejor aproximación sería encontrar sólo los puntos que tienen máximos locales en el gradiente y considerar a éstos los puntos de borde. Esto significa que en los puntos de borde existe un pico en la primera derivada y equivalentemente a un cruce por cero en la segunda. Así pues los puntos de borde se detectan encontrando los cruces por cero de la segunda derivada en las intensidades de la imagen. Existen dos operadores bidimensionales que corresponden a la segunda derivada: el Laplaciano y la Segunda Derivada Direccional.

◇ **Operador Laplaciano.** El laplaciano es el equivalente bidimensional de la segunda derivada. La fórmula para el laplaciano de una función  $f(x,y)$  es:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

[Ecu. 3. 42]

La segunda derivada en las direcciones  $x, y$ ; se aproximan utilizando las ecuaciones en diferencias,

$$\begin{aligned}
 \frac{\partial^2 f}{\partial x^2} &= \frac{Gx}{\partial x} \\
 &= \frac{\partial (f(i+1, j) - f(i, j))}{\partial x} \\
 &= f(i+2, j) - 2f(i+1, j) + f(i, j)
 \end{aligned}$$

[Ecu. 3.43]

Análogamente obtendríamos,

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+2) - 2f(i, j+1) + f(i, j)$$

[Ecu. 3.44]

Sin embargo, estas dos aproximaciones no están centradas en la posición (i,j) de modo que una aproximación más razonable, sería:

$$\begin{aligned}
 \frac{\partial^2 f}{\partial x^2} &= f(i+1, j) - 2f(i, j) + f(i-1, j) \\
 \frac{\partial^2 f}{\partial y^2} &= f(i, j+2) - 2f(i, j+1) + f(i, j)
 \end{aligned}$$

[Ecu. 3.45]

Combinando estas dos ecuaciones [Ecu. 3.45], en un único operador, podemos usar la siguiente máscara para aproximar el laplaciano.

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Algunas veces es deseable darle más peso a los pixeles centrales del entorno. Otra aproximación del Laplaciano es:

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

El laplaciano se puede definir en términos de z de la siguiente manera:

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

[Ecu. 3.46]

El operador laplaciano marca la presencia de un borde cuando la salida del operador realiza una transición por cero. Los ceros triviales, es decir, las regiones de cero uniforme tienen que ser ignoradas. En principio, las localizaciones de los cruces por cero pueden ser estimadas a nivel de subpíxeles usando interpolación lineal; pero los resultados pueden ser inexactos.

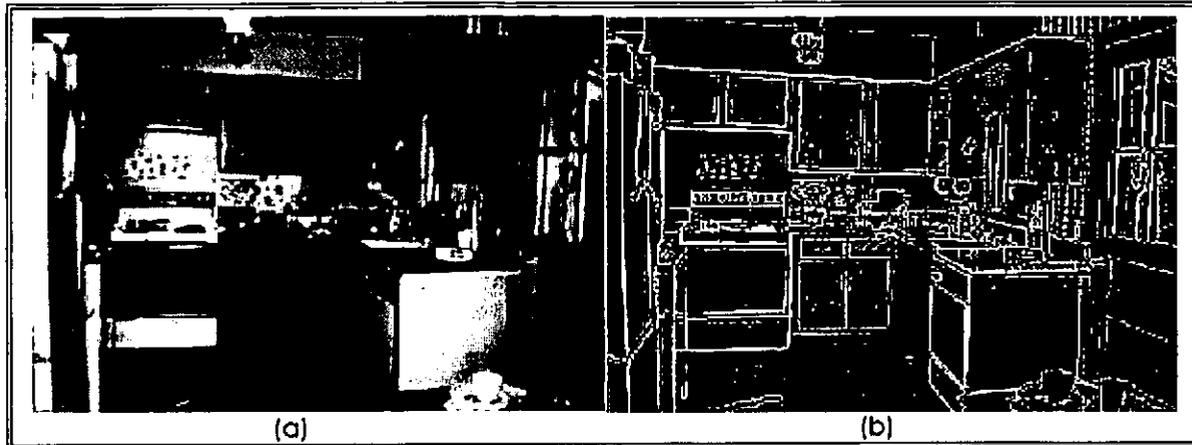


Figura 3. 15 (a) imagen original y (b) imagen al aplicar el laplaciano [A]

Consideremos el ejemplo que se muestra en la figura 3.15. Esta figura muestra el resultado de aplicar el laplaciano a una imagen de un paso salto simple. En este ejemplo, el cruce por cero que corresponde al borde en la imagen, está a mitad del camino de los dos píxeles con valores no nulos del laplaciano, en este caso los bordes deberían ser marcados o bien a la izquierda o la derecha, pero de una forma consistente en toda la imagen.

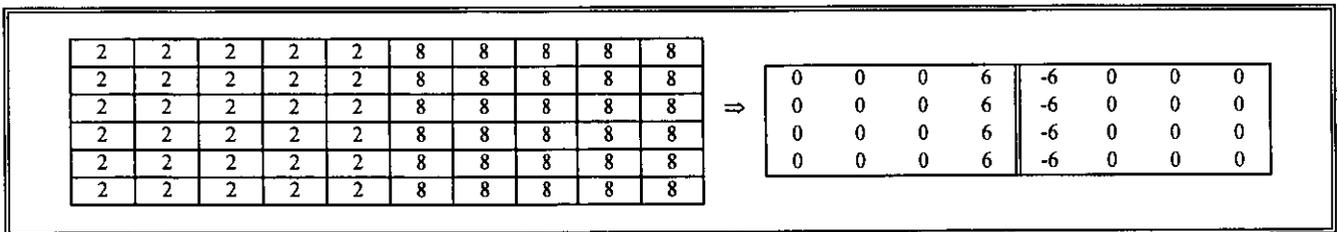


Figura 3. 16 Salto y resultado de aplicar el Laplaciano

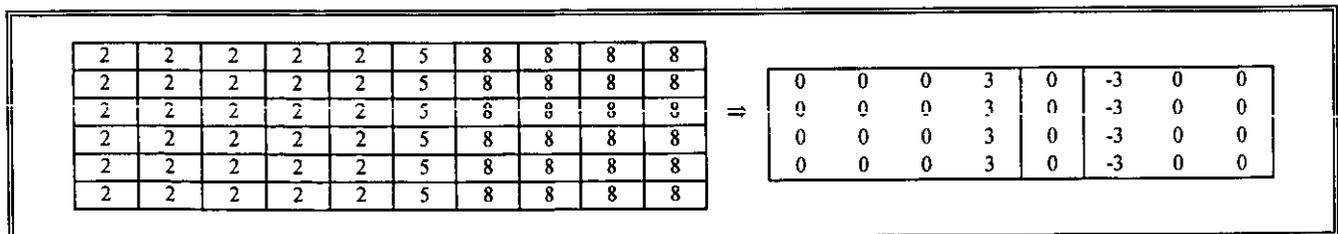


Figura 3. 17 Imagen con un salto y resultado de aplicar el Laplaciano

◊ **Cruces por cero de la segunda Derivada.** Una técnica de detección de bordes basada en los cruces por cero de la segunda derivada, explora el hecho de que un borde de salto corresponde a un cambio brusco de la función de la imagen. La primera derivada de la función de la imagen debe tener un extremo en la posición correspondiente a un borde en la imagen, y la segunda derivada debe ser cero en la misma posición. Sin embargo, es mucho más fácil y preciso encontrar un cruce por cero que un extremo. En la figura 3.18 ilustra el problema en el caso unidimensional.

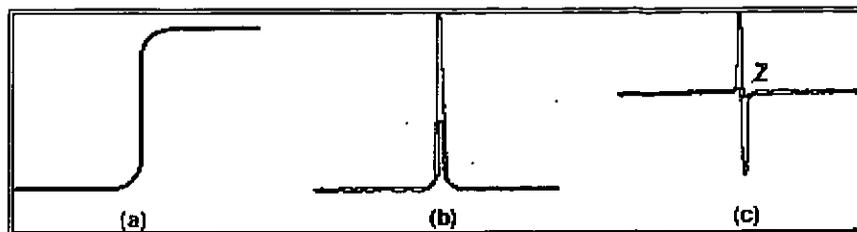


Figura 3. 18 (a) Intensidad, (b) Derivada Primera (c) Derivada segunda

El punto crucial es cómo calcular la segunda derivada de una forma robusta, una posibilidad es suavizar la imagen primero para reducir ruido y después calcular la segunda derivada. Cuando se selecciona un filtro para suavizar se deben cumplir dos criterios: debe ser suave y de banda limitada para reducir el número de frecuencias en las que puede producirse el cambio y también que la respuesta sea casi puntual. Aunque estos dos criterios entran en conflicto, la distribución que optimiza simultáneamente ambos criterios es la distribución gaussiana.

El filtro de suavizamiento gaussiano,  $G(x,y)$  viene dado por:

$$G(x, y) = c \cdot e^{\left[-\frac{x^2+y^2}{2\sigma^2}\right]}$$

[Ecu. 3. 47]

donde  $x, y$  son las coordenadas de la imagen,  $\sigma$  es la desviación típica de la distribución de probabilidad y  $c$  es una constante de normalización. el objetivo es obtener la segunda derivada de una versión suave de la imagen original  $f(x,y)$ . El operador laplaciano  $\nabla^2$  produce la segunda derivada y es, además, isotrópico. Consideremos entonces el laplaciano de una imagen  $f(x,y)$  suavizada por una distribución gaussiana. Tendremos entonces que calcular,

$$\nabla^2(G(x, y, \sigma) * f(x, y))$$

[Ecu. 3. 48]

El orden en el que se realiza la convolución y diferenciación puede ser intercambiado debido a la linealidad de los operadores y tendremos:

$$(\nabla^2 * G(x, y, \sigma)) * f(x, y)$$

[Ecu. 3. 49]

si notamos el filtro de convolución por  $h(x,y,\sigma)$  tendremos que:

$$h(x, y, \sigma) = \frac{c}{\sigma^2} \left( \frac{x^2 + y^2}{\sigma^2} - 2 \right) \cdot e^{\left[ -\frac{x^2 + y^2}{2\sigma^2} \right]}$$

[Ecu. 3. 50]

que recibe el nombre de filtro en forma de sombrero charro.

Este proceso de búsqueda de la segunda derivada es muy robusto, el suavizamiento gaussiano suprime la influencia de pixeles lejanos y el laplaciano es una medida eficiente y estable de los cambios en una imagen. La ventaja de esta aproximación comparada con los clásicos operadores de bordes es que una mayor área que rodea el píxel se toma en cuenta y la influencia de los pixeles distantes está determinada por  $\sigma$ .

Esta aproximación al cruce por cero tiene también inconvenientes, suaviza mucho la forma, además de crear contornos cerrados en forma de espagueti.

### 3.7 DETECTORES DE BORDES GAUSIANOS

La idea esencial en la detección de bordes de salto es encontrar puntos en la imagen que tengan localmente máximo gradiente, muchas de las investigaciones en la detección de bordes está dedicada a la aproximación numérica del gradiente en imágenes. Los saltos en las imágenes reales no son abruptos ya que éstos son suavizados por los filtros paso bajo inherentes a los procesos de captación de imágenes. Análogamente las imágenes son distorsionadas por el ruido. Por tanto, cualquier proceso de detección de bordes basado en fronteras debe por una parte suprimir el ruido y por otra localizar los bordes lo más preciso posible. Si consideramos un operador que sea la combinación de un filtro gaussiano con una aproximación de gradiente tendremos un operador que es sensible al borde en la dirección de máximo cambio. Dentro de este modelo se encuadra el detector de Canny.

◊ **El detector de Bordes de Canny.** El detector de bordes de Canny es la primera derivada de una gaussiana y aproxima muy bien el operador que optimiza el producto del cociente señal ruido y localización. El algoritmo se resume como sigue:

Sea  $f(x,y)$  la imagen. El resultado de convolucionar la imagen con un filtro gaussiano es la imagen suavizada,

$$s(x, y) = G(x, y, \sigma) * f(x, y)$$

[Ecu. 3. 51]

donde  $\sigma$  es la anchura de la función gaussiana y controla el grado de suavización. A continuación se calcula el gradiente de la imagen suavizada usando una aproximación 2x2 de las primeras diferencias y producen las matrices  $P(i,j)$  y  $Q(i,j)$  para las derivadas parciales en  $x$ ,  $y$

$$P(i, j) \approx (s(i+1, j) - s(i, j) + s(i+1, j+1) - s(i, j+1)) / 2$$

$$Q(i, j) \approx (s(i, j) - s(i, j-1) + s(i+1, j) - s(i+1, j-1)) / 2$$

[Ecu. 3. 52]

Las diferencias finitas son promediadas sobre un cuadrado de 2x2 de tal forma que las derivadas parciales se calculan en el mismo punto de la imagen. La magnitud y orientación del gradiente se puede calcular mediante

$$M(i, j) = \sqrt{P(i, j)^2 + Q(i, j)^2}$$

$$(i, j) = \arctang(Q(i, j), P(i, j))$$

[Ecu. 3. 53]

donde la función arcotangente toma dos argumentos y genera un ángulo sobre el círculo completo de direcciones posibles.

La matriz de magnitudes  $M(i,j)$  tendrá valores grandes donde el gradiente sea grande, pero esto no es suficiente para identificar los bordes ya que el problema de encontrar los puntos donde hay un cambio rápido de la intensidad se ha transformado en la búsqueda de los puntos donde existe un máximo local de  $M(i,j)$ . Para identificar los bordes, las crestas en la matriz de magnitudes debe reducirse de forma que las magnitudes donde existe un máximo local sean las únicas que persistan. Dos son las aproximaciones que vamos a examinar para reducir las crestas. Una vez calculada la magnitud del gradiente y su dirección, estos valores son examinados en la imagen completa junto con la magnitud de los bordes de los dos pixeles vecinos en la dirección perpendicular al borde. Si al menos uno de estos pixeles tiene magnitud superior que la del pixel central, a este último se le asigna el valor cero, es decir,  $M(i,j)=0$ .

Este proceso hace más estrecha las crestas. Los valores de las alturas de las crestas se retienen en la magnitud de no supresión que es denotada por  $N(i,j)$ . Observemos que  $N(i,j)$  vale  $M(i,j)$  o cero si no lo hemos localizado como máximo local. Los valores no nulos en  $N(i,j)$  corresponden a la cantidad de contraste en un salto en la imagen.

A pesar de la suavización realizada previamente, esta nueva matriz contendrá muchos falsos bordes. Un procedimiento típico que se usa para reducir el número de fragmentos de falsos bordes es aplicar un umbral a  $N(i,j)$ , todos los valores por debajo del umbral se cambian a cero. Todavía existirán algunos bordes falsos por ser el umbral muy bajo  $\tau$  (falsos positivos) y partes del contorno no aparecerán (falsos negativos). Para resolver este problema se usa un umbral doble. A  $N(i,j)$  le aplicamos dos umbrales  $\tau_1$  y  $\tau_2$  con  $2\tau \approx \tau_1$  y tenemos dos imágenes umbralizadas  $T_1 = (i,j)$  y  $T_2 = (i,j)$ , puesto que la imagen  $T_2$  se formó con un umbral superior, ésta tendrá menor número de falsos positivos, pero también tendrá numerosos huecos en los contornos (falsos negativos).

El algoritmo de doble umbral une los bordes de  $T_2$  para formar contornos. Cuando llega al final de un contorno, el algoritmo mira a  $T_1$  y a los 8 pixeles vecinos que pueden ser unidos al contorno. El algoritmo continúa utilizando pixeles de  $T_1$  hasta llegar a otro pixel en  $T_2$ .

### 3.8 TRANSFORMADA DE HOUGH

La técnica conocida como transformada de Hough es capaz de agrupar puntos pendiente aislados para formar estructuras. Considerando un punto aislado  $(x,y)$ , existen un número infinito de líneas que pasan por ese punto. Cada una de esas líneas puede caracterizarse como una solución particular de la ecuación [Ecu. 3.54], que define una línea dada por,

$$y = mx + b$$

[Ecu. 3. 54]

donde  $m$  es la pendiente de la línea y  $b$  es el corte por el eje  $y$ . Cualquier línea puede caracterizarse por estos dos parámetros. Se pueden caracterizar las líneas que pasan por el punto  $(x,y)$  a través de sus valores de pendiente ( $m$ ) y cruce por el eje "y" ( $b$ ) en un espacio pendiente-cruce. De hecho, todas las líneas que pasan por el punto  $(x,y)$  tiene una relación entre  $b$  y  $m$  dada por  $b = -mx + y$ , de esta ecuación se puede deducir que el conjunto de líneas que pasan por el punto  $(x,y)$  forman una recta en el espacio pendiente-cruce. Cada punto del espacio imagen  $(x,y)$  corresponde a una línea en el espacio de parámetros  $(m,b)$  y cada punto en el espacio  $(m,b)$  corresponde a una línea en el espacio  $(x,y)$ .

En la transformada de Hough cada punto pendiente  $(x,y)$  es viable en el espacio  $(m,b)$  para cada posible línea que pase por él. Se enumeran las posibilidades en un acumulador, si un punto del espacio  $(m,b)$  tiene más de una posibilidad quiere decir que hay más de un punto característico en esa línea.

El algoritmo de la transformada de Hough puede expresarse en los siguientes pasos:

- a) Localizar todos los puntos característicos presentes en la imagen.
- b) Identificar cada punto característico.
- c) Incrementar cada punto del acumulador que pase por dicho punto característico.
- d) Localizar máximos locales en el acumulador.
- e) Transformar cada máximo local del acumulador al espacio imagen.

La representación de líneas mediante el espacio pendiente-cruce, tiene un serio problema con las líneas verticales: tanto  $m$  como  $b$  tiene valor infinito. Otra vía para expresar una línea es hacerlo a través de la expresión,

$$x \cos \theta + y \sin \theta = \rho$$

[Ecu. 3. 55]

con lo cual puede ser parametrizada en el espacio  $(\theta,\rho)$ , para interpretar su significado suponer que se traza una perpendicular desde el origen a la recta,  $\theta$  es el ángulo de la perpendicular con el eje "x" y  $\rho$  es la longitud de la línea perpendicular. En imágenes,  $\theta$  está limitada al intervalo  $[0,2\pi]$  y  $\rho$  a la diagonal de la imagen, en este caso cada punto característico elige una senoide en el acumulador.

Resulta sencillo extender la transformada de Hough a cualquier otro tipo de contorno que pueda ser parametrizado. Por ejemplo, un círculo puede parametrizarse a través de la posición del centro y su radio  $(x,y,r)$ . Si un punto eje pertenece a un círculo existen un conjunto de puntos centros y radios que pueden ser el origen del círculo, estos estarán distribuidos sobre círculos concéntricos. La figura de comparación puede ser un cilindro en un espacio tridimensional.

Se puede extender las transformada de Hough para representar contornos tan complejos como se desee, siempre y cuando estos puedan ser parametrizables, el número de parámetros necesarios para describir el contorno condicionará la dimensión del acumulador en el espacio de comparación. Algunos contornos no son fácilmente representables utilizando un conjunto pequeño de parámetros. En este caso se deben indicar explícitamente la lista de puntos que forman dicho contorno, suponer que se dispone de una lista de todos los píxeles del contorno objetivo, se puede almacenar la posición relativa de los puntos del contorno objetivo respecto a los puntos de referencia del contorno.

El algoritmo para construir la transformada generalizada de Hough puede expresarse en los siguientes términos:

- a) Localizar todos los puntos característicos presentes en la imagen.
- b) Identificar cada punto característico.
- c) Identificar cada punto del contorno objetivo.
- d) Tomar la posición relativa del punto de referencia a  $i$  (punto característico).
- e) Añadir ese desplazamiento a las posiciones de  $i$ .
- f) Incrementar el acumulador en esas posiciones.
- g) Localizar máximos locales en el acumulador.
- h) Transformar cada máximo local del acumulador al espacio imagen.

Usualmente el espacio de parámetros es discreto, la granularidad en la discretización del acumulador determina la calidad de los resultados obtenidos.

Debido al ruido inherente, a la discretización del espacio imagen y del acumulador y a factores inherentes al proceso, es conveniente introducir una pequeña tolerancia en la búsqueda de los contornos, esto se consigue haciendo que los distintos puntos característicos sean viables no sólo en un punto del acumulador sino que lo hagan también, aunque quizás con menor peso, en los puntos vecinos. Esto implica buscar en los contornos que pasan exactamente por el punto característico y en los que pasan relativamente cerca, esta técnica se puede implementar directamente en el proceso de elección o aplicarla al final mediante una corrección del acumulador.

Si se está buscando una forma concreta (por ejemplo, un círculo de radio fijo) sólo es necesario localizar el máximo absoluto del acumulador. Si se buscan contornos más complejos será necesario localizar todos los máximos relativos presentes en el acumulador, en este caso puede ser conveniente exigir que esos máximos

relativos tengan un número mínimo de posibilidades. Cuanto más preferencias tenga un máximo local es más probable que forme parte de un contorno.

En la transformada de Hough cada punto característico elige todos los puntos accesibles del acumulador. Este hecho puede provocar errores en la determinación posterior de máximos locales, una forma de evitar este problema es realizar una segunda etapa de elección en la cual cada punto característico sólo elige al que presenta un valor máximo entre los puntos a los que puede acceder. Esta segunda etapa de votación elimina el desorden innecesario y fuerza a que en el acumulador resultante aparezcan unos pocos puntos aislados.

### 3.9 BUSQUEDA DE GRAFOS

En este tipo de estrategias no se buscan agrupaciones de píxeles en líneas sino que se intenta determinar curvas del píxeles para localizar el contorno del objeto. Este tipo de proceso es más complejo pero, normalmente, permite obtener mejores resultados.

Un grafo  $G=(N,A)$  es un conjunto finito, no vacío, de nodos  $N$  junto con un conjunto  $A$  de pares de los elementos de  $N$ . Cada par  $(n_i, n_j)$  de  $A$  se denomina un arco (se dice que  $n_j$  es el sucesor de  $n_i$ ). El proceso de identificar los sucesores de un nodo se denomina expansión del nodo. Además, puede asignarse un coste  $c(n_i, n_j)$  a cada arco, una secuencia de nodos  $n_1, n_2, \dots, n_k$  en la cual cada nodo  $n_i$  es el sucesor del nodo  $n_{i-1}$  se denomina un camino de  $n_1$  a  $n_k$  y el costo de la ruta viene dado por:

$$c = \sum_{i=2}^k c(n_{i-1}, n_i)$$

[Ecu. 3. 56]

Una vía para realizar esta estrategia es considerar que en la imagen existe un grafo en el cual los nodos del grafo son puntos del contorno (parejas de píxeles entre las cuales pasa el borde) a detectar, y los arcos del grafo son la dirección hacia la cual se extiende el contorno. Si se usa el gradiente o cualquier otro operador de detección de bordes se puede asignar un determinado costo a cada arco en función de la respuesta del operador seleccionado en la pareja de puntos del nodo de destino, las posiciones por las cuales pasa el borde presentarán un valor pequeño de costo con lo cual el problema de buscar el contorno óptimo se reduce a encontrar el camino de coste mínimo en el grafo.

La resolución del problema de detección de bordes mediante la técnica de grafos se puede subdividir en cuatro subproblemas: Cómo formular o construir el grafo, como asignar una función de costo a un arco, cómo seleccionar los puntos de origen y final del grafo y cómo buscar la ruta de mínimo coste. [1]

- **Formulación del grafo:** Para formular un grafo es necesario tener algún conocimiento sobre el contorno que se desea localizar. No es necesario un conocimiento muy explícito (si se tuviera, probablemente no sería necesario emplear una técnica de este tipo) pero si algo que de una pista sobre como plantearlo.

**Ejemplo:** Considerar el caso de una imagen en la cual se conoce que el contorno existente comienza en la parte superior y termina en la inferior. El grafo puede comenzar con un nodo que represente la parte superior de la imagen. Los ejes que salen de ese nodo van hacia las posibles posiciones por los cuales puede entrar el contorno en la imagen. A su vez, para cada uno de esos nodos pueden existir tres ejes que indican

las posiciones por los cuales continúa el contorno hacia abajo, hacia la derecha o hacia la izquierda. Esta situación se repite hasta que por la parte de abajo de la imagen se llega al nodo final. Ver figura 3.20.

- **Funciones de costo:** Las funciones de costo suelen construirse en términos de operadores que calculan la magnitud del gradiente, la orientación del gradiente, los cruces por cero del Laplaciano, y en general, con base en resultados proporcionados por operadores de determinación de ejes. La función de costo no tiene porqué construirse sólo con términos, denominados externos, basados en los datos de las imágenes sino que también puede estar formada por términos, denominados internos, basados en propiedades impuestas a los contornos que se buscan, como su suavidad, grado de irregularidad, etc. Ejemplo: Una posible solución para el costo de un eje que pasa entre dos píxeles  $p$  y  $q$  es:

$$c(p, q) = f_{max} - [f(p) - f(q)]$$

[Ecu. 3. 57]

donde  $f_{max}$  es el valor máximo de intensidad y  $f(p)$  y  $f(q)$  son los niveles de intensidad de los dos píxeles que forman el nodo destino.

- **Selección de puntos origen y fin:** Existen tres puntos para elegir el inicio, y tres para elegir el fin del grafo: usar algunos puntos especiales conocidos (el punto de máxima respuesta de gradiente, etc.), usar puntos con propiedades geométricas especiales (esquinas de la imagen, etc.) o simplemente, preguntar al usuario. La selección de los puntos de origen y fin del grafo es fundamental en los resultados que se van a obtener. En un caso simple se pueden escoger como inicio la parte superior y como final la parte inferior de la imagen.

**Búsqueda de caminos de mínimo:** La búsqueda de caminos de mínimo es un tópico muy estudiado en la literatura de este campo. Algunas vías pueden ser búsqueda exhaustiva, se estudian todos los caminos posibles sobre el grafo completo, algoritmos Greedy (los algoritmos Greedy buscan el mejor en cada paso y construyen el contorno uniendo pixel a pixel el mejor candidato en cada paso), algoritmos de expansión, la idea es hacer crecer la búsqueda para añadir las mejores opciones a cada camino, algoritmos de expansión con algoritmos heurísticos de poda, se utilizan reglas heurísticas para eliminar aquellas ramas que van a conducir a caminos poco productivos y programación dinámica, se utilizan reglas heurísticas para eliminar aquellas ramas que van a conducir a caminos poco productivos, todas estas vías involucran después una etapa de optimización.

	0	1	2
0	• (7)	• (2)	• (2)
1	• (5)	• (7)	• (2)
2	• (5)	• (1)	• (0)

Figura 3. 19 Imagen 3x3 [B]

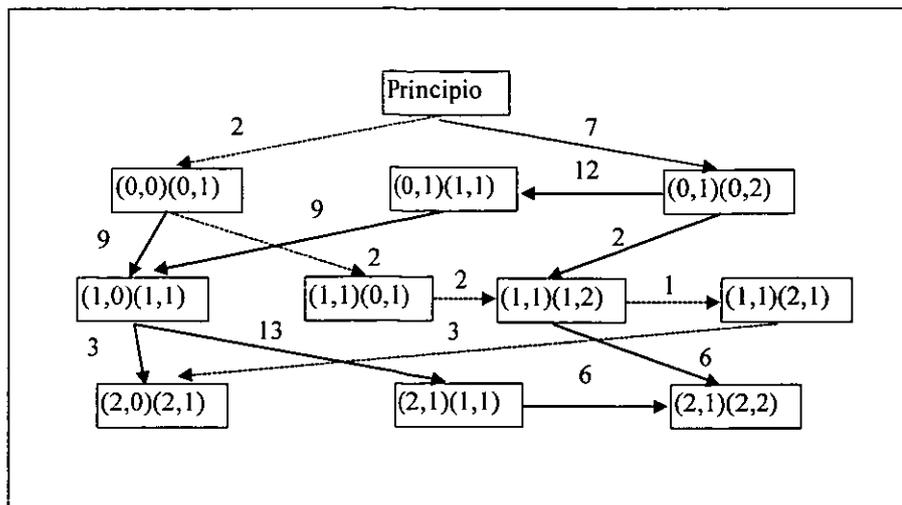


Figura 3. 20 [B]

En la figura 3.20 muestra el grafo para encontrar un borde de la figura 3.19. El par (a,b) (c,d) de cada cuadro hace referencia a los puntos p y q, respectivamente. Se supone que p está a la derecha del camino, así como se recorre la imagen de arriba abajo. Las líneas punteadas indican la ruta de coste mínimo.

### 3.10 UMBRALIZACION

En la umbralización de una región, la segmentación se realiza identificando regiones como zonas con propiedades comunes, la propiedad más simple que posee un pixel en una región es su intensidad. La vía más natural para segmentar las regiones es la umbralización, en la cual se separan zonas claras de zonas oscuras. La umbralización genera imágenes binarias poniendo a “1” los pixeles con valores mayores que el umbral y a “0” los pixeles con valores menores.

Si  $g(x,y)$  es una versión umbralizada de  $f(x,y)$  para un determinado umbral  $t$ ,

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) \geq T \\ 0 & \text{en otro caso} \end{cases}$$

[Ecu. 3. 58]

El mayor problema con la umbralización surge por el hecho de considerar sólo la intensidad y no incluir las relaciones entre pixeles. No existe ninguna garantía de que pixeles identificados en el proceso de umbralización como pertenecientes a una región sean contiguos. Se pueden encontrar fácilmente puntos

aislados, en especial en las proximidades de los bordes de la región, que han surgido por el ruido que está presente en la imagen. Cuando se usa la umbralización es necesario tomar en cuenta estas implicaciones.

Otro problema que aparece al utilizar la umbralización global es que ésta es sensible a los cambios de iluminación locales de la imagen, una forma sencilla de eliminar esta dependencia de la iluminación es realizar una umbralización local de la imagen en la cual el valor del umbral  $T$  varía en función de la posición espacial que se esté analizando en cada caso. Para calcular el umbral global o para adaptar el umbral local a una determinada área, usualmente se utiliza el histograma si se pueden localizar dos o más picos distintos. Un histograma (normalizado al tamaño de la imagen) no es más que una distribución de probabilidad,

$$p(z) = \frac{n_z}{n}$$

[Ecu. 3. 59]

donde  $n_z$  es el número de píxeles que tienen una intensidad  $z$  y  $n$  es el número total de píxeles de la imagen o de la región a analizar.

Hay cuatro vías para resolver el problema:

i) Localización de picos y valles: Una solución extremadamente fácil es localizar los puntos de umbralización mediante la localización de los valles (mínimos) existentes entre los picos.

ii) Clusterización: En una imagen pueden existir varios grupos de píxeles con diferente intervalo de iluminación, es difícil localizar el punto de umbralización, pues estos intervalos pueden estar ocultos. En este caso la solución proviene de minimizar el error de clasificación entre grupos de píxeles, dado que no se dispone de los histogramas individuales de cada región habrá que intentar buscar un valor del umbral en el histograma combinado que minimice el área de ocultamiento entre grupos.

La primera opción es considerar los valores en las dos regiones como dos clusters, considerando los clusters tan excluyentes como sea posible para poder minimizar su ocultamiento, después se pueden cambiar las distribuciones de píxeles en los clusters ajustando el punto de separación (el umbral). Se puede cambiar el umbral incrementando la extensión de un cluster y disminuyendo la de otro, el objetivo es seleccionar el umbral que minimiza la extensión combinada.

Se define la varianza interior de los grupos de píxeles como la suma ponderada de la varianza de cada clase

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

[Ecu. 3. 60]

donde:

$$q_1(t) = \sum_{i=0}^{t-1} p(i)$$

$$q_2(t) = \sum_{i=t}^{N-1} p(i)$$

$\sigma_1^2(t)$  varianza de los pixeles del cluster 1

$\sigma_2^2(t)$  varianza de los pixeles del cluster 2

N es el número de niveles de intensidad y  $p(i)$  es el valor del histograma en el nivel  $i$ .

Calculando la varianza interior de los grupos de pixeles, para cada pareja de los grupos provenientes de cada posible umbral implica un alto procesamiento de información. Sin embargo, si se resta la varianza interior de los grupos de la varianza total se obtiene la varianza entre grupos:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = q_1(t)[\mu_1(t) - \mu]^2 + q_2(t)[\mu_2(t) - \mu]^2$$

[Ecu. 3. 61]

donde  $\sigma$  es la varianza combinada y  $\mu$  es la media combinada. La varianza entre grupos es la suma ponderada de medias. Substituyendo  $\mu = q_1\mu_1 + q_2\mu_2$  y simplificando se obtiene la expresión final de la varianza entre grupos:

$$\sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2$$

[Ecu. 3. 62]

El umbral óptimo es aquel que permite maximizar la varianza entre grupos, es decir, aquel que minimiza la varianza interior de los grupos, en principio es necesario recalculan  $q_i(t)$  y la media para cada umbral pero este problema computacional se puede disminuir calculando los valores recursivamente conforme  $t$  se incrementa. [1]

iii) Modelo mixto: Otra forma de realizar la umbralización es asumir que cada grupo de pixeles responde a una distribución Gaussiana, cada grupo presente en la distribución tiene una media y una varianza independiente del umbral que se busca. Si el modelo de Clusterización separaba los clusters en función del umbral y optimizaba medidas estadísticas, el modelo mixto asume que existen las distribuciones e intenta localizarlas. Una vez localizadas, determina el mejor umbral, este método implica que hay seis parámetros desconocidos ( $q_1, q_2, \mu_1, \mu_2, \sigma_1, \sigma_2$ ) y se intenta estimarlos. Normalmente se realiza una búsqueda de dos distribuciones Gaussianas que aproximen el histograma conjunto de las dos distribuciones; esta búsqueda se puede realizar de forma exhaustiva o por métodos de descenso de error.

iv) Umbralización mediante contornos: Una forma de guiar la umbralización mediante contornos es utilizar algún método de búsqueda de contornos (por ejemplo, detección de puntos eje) y utilizar en la distribución únicamente aquellos puntos que tienen una alta probabilidad de pertenecer a un contorno. En la práctica se suelen considerar los que tienen una alta probabilidad de pertenecer al contorno y los que están en su proximidad.

◊ **Umbralización de la magnitud del gradiente.** Un paso que se realiza habitualmente en la detección de bordes, tras la aplicación del detector correspondiente, es la umbralización de la magnitud del gradiente,

$$E(x, y) = \begin{cases} 1 & \text{si } \|\nabla f(x, y)\| > T \\ 0 & \text{en otro caso} \end{cases}$$

[Ecu. 3. 63]

El conjunto:

$$\{ (x, y) : E(x, y) = 1 \}$$

[Ecu. 3. 64]

es el conjunto de pixeles del borde.

## II COMENTARIO AL CAPITULO II

Al estudiar los procedimientos de mejora en imágenes digitales, nos ofreció el primer acercamiento a la manipulación de pixeles en forma global de una imagen de alta calidad visual. Los procesos de mejoramiento de la imagen adecuan considerablemente el aspecto de la imagen que ha sido transformada de la realidad a su formato digital, La existencia de un número considerable de procesos es reflejo de la diversidad de aplicación o interés que pueden ser generados por medio de estos métodos, existen infinidad de rasgos que pueden ser mejorados en una imagen, la nitidez, el contraste, eliminación de ruido, etc. Los procesos en el dominio espacial ofrecen una gran variedad de herramientas para el análisis de imágenes como el histograma y la densidad de distribución estadística. Los procesos de filtrado bajo su característica global, tienen la capacidad de poder extraer rasgos importantes como son los contornos de objetos contenidos en la escena de la imagen; eliminan el ruido en el caso de filtros suavizantes, evitando errores y suavizando defectos, así la imagen es más viable para alguna aplicación posterior de procesamiento.

En el capítulo la segmentación está enfocada a trabajar en las fronteras, es decir, al realce de bordes y contornos. Cabe hacer la siguiente reflexión, al extraer los bordes en una imagen se elimina una gran parte de la información contenida en la imagen. Sin embargo, la imagen no pierde su comprensión visual, ya que solo se pierde la textura que rellena los objetos conservando su forma, por esto podemos concluir que un objeto puede ser comprensiblemente bien representado en base a líneas que conforman su contorno. En esta sección se abordan varios métodos globales que presentan buenos resultados en la extracción de bordes, aunque realmente dependerá de la complejidad de la imagen que se esté procesando y el cómo se haya adecuado en el proceso de mejora, en una imagen con un intervalo dinámico grande será más difícil de identificar saltos rápidos en la intensidad de los pixeles que la forman, por ello siempre se busca que tengan un alto grado de contraste para identificar eficientemente los saltos de intensidad que con seguridad forman parte de un borde.

En el siguiente capítulo tendrá como objetivo, una vez obtenida la información que representa los bordes en la imagen, el codificarla en arreglos lógicos para que estos rasgos puedan tener una interpretación analítica y no solo visual.

· C · A · P · I · T · U · L · O ·

IV

REPRESENTACION Y  
DESCRIPCION

Después de segmentar una imagen, el conjunto de píxeles segmentados se representan y describen normalmente en forma adecuada para su posterior procesamiento en computadora, los contornos se pueden describir en sus principales características tales como su longitud, la orientación de una línea recta que une puntos extremos, y el número de concavidades. Las características seleccionadas como descriptores deberían ser tan insensibles como fuera posible a variaciones tales como cambios de tamaño, traslación y rotación.

En general, los descriptores son un conjunto de números producidos para describir la forma de un objeto o el contenido de una región. Es posible que la forma del objeto no se pueda reconstruir completamente a partir de los descriptores, pero estos deben ser suficientemente adecuados para permitir discernir objetos con formas o contenidos diferentes.

El reconocimiento de un objeto es una de las tareas fundamentales en el área de visión artificial, el reconocimiento se basa en niveles de gris o colores y en las características dimensionales de los objetos. La meta del reconocimiento de un objeto es encontrar la suficiente información para distinguirlo entre diversos objetos. El diseño de los objetos potencialmente válidos es basado principalmente en su composición de sus elementos geométricos, las principales dificultades se presenta al adquirir la imagen, una mala adquisición creará deformaciones geométricas en el objeto, existen varias técnicas que se encargarán de corregir dichos defectos.

Hay varios métodos para el análisis de la dimensión de una variable, estos objetos se pueden dividir en dos categorías, los métodos basados en área y los métodos basados en contornos. Estos últimos tienen un gran interés en este trabajo. Los métodos basados en contorno incluyen las siguientes técnicas, los descriptores simples, por ejemplo existen la longitud del perímetro, curvatura y energía de flexión, se ha utilizado el concepto de Fourier para representar contornos de acuerdo con una serie de coeficientes. Existen muchos métodos para la representación de objetos con una versatilidad para adecuarse a distintas aplicaciones.

Es difícil decir como debe ser un buen descriptor. En general, el descriptor perfecto es aquel que aporta diferencias grandes para objetos con formas o contenidos significativamente diferentes y diferencias muy pequeñas para objetos con formas o contenidos similares. Sin embargo, ni siquiera existe una definición aceptada universalmente de similaridad. Este es un concepto muy propio de la intuición humana.

## 4.1 ESQUEMAS DE REPRESENTACION

### 4.1.1 Códigos de cadena

Los códigos de cadena se utilizan para representar un contorno por medio de una sucesión conexas de segmentos de longitud y dirección especificadas. Normalmente esta representación se basa en segmentos de conectividad 4 u 8, la dirección de cada segmento se codifica utilizando un esquema de numeración. Ver figura 4.1.

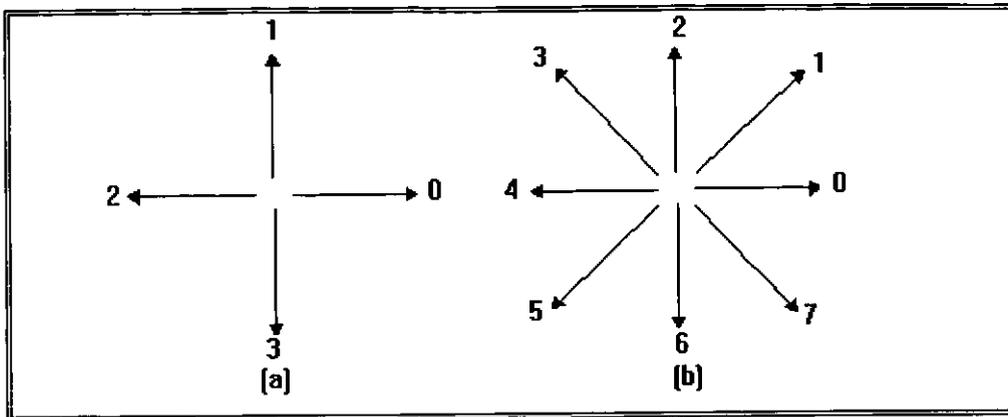


Figura 4. 1 Direcciones de: (a) código de cadena de 4 direcciones y (b) código de cadena de 8 direcciones [B]

Se puede generar un código de cadena siguiendo un contorno, por ejemplo, en el sentido de las manecillas del reloj y asignando una dirección a los segmentos que conectan cada par de píxeles. Este método es generalmente inaceptable por dos principales razones:

- 1) La cadena de códigos resultante es normalmente bastante grande
- 2) Cualquier pequeña perturbación a lo largo del contorno debida al ruido o a una perturbación a lo largo del contorno debida al ruido o a una segmentación imperfecta origina cambios en el código, que pueden no estar necesariamente relacionados con la forma del contorno

Una solución consiste en volver a hacer un muestreo el conjunto seleccionado un espacio de cuadrícula mayor

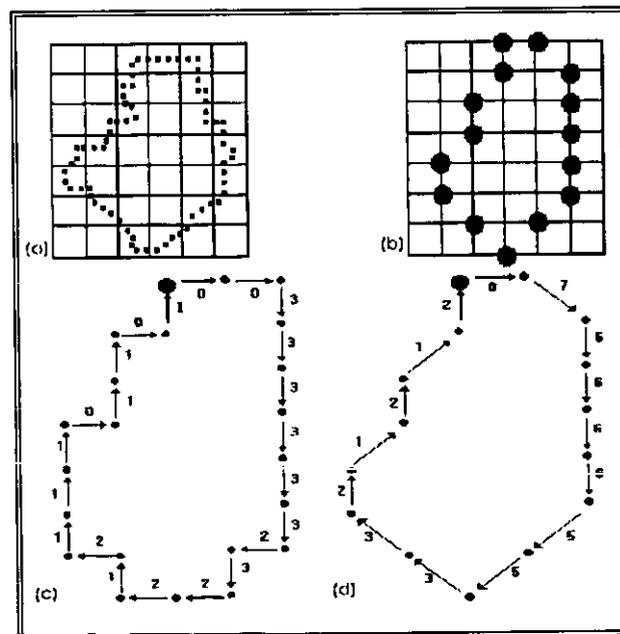


Figura 4. 2(a) Contorno digital con el cuadrículado de remuestreo superpuesto; (b) resultado de remuestreo; (c) código de cadena de 4 direcciones; (d) código de cadena de 8 direcciones. [B]

El código de cadena depende del punto de partida, pero dicho código se puede normalizar por un procedimiento directo: dado como un código de cadena generado empezando en una posición arbitraria, se trata como una serie circular de números de dirección y se redefine el punto de partida de modo que la serie de números resultante forme un entero de módulo mínimo, esta normalización es exacta solamente si los propios contornos son invariables a la rotación y al cambio de escala, lo cual en la práctica ocurre raramente.

#### 4.1.2 Curvas $\phi$ -s

Este método es similar al de los códigos de cadena. En este caso se almacenan los vectores tangentes en cada punto del contorno. De hecho, sólo es importante la dirección del vector tangente y no su magnitud y por ello sólo se codifica el ángulo de la tangente  $\phi$  como función de la longitud de arco  $s$  del contorno. Por este motivo la representación se llama curva  $\phi$ -s.

Cuando se codifica el ángulo del vector tangente debe utilizarse algún marco de referencia (tangente a cero grados). Una suposición típica es considerar que la tangente en el punto origen del contorno es nula y codificar el resto respecto a esta. De esta forma, la curva  $\phi$ -s comienza y termina en cero.

La representación mediante curvas  $\phi$ -s tiene las siguientes ventajas:

- 1.- La representación es invariante a traslaciones.
- 2.- Las limitaciones en los ángulos posibles sólo están condicionadas por el método de segmentación empleado.
- 3.- Es invariante a rotaciones. Cuando se trata de comparar dos curvas con diferentes puntos de inicio, no hay más que desplazar el origen del plano  $\phi$ -s al nuevo punto de origen y aplicar un modelo toroidal en el eje  $s$ .

#### 4.1.3 Aproximaciones poligonales

Un contorno digital se puede tratar con una precisión arbitraria mediante un polígono. Para una curva cerrada, la aproximación es exacta cuando el número de los lados del polígono es igual al número de puntos del contorno, de modo que cada par de puntos adyacentes define un lado del polígono, para captar la esencia del contorno mediante un polígono. Existen varias técnicas para lograr esta aproximación [1].

La técnica de perímetro mínimo, se encierra el contorno en un conjunto de células concatenadas, si cada célula abarca solamente un punto del contorno, el error en cada célula entre el contorno original y la aproximación de perímetro mínimo sería, como máximo,  $\sqrt{2d}$ , siendo  $d$  la distancia entre píxeles. Este error puede reducirse a la mitad haciendo que cada célula esté centrada en su píxel correspondiente.

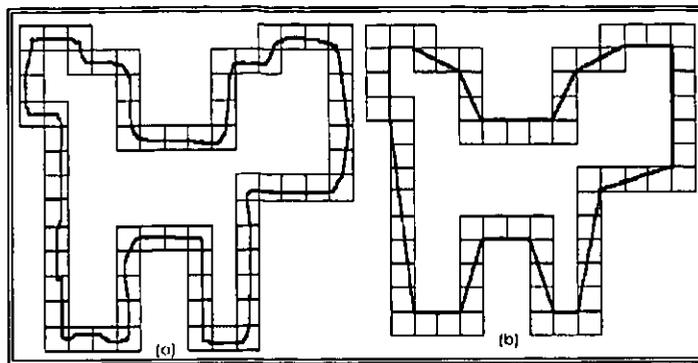


Figura 4. 3 (a) Contorno de un objeto encerrado por células; (b) polígono de perímetro mínimo. [B]

La técnica de fusión, consiste en fusionar puntos a lo largo del contorno hasta que el ajuste de la curva de error mínimo cuadrado de los puntos fusionados hasta el momento traspase un umbral preestablecido. Una de las principales dificultades de este método es que los vértices generalmente no corresponden a inflexiones (tales como esquinas) del contorno, porque no se empieza una línea hasta que el error no traspasa el umbral. Para dividir los lados del contorno consiste en subdividir sucesivamente el lado en dos partes hasta que satisfaga un criterio dado, esta aproximación tiene la ventaja de buscar puntos de inflexión destacados. Para un contorno cerrado, los mejores puntos para comenzar son normalmente los dos puntos más separados del contorno.

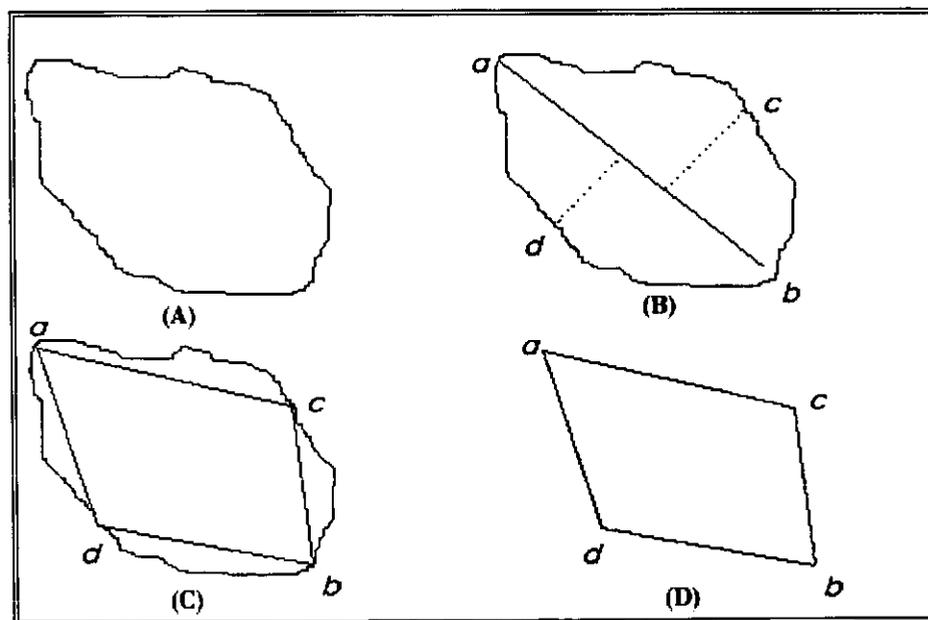


Figura 4. 4 (A) Contorno original; (B) contorno dividido en dos lados basándose en el cálculo de distancias; (C) unión de vértices; (D) polígono resultante. [B]

#### 4.1.4 Firma

Una firma es una representación funcional unidimensional de un contorno y se puede generar de varias formas, una de las más simples es representar la distancia desde el centro al contorno como una función del ángulo, la generación de firmas es un procedimiento que no varía con la traslación, pero depende de la rotación y la escala. Se puede conseguir la normalización con respecto a la rotación encontrando un modo de seleccionar el mismo punto de partida para generar la firma, independientemente de la orientación de la forma, utilizando como punto de partida el punto más alejado del centro.

La principal ventaja de este método es la simplicidad, pero presenta la desventaja potencial de que el escalado de la función completa depende sólo de dos valores: el mínimo y el máximo. Si las formas tienen ruido esta dependencia puede ser una fuente de error de objeto a objeto.

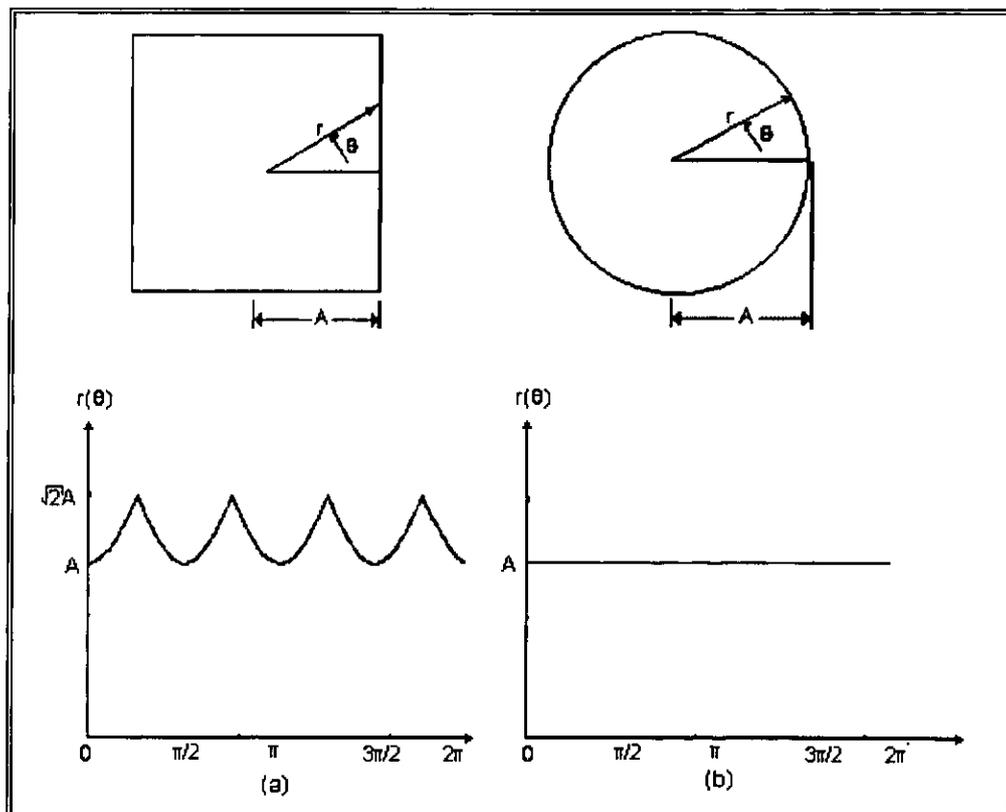


Figura 4. 5 Dos curvas de contornos sencillos y sus correspondientes firmas de distancia-ángulo. En (a)  $r(\theta)=A\sin\theta$ , mientras que en (b)  $r(\theta)$  es constante. [B]

#### 4.1.5 Lados del contorno

A veces es útil descomponer un contorno en lados, la descomposición reduce la complejidad del contorno y simplifica así el proceso de descripción. Esta solución es particularmente atractiva cuando el contorno presenta una o más concavidades significativas que contienen información sobre el tema. En este caso el empleo del cerco convexo de la región abarcada por el contorno es una poderosa herramienta para una descomposición robusta del contorno.

**ESTA TESIS NO SALE  
DE LA BIBLIOTECA**

### 4.1.6. El esqueleto de una región

Una aproximación importante consiste en reducir una región a un grafo, esta reducción se puede conseguir mediante el esqueleto. El esqueleto de una región se puede definir mediante la transformación del eje medio (MAT, de inglés Medial Axis Transformation). La MAT de una región  $R$  con borde  $B$  es la siguiente. Para cada punto  $p$  de  $R$ , se encuentra su vecino más próximo en  $B$ . Si  $p$  tiene más de un vecino de éstos, se dice que pertenece al eje medio (esqueleto) de  $R$  cabe mencionar que la implementación directa con base en estos cálculos es extremadamente laboriosa para una computadora

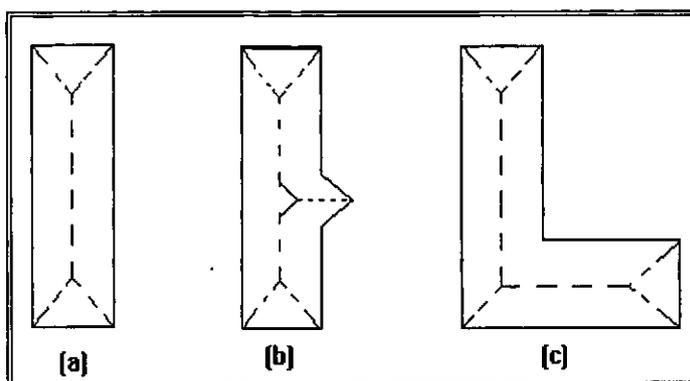
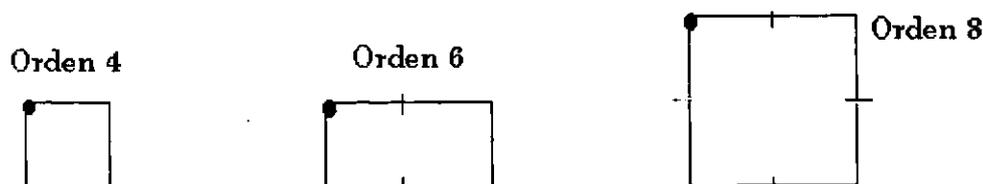


Figura 4. 6 Eje medio de tres regiones sencillas. [B]

## 4.2 DESCRIPTORES DE CONTORNO

### 4.2.1 Números de forma

El código de cadena de un contorno depende del punto de partida, pero dicho código se puede normalizar por un procedimiento directo: dado un código de cadena generado empezando en una posición arbitraria, se trata como una serie circular de números de dirección y se redefine el punto de partida de modo que la serie de números resultante forme un entero de módulo mínimo. También se puede normalizar por rotación utilizando la primera diferencia del código de cadena en vez del propio código. Esta diferencia se obtiene simplemente contando (en el sentido contrario a las agujas del reloj) el número de direcciones que separan dos elementos adyacentes del código. Por ejemplo, la primera diferencia del código de cadena de 4 direcciones 10103322 es 3133030. [1]



	Orden 4	Orden 6	Orden 8
Código de Cadena	0 3 2 1	0 0 3 2 2 1	0 0 3 3 2 2 1 1
Diferencia:	3 3 3 3	3 0 3 3 0 3	3 0 3 0 3 0 3 0
Número de Forma:	3 3 3 3	0 3 3 0 3 3	0 3 0 3 0 3 0 3

El *número de forma* de un contorno, basado en el código de 4-direcciones, se define como la primera diferencia del módulo menor. El *orden n* de un número de forma se define como el número de dígitos de su representación, si *n* es par es un contorno cerrado. Aunque la primera diferencia de un código de cadena es independiente de la rotación, en general el contorno codificado depende de la orientación del cuadrículado.

El *eje mayor* de un contorno es el segmento de recta que une los dos puntos más separados entre sí. El *eje menor* es perpendicular al eje mayor y de tal longitud que se podría formar un rectángulo que contenga exactamente al contorno. La relación entre el eje mayor y el menor se denomina *excentricidad* del contorno, y el rectángulo descrito anteriormente se denomina *rectángulo básico*. En la mayor parte de los casos se obtendrá un único número de forma alineando el cuadrículado del código de cadena con los lados del rectángulo básico.

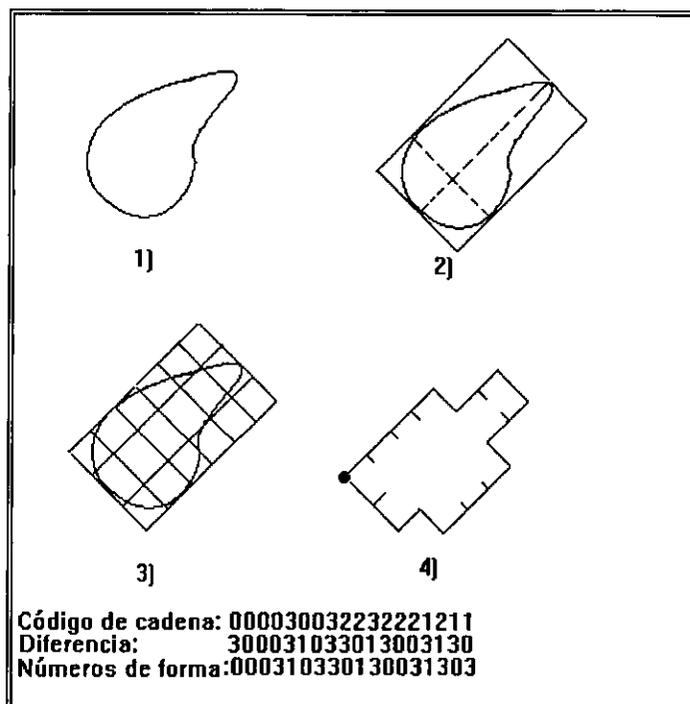


Figura 4. 7 Etapas de la generación de un número de forma. [B]

#### 4.2.2 Descriptores de Fourier

Un contorno es un objeto que tiene *N* píxeles numerados desde 0 hasta *N*-1. El píxel *k*-ésimo del contorno estará en la posición  $(x_k, y_k)$ . Se puede entonces describir el contorno mediante las ecuaciones paramétricas:

$$x(k) = x_k$$

$$y(k) = y_k$$

[Ecu. 4. 1]

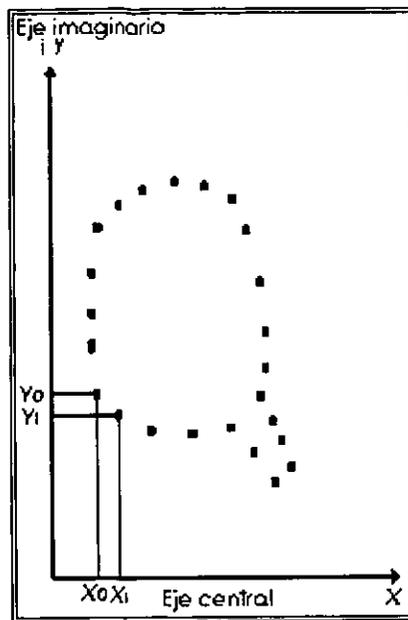


Figura 4. 8 Un contorno digital y su representación por medio de una serie compleja. Los puntos  $(x_0, x_0)$  y los  $(x_1, y_1)$  son (de forma arbitraria) los dos primeros puntos de la serie. [B]

Si se calcula la transformada de Fourier de cada función se obtiene dos espectros de la forma:

$$\begin{aligned} a_x(v) &= F(x(k)) \\ a_y(v) &= F(y(k)) \end{aligned}$$

[Ecu. 4. 2]

Dado que el número de puntos es finito se utiliza habitualmente la transformada discreta de Fourier (trata la señal como periódica pero eso no es problema pues el contorno es intrínsecamente periódico). Estos dos espectros se denominan descriptores de Fourier [1].

Considerando las coordenadas de los puntos del contorno como puntos del plano complejo

$$s(k) = x(k) + iy(k)$$

[Ecu. 4. 3]

Se puede obtener un único descriptor de Fourier si se transforma esta función compleja. Como se puede observar a continuación existe una relación entre este descriptor y los dos planteados anteriormente,

$$\begin{aligned} a(v) &= F(s(k)) = F(x(k) + iy(k)) = F(x(k)) + iF(y(k)) = a_x(v) + ia_y(v) = \\ &= \left[ R(a_x(v)) - I(a_y(v)) \right] + i \left[ I(a_x(v)) + R(a_y(v)) \right] \end{aligned}$$

[Ecu. 4. 4]

en forma discreta:

$$a(v) = \sum_{k=0}^{N-1} s(k) e^{\left[-\frac{j2\pi vk}{N}\right]}$$

[Ecu. 4. 5]

Para  $v = 0, 1, 2, \dots, N-1$ . Los coeficientes complejos  $a(v)$  se denominan *descriptores de Fourier* del contorno. La transformada inversa de Fourier de los  $a(v)$  restaura  $s(k)$ . Esto es:

$$s(k) = \frac{1}{N} \sum_{v=0}^{N-1} a(v) e^{[j2\pi vk/N]}$$

[Ecu. 4. 6]

para  $k = 0, 1, 2, \dots, N-1$ . Supongamos, sin embargo, que en vez de todos los valores de  $a(v)$  solamente se utilizan los primeros  $M$  coeficientes. Esto es equivalente a hacer  $a(n) = 0$  para  $n > M-1$ . El resultado es la siguiente aproximación a  $s(k)$ :

$$s(k) = \frac{1}{N} \sum_{n=0}^{M-1} a(n) e^{[j2\pi nk/N]}$$

[Ecu. 4. 7]

para  $k = 0, 1, 2, \dots, N-1$ . Aunque sólo se utilizan  $M$  términos para obtener cada componente de  $s(k)$ ,  $k$  pertenece al intervalo de 0 a  $N-1$ .

Considerando el significado de los espectros. En primer lugar, si hay contenido de alta frecuencia significa que hay cambios rápidos en la coordenada correspondiente. Y en segundo lugar, si hay contenido de baja frecuencia significa que el contorno varía de forma suave. Luego, las componentes de baja frecuencia capturan la forma general del contorno y las de alta frecuencia capturan los detalles.

De esta forma, por ejemplo, realizar un filtro paso bajo de un descriptor de Fourier equivale a suavizar el contorno del objeto.

Por tanto, no es necesario usar todas las componentes para caracterizar el contorno, si bien es cierto que cuantas más se utilicen más se refinará la aproximación.

Se pueden utilizar las propiedades de la transformada de Fourier para describir las propiedades de los descriptores de Fourier.

**Traslación:** Si se traslada el objeto se suma un valor constante a todos los puntos del contorno. En los descriptores de Fourier cambiará la componente de frecuencia cero. Pero esta componente no tiene información sobre la forma luego los descriptores de Fourier son invariantes a traslaciones excepto para la componente de frecuencia cero.

**Rotación:** La rotación en el plano complejo de un ángulo  $\theta$ , respecto al origen de coordenadas, es equivalente a multiplicar el espectro por  $e^{j\theta}$ .

$$a_r(v) = a(v)e^{-rv}$$

[Ecu. 4. 8]

Escalado: Escalar un objeto es multiplicar las posiciones del contorno por una constante, con lo cual sólo se ve afectada la componente de frecuencia nula de los descriptores de Fourier.

Punto de origen: Cambiar el punto de origen es trasladar la señal unidimensional  $s(k)$  en la dimensión  $k$  y esta translación en el dominio espacial desplaza la fase pero no la magnitud.

### 4.2.3 Momentos

La forma de los lados del contorno ( y de las firmas) se pueden describir cuantitativamente utilizando momentos. Transformando un contorno, como una función unidimensional en función de una variable aleatoria. Los momentos son propiedades estadísticas.

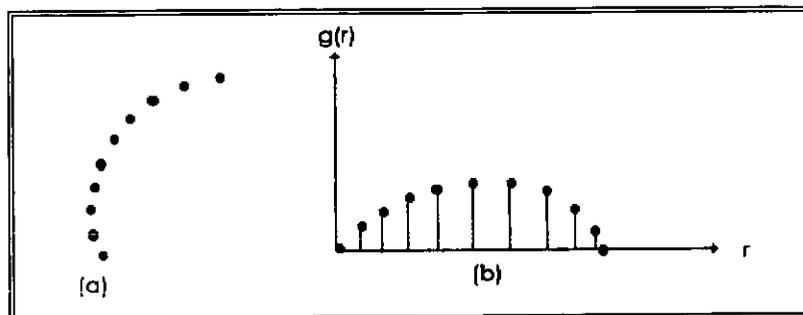


Figura 4. 9 (a) lados de un contorno; (b) representación como función unidimensional. [B]

Para una función  $f(x)$ , que representa un contorno, se puede calcular el valor medio de la función mediante:

$$\mu = \frac{\sum_{x=1}^N xf(x)}{\sum_{x=1}^N f(x)} \quad (1^{\text{er}} \text{ Momento})$$

[Ecu. 4. 9]

La varianza se puede calcular mediante

$$\sigma^2 = \frac{\sum_{x=1}^N (x - \mu)^2 f(x)}{\sum_{x=1}^N f(x)} \quad (2^{\text{do}} \text{ Momento})$$

[Ecu. 4. 10]

Una tercera propiedad estadística, llamada sesgo, describe la simetría de una función mediante:

$$skew = \frac{\sum_{x=1}^N (x - \mu)^3 f(x)}{\sum_{x=1}^N f(x)} \quad (3^{\text{er}} \text{ Momento})$$

[Ecu. 4. 11]

Todos estos ejemplos son momentos de una función  $f(x)$ .

En general se pueden definir momentos sobre cualquier punto arbitrario, aunque usualmente se definen sobre el cero o sobre la media. El  $n$ -ésimo momento sobre el cero,  $m_n$ , se define mediante,

$$m_n = \sum_{x=1}^N f(x)x^n$$

[Ecu. 4. 12]

$m_0$  es el valor total de la función. La media es el primer momento cero.

$$\mu = \frac{m_1}{m_0}$$

[Ecu. 4. 13]

El  $n$ -ésimo momento sobre la media, llamado  $n$ -ésimo momento central, es

$$\mu_n = \sum_{x=1}^N f(x)(x - \mu)^n$$

[Ecu. 4. 14]

El momento cero central es, de nuevo, el valor total de la función. El primer momento central es, obviamente, cero. El segundo momento central, normalizado por el valor total de la función, es la varianza,

$$\sigma^2 = \frac{\mu_2}{\mu_0}$$

[Ecu. 4. 15]

El tercer momento central, normalizado por el valor total de la función, es el sesgo de la función,

$$skew = \frac{\mu_3}{\mu_0}$$

[Ecu. 4. 16]

El cuarto momento central, normalizado por el valor total de la función, es el "kirtosis" de la función

$$kirtosis = \frac{\mu_4}{\mu_0}$$

[Ecu. 4. 17]

Existe un número infinito de momentos centrales con los cuales se puede llegar a describir completamente la forma de la función.

Hay muchas firmas que pueden ser descritas mediante funciones unidimensionales:

- 1.- Segmentos de contornos que pueden describirse como alguna función de distancia.
- 2.- Firmas unidimensionales como códigos de cadena, diferencias de códigos de cadena, etc.

Se pueden describir cualquiera de estas firmas de N elementos mediante los N-ésimos momentos centrales. En general no es necesario tomar el conjunto completo de momentos centrales si bien la interpretación será más inexacta cuantos menos momentos centrales incluya.

Se puede extender el concepto de momento definiendo el momento ij-ésimo sobre el centro mediante:

$$m_{ij} = \sum_{x=1}^N \sum_{y=1}^N f(x, y) x^i y^j$$

[Ecu. 4. 18]

$m_{00}$  es el valor total de la función. El cociente  $m_{10}/m_{00}$  es la componente x de la media y es la componente y de la media.

El momento central ij-ésimo vendrá dado por

$$\mu_{ij} = \sum_{x=1}^N \sum_{y=1}^N f(x, y) (x - \mu_x)^i (y - \mu_y)^j$$

[Ecu. 4. 19]

Como en el caso unidimensional  $\mu_{10} = \mu_{01} = 0$ .

Si no se utilizan suficientes momentos se corre el riesgo de que dos contornos diferentes tengan los mismos descriptores. En cierto sentido, la secuencia de momentos es análoga a las componentes de los descriptores

de Fourier: los primeros momentos dan la forma general y conforme se añaden momentos se añaden detalles finos de la forma.

Los momentos se han mostrado muy útiles como conjunto de descriptores para hacer una sobreposición. Su respuesta a transformaciones es la siguiente:

- 1.-Traslación: Sólo cambia la media, pero no los momentos de orden superior.
- 2.-Rotación: Cambian las varianzas relativas y los momentos de orden mayor. Los valores característicos de la matriz de covarianza son invariantes a rotación.
- 3.-Escala: Escalar el objeto por un factor S es lo mismo que escalar las coordenadas x e y por un factor 1/S. En general, el n-ésimo momento se escala por  $(1/S)^n$ .

La ventaja de los momentos sobre otras técnicas es que la implementación de los descriptores anteriores es directa y que, además, contienen también una interpretación "física" de la forma del contorno.

#### 4.2.4 El Histograma del código de cadena

El histograma del código de cadena (HCC) se utiliza para representar objetos que parecen similares a un observador humano. No se utiliza para tareas exactas de la detección y de la clasificación. El HCC se calcula de acuerdo con la representación del código de cadena de un contorno. [8]

El código de cadena de Freeman es una manera compacta de representar un contorno de un objeto. El código de cadena es una secuencia de n conexiones,  $\{c_i, i=1,2,3,\dots, n\}$ , donde  $c_i$  es un vector que conecta los pixeles vecinos del contorno. Las direcciones de  $c_i$  se cifran con valores de números enteros  $k=0,1,\dots,K-1$  en sentido opuesto a las manecillas del reloj teniendo como referencia el eje x positivo. El número de direcciones de K toma valores enteros de  $2^{(M+1)}$  donde M es el número de columnas del histograma. Los códigos de cadena donde  $K > 8$  se les llaman códigos de cadena generalizados.

El cálculo del histograma del código de cadena es rápido y simple, el HCC es una función discreta,

$$p(k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, K-1,$$

[Ecu. 4. 20]

Donde  $n_k$  es el número de los valores del código de cadena  $k$ , y  $n$  es el número de conexiones del código de cadena.

Un ejemplo simple se representa en la figura 4.10. En la figura 4.10(a) están las ocho direcciones del código de cadena. En la figura 4.10(b) es un objeto a representar. El punto de partida para la codificación de cadena está marcado con un punto negro, y la dirección de cadena de la codificación es a la derecha. En la figura 4.10(c)-(d) es el código de cadena y los HCC del contorno de la figura.

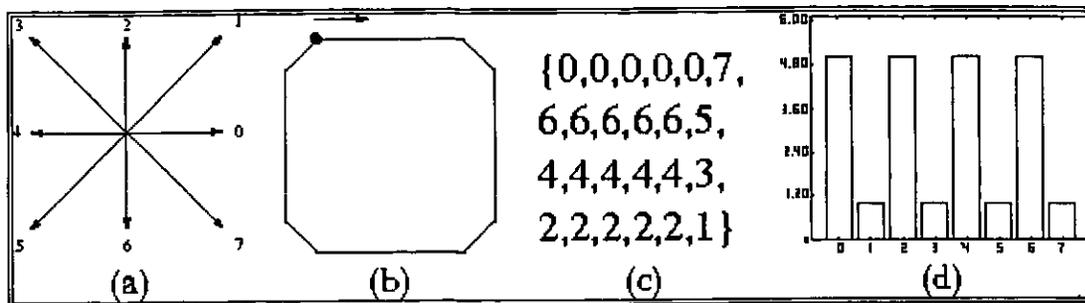


Figura 4. 10 (a) Las direcciones de los ocho conectaron el código de cadena ( $K= 8$ ). (b) Un objeto-muestra, (c) presentación del código de cadena de la figura, y (d) el histograma del código de cadena del objeto. [F]

El HCC es una traducción y un descriptor invariante de la dimensión de una variable de escala. Es invariante a las rotaciones de  $90^0$ , porque la rotación de  $90^0$  causa solamente una rotación circular en el HCC, alcanza una mejor invariación a la rotación que el histograma normalizado del código de cadena (NHCC). Considera las longitudes de diversas direcciones. [8]

#### 4.2.5 El Histograma en parejas geométricas

El histograma en parejas geométricas (HPG) es un descriptor de gran alcance para dimensiones de una variable aplicable a dimensiones poligonales. Puede ser aplicado también a una dimensión de una variable irregular, si la dimensión se aproxima primero a un polígono. [8]

Considere un polígono definido por sus puntos del borde de la siguiente manera

$$(\bar{x}(t), \bar{y}(t)) \in R^2$$

[Ecu. 4. 21]

Ahora los puntos sucesivos del borde definen los segmentos de línea que forman el polígono. Se calcula el HPG usando la siguiente estrategia: Considere cada segmento de línea, una línea de referencia girada. Entonces existe un ángulo relativo  $\theta \in [0, \pi]$  y dos distancias perpendiculares, mínima y máxima ( $d_{min}$  y  $d_{max}$ ), estos valores pueden ser calculados para todas las otras líneas referidas a la línea de referencia. El valor del histograma se genera en función de un ángulo  $\theta$  y el valor del segmento obtenido de la diferencia entre  $d_{min}$  y  $d_{max}$ , ver figura 4.11.

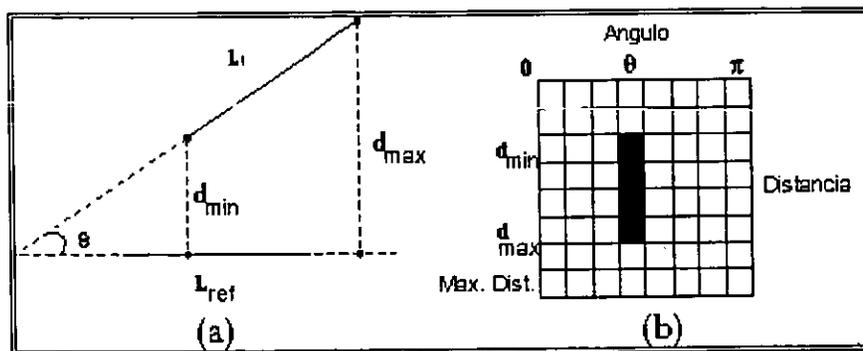


Figura 4. 11 (a) Angulo relativo y distancias perpendiculares entre dos líneas, y (b) el histograma en parejas geométrico. [F]

Un nuevo esquema para reducir el tamaño del HPG es el siguiente, las expectativas condicionales de cada fila y columna se calculan y se recogen en un vector característico. Donde  $p(i,j)$  es el valor de HPG en la posición  $(i,j)$ . Entonces el vector característico  $f_{HPG}$  se define como:

$$f_{HPG} = [E_r(1)E_r(2)\dots E_r(N)E_c(1)E_c(2)\dots E_c(M)]^T$$

[Ecu. 4. 22]

donde N es el número de las filas del HPG, M es el número de las columnas del HPG,  $E_r(i)$  es la expectativa condicional de la fila del histograma en i,

$$E_r(i) = \frac{\sum_j j p(i, j)}{\sum_j p(i, j)}$$

[Ecu. 4. 23]

y  $E_c(j)$  es la expectativa condicional de la columna del histograma en j,

$$E_c(j) = \frac{\sum_i i p(i, j)}{\sum_i p(i, j)}$$

[Ecu. 4. 24]

Así el número de características se reduce del NM a N+M. Esto presenta ahorros significativos en requisitos del tiempo y de memoria de cálculo, especialmente si N y M son grandes.

#### 4.2.6 Combinación de los descriptores simples.

Existen descriptores simples, donde se utilizan un objeto para su reconocimiento, estos descriptores son limitados para tareas complejas de reconocimiento, pero la combinación de estos puede dar buenos resultados, por lo general se requiere solamente un paso progresivo de cálculo  $O(N)$ , excepto en la convexidad donde se requieren  $O(N^2)$  pasos progresivos de cálculo. [8]

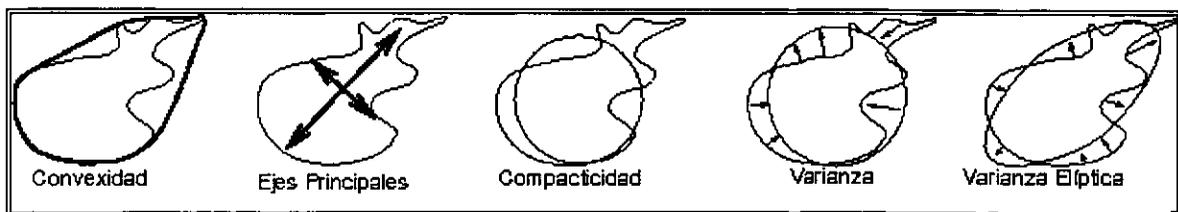


Figura 4. 12 Cinco descriptores simples. [F]

La convexidad se puede definir como la relación de transformación de perímetros del casco convexo del contorno original. El casco convexo es la cubierta convexa mínima de un objeto. El algoritmo para construir

un casco convexo implica trazar un contorno reduciendo al mínimo los ángulos en cada paso de progresión. En la práctica, el contorno resultante puede ser mayor que el original

Las dimensiones principales de un objeto se pueden definir como segmentos de las líneas que se cruzan ortogonalmente y en el centro de figura, este objeto representa un marco de referencia con cruce por cero. De esta manera, un contorno se considera como una distribución estadística. La relación de transformación del eje principal se puede calcular de la matriz de covariación de un contorno. No es necesario calcular vectores propios reales.

La compacticidad se define a menudo como la relación de transformación del perímetro ajustado y de su área. Alcanzando el mínimo en un objeto circular y tendiendo a hacerse muy grande en objetos finos y complejos.

Algunas veces la forma puede ser comparada nuevamente con una figura. Un círculo es una figura general y obvia para elegir. La varianza es el error cuadrático medio proporcional con respecto al círculo sólido. Sería cero en el caso de un círculo perfecto y se incrementa análogamente con la complejidad de la figura.

La varianza elíptica se define similarmente a la varianza circular. Utilizando una elipse para medir el error medio cuadrado.

### 4.3 DESCRIPTORES RELACIONALES

Los métodos expuestos hasta ahora se aplican generalmente a contornos individuales y a regiones de interés de una imagen. El siguiente nivel de complejidad en el proceso de descripción es organizar estos componentes para explorar cualquier relación estructural que puede existir entre ellos. [1]

Introduciremos este concepto con la sencilla estructura de una escalera, suponiendo que esta estructura ha sido segmentada de una imagen y que queremos describirla. Definiendo los elementos primitivos **a** y **b** podemos codificarla, las propiedades más evidentes de en la estructura codificada es la respectividad de los elementos **a** y **b**.

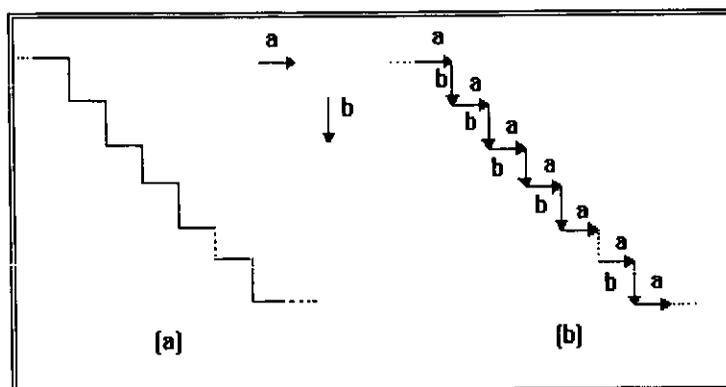


Figura 4. 13 (a) una estructura simple de escalera; (b) estructura codificada. [B]

Por lo tanto, un método de descripción simple consiste en formular una relación recursiva que implique a estos elementos primitivos. Una posibilidad es utilizar reglas de reescritura:

- 1)  $S \rightarrow aA$
- 2)  $A \rightarrow bS$ , y
- 3)  $A \rightarrow b$

Donde  $S$  y  $A$  son variables y los elementos  $a$  y  $b$  son constantes que corresponden a los primitivos definidos anteriormente. La regla 1 indica que  $S$ , denominado símbolo de partida, se puede remplazar por el primitivo  $a$  y la variable  $A$ . Esta variable a su vez, se puede reemplazar por  $b$  y  $S$  o por  $b$  solamente. Remplazar  $A$  por  $bS$  (segunda regla) conduce a la primera regla y se puede repetir el procedimiento. Remplazar  $A$  por  $b$  (tercera regla) termina el procedimiento, ya que no hay variables en la expresión. La figura 4.14, ilustra algunos ejemplos de derivaciones de estas reglas, con los números de abajo de la estructura representa el orden en que se aplicaron las reglas 1,2 y 3. Se mantiene la relación entre  $a$  y  $b$ , porque estas reglas obligan a que una  $a$  vaya seguida por una  $b$ . Notablemente estas tres simples reglas de reescritura se pueden utilizar para generar (o describir) muchas estructuras similares.

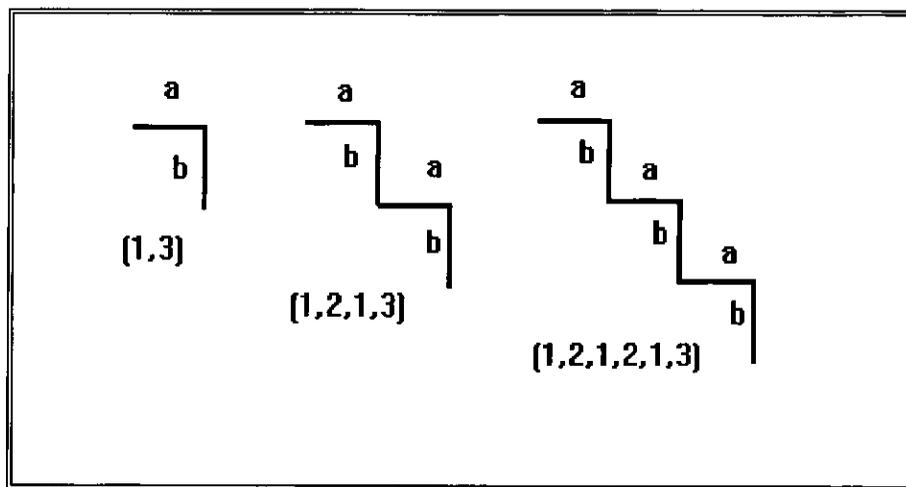


Figura 4. 14 Ejemplos de derivaciones de las reglas  $S \rightarrow aA$ ,  $A \rightarrow bS$ , y  $A \rightarrow b$ . [B]

Como las cadenas son estructuras unidimensionales, su aplicación a la descripción de imágenes requiere establecer un método adecuado para reducir las relaciones posicionales bidimensionales a la forma unidimensional. La mayoría de las aplicaciones de las cadenas a la descripción de imágenes se basan en la idea de extraer segmentos conexos de los objetos de interés, una aproximación consiste en seguir el contorno de objeto y codificar el resultado con segmentos de dirección y/o longitud específicas.

Otra aproximación más general, es describir secciones de una imagen (tales como pequeñas regiones homogéneas) por medio de segmentos dirigidos, que se pueden unir de otra forma además de las conexiones cadeza-a-cola.

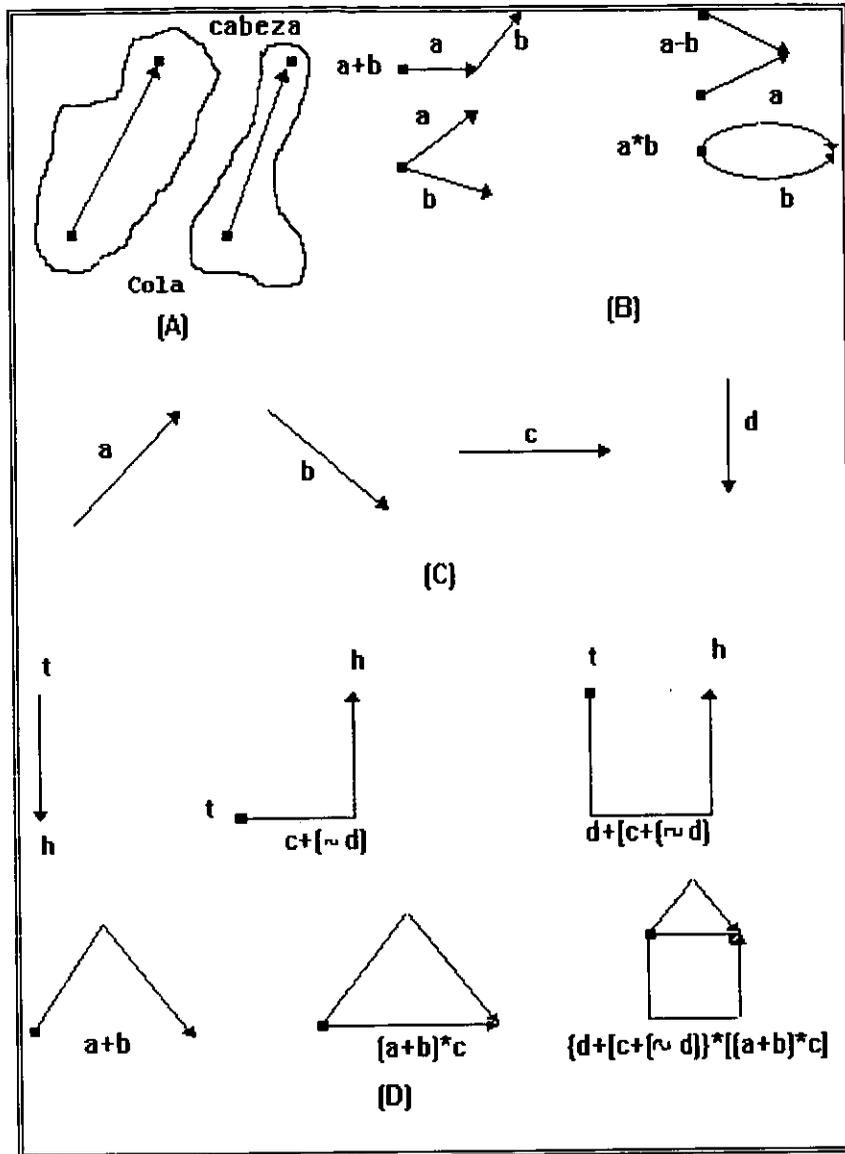
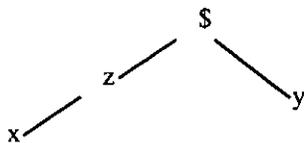


Figura 4. 15 (A) Primitivos abstractos; (B) operaciones entre primitivos; (C) conjunto de primitivos específicos, donde  $(\sim d)$  indica el primitivo "d" con su dirección invertida; (D) etapas de construcción de una estructura. [B]

Si se trata de estructuras disjuntas, una de las aproximaciones más útiles para hacerlo es utilizar descriptores tipo árbol. Un árbol  $T$  es un conjunto finito de uno o más nodos para los que:

- 1) Hay un único nodo  $\$$  designado como la raíz, y
- 2) Los nodos restantes están divididos en "m" subconjuntos disjuntos  $T_1, \dots, T_m$ , cada uno de los cuales es a su vez un subárbol de  $T$ .

El árbol de frontera es el conjunto de nodos de la parte inferior del árbol (las hojas) tomando estos valores de izquierda a derecha, por ejemplo, el siguiente árbol tiene raíz  $\$$  y frontera  $xy$ .



Generalmente, dos tipos de información son importantes en un árbol:

- 1) Información sobre el nodo almacenado como un conjunto de palabras que describen el nodo, y
- 2) Información que relaciona un nodo con sus vecinos almacenada con un conjunto de punteros a esos vecinos. [12]

Cuando se utilizan en la descripción de imágenes, el primer tipo de información identifica una forma primitiva, mientras que el segundo tipo define la relación física del primitivo con otras subestructuras.

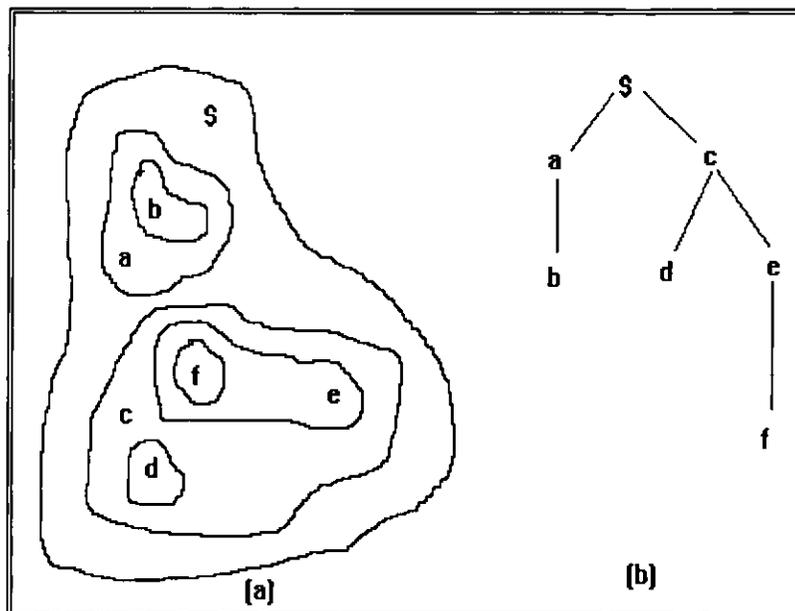


Figura 4. 16(a) Una simple región compuesta; (b) representación en un árbol obtenida al utilizar la relación "dentro de" [B]

## Π COMENTARIO AL CAPITULO Π

Los procesos de codificación a través de descriptores formulan el primer paso para ejecutar un proceso de comparación, para conseguir una adecuada descripción es establecer la complejidad de la figura y sus características que la conforman, una imagen que presenta textura siempre implicará un aumento considerable de tiempo de procesamiento. Cuando la imagen ha sido conformada por contornos o líneas rectas, esto nos facilitara su caracterización al bajar el numero de puntos para procesar. Aún en este caso los puntos que conforman un contorno en muchos casos puede considerarse muy grande, los procesos de representación y descripción tienen el objetivo de reducir la información que representa al objeto. Esta información difícilmente tendrá una representación gráfica en muchas ocasiones, pero facilitara los procesos de manejo de información en el reconocimiento, dichos procesos serán estudiados en el siguiente capítulo.

Existe una gran variedad de descriptores y formas de codificarlos, esto es consecuencia de la diversidad de objetos que pueden ser representados como un código, los descriptores no son en general rígidos para su programación, ya que ofrecen una gran libertad para que puedan adecuarse a los objetos con los que se este trabajando, lo importante es establecer una estrategia basadas en la configuración y características del objeto a representar. Si se quiere crear un método que automatice este proceso, debe generarse un proceso que prevean inconvenientes en el sistema integral de procesamiento digital de imágenes, como puede ser la introducción de ruido en la imagen en los procesos primarios de extracción de información.

Los principales problemas que se presentan cuando se generar un código de representación es que este código representará el objeto en ese momento y con las características de tamaño posición y orientación, cualquier variación en estos aspectos generará un código totalmente diferente, muchos descriptores son susceptibles de cometer errores por estas implicaciones, por ello es importante establecer un sistema integral con reglas bien establecidas de representación para evitar la acumulación de errores. Una alternativa para tener más confianza en los métodos de descripción es generar esquemas que codifiquen globalmente una serie de descriptores de varios objetos que conforman una escena, como lo son los árboles, que facilitaran posteriormente el proceso de reconocimiento de patrones.

· C · A · P · I · T · U · L · O ·

V

RECONOCIMIENTO DE  
PATRONES

En la actualidad la globalización de la información ha creado seres impersonales, los individuos son datos procesados en una computadora, por ello existe la necesidad de que cada individuo tenga un rasgo inimitable que le devuelva su individualidad, rasgos particulares y muy simples pueden ser modelados en imágenes digitales. El procesamiento digital de imágenes ofrece una gran cantidad de técnicas enfocadas al manejo de este tipo de información, el reconocimiento de patrones en imágenes es un área medular si se intenta crear un proceso de identificación personalizada.

En la medicina existen muchos casos en los que es importante reconocer patrones, un ejemplo es el análisis de radiografías y tomografías; otro es el conteo celular. En tareas como estas, confiamos en la capacidad de los médicos, pero esta se ve limitada por las aptitudes de observación y las herramientas de análisis empleadas. Los conceptos de aprendizaje y reconocimiento de patrones han encontrado diversos usos en la medicina. El reconocimiento de la voz y sonidos se ha empleado en la terapia del lenguaje. El reconocimiento de patrones tiene amplias aplicaciones en las imágenes de rayos X, de positrones y de resonancia magnética nuclear (RMN), así como en las de ultrasonido (donde se desarrollan técnicas para mejorar la identificación y seguimiento de formas, como la actividad del feto en el útero).

Desde la década de los setenta, se ha considerado que los problemas de Reconocimiento de Patrones son de carácter extraordinariamente complejos, y que en la mayoría de los casos su solución está en la formación de algún sistema híbrido, que combina las diferentes técnicas de Inteligencia Artificial, como son, Lógica Difusa, Redes Neuronales, Sistemas Expertos, bajo este prisma y empujado por satisfacer las necesidades de desarrollo industrial, aparece la necesidad de clasificación de los eventos de forma automática. La línea de estudio llevada a cabo durante los últimos cuatro años, consiste en buscar metodologías de clasificación mediante la combinación de técnicas tradicionales. Una de las líneas más fructíferas y de actual investigación, consiste en la combinación de mapas auto organizados (SOM) con técnicas estadísticas. El uso de redes no supervisada, como los mapas de Kohonen, sirven para el conocimiento de la función de densidad del espacio de entrada, elemento importante para el análisis de los datos y la parametrización del espacio en grupo y en sus correspondientes probabilidades condicionadas; Utilizándose técnicas de prueba de seguimiento de distribuciones, transformaciones Box-Cox, selección de características con técnicas MANOVA transformaciones de separación de grupos, y elección del espacio como minimización de la probabilidad de error, pueden obtenerse los clasificadores óptimos al permitir introducir el conocimiento *a priori* en las diversas regiones del espacio de medidas.

## 5.1 DESCRIPCION

Desde el inicio de la informática moderna ha habido un interés significativo, en la emulación del comportamiento inteligente de los seres humanos, incluyendo la capacidad de reconocer y de percibir varios modelos dimensionales, especialmente existe un interés significativo en modelos acústicos y visuales en las últimas dos décadas [14].

Cualquier sistema de reconocimiento puede estar comprendido en alguno de estos dos modos de operación:

- i) Modo del análisis (el aprender)
- ii) Modo del reconocimiento (aplicación).

En el modo de análisis, los modelos de prueba se analizan para aprender qué información es relevante para una clasificación. En el modo de reconocimiento, los modelos nuevos se clasifican según el conocimiento obtenido en el modo del análisis.

Las técnicas para el reconocimiento basado en un proceso computacional se pueden agrupar en cuatro grupos:

- i) Estadísticos
- ii) Estructural
- iii) Basado en inteligencia artificial
- iv) Redes Neuronales.

◊ *Los métodos estadísticos* se basan en una descripción de un modelo, donde cada muestra es representada por un vector dimensional de  $n$  características. Cada característica es una medida numérica de las muestras, por ejemplo, su longitud o color. Cuando se elige alguna característica para su análisis, las variaciones estadísticas de ésta, pueden ser descritas y evaluadas en cualquier región, cada región es clasificada como un modelo específico. En el proceso de reconocimiento de una región desconocida es clasificarla e identificada dentro de los modelos establecidos.

◊ *Los métodos estructurales* hacen uso explícito del conocimiento sobre la estructura del objeto, utilizan una descripción del modelo donde cada modelo se divide en modelos secundarios llamados primitivos, los primitivos no tienen ninguna relación directa a la estructura modelo. Las técnicas estructurales se subdividen en prototipos estructurales y métodos sintácticos. Los métodos sintácticos, en su modo de análisis, consisten en construir las reglas para combinar primitivos y obtener la estructura de un objeto dado. Los métodos se formulan alrededor del concepto de lenguajes formales, cada primitiva se representa como un símbolo terminal y su gramática, deducidos para cada modelo. En el modo de reconocimiento, tratan de construir el objeto usando las reglas asociadas a una clase específica, este proceso se llama el análisis de sintaxis o el análisis de una cadena.

◊ En los métodos basados en *inteligencia artificial*, numérico, estructural y de otros tipos, tienen la finalidad de describir modelos. El modo de análisis, consiste en determinar apropiadamente la discriminación de la información, especificar cada modelo como un concepto abstracto, y especificar la relación conceptual entre la entidad implicada y el sistema. La fase del reconocimiento implica tomar observaciones de un objeto y usar un modo de inferencia para determinarse si el modelo puede ser un caso específico de un concepto abstracto.

◊ El acercamiento a las *redes neuronales* tiene muchas semejanzas con el reconocimiento de modelo estadístico, referente a la representación de datos y a los principios de la clasificación. La puesta en práctica es muy diferente. El modo del análisis implica la configuración de una red de neuronas artificiales y del entrenamiento de la red para determinarse cómo las neuronas individuales pueden relacionarse. El modo del reconocimiento implica enviar datos a través de la red y evaluar que patrón contó con el mayor número de preferencias.

Actualmente existen nuevos enfoque para tratar de lograr una mayor efectividad en el reconocimiento de patrones. Los métodos híbridos se han diseñado para reunir la potencialidad de dos o más de los métodos puros, combinando la mayor potencialidad de cada método, dependiendo del proceso a realizar y de la naturaleza de la aplicación. [14]

### 5.1.1 Elementos del análisis de imágenes

El análisis de imágenes consiste en descubrir, identificar y comprender los patrones que son relevantes para poder dotar a una máquina inteligente, en algún sentido, de la capacidad de aproximación, similar a la de los seres humanos. El concepto de inteligencia es algo vago, particularmente con referencia a una máquina, existen diversas características asociadas con la inteligencia:

- 1) La capacidad de extraer información de interés, separando lo importante de un conjunto de detalles irrelevantes.
- 2) La capacidad de aprender a partir de ejemplos y de generalizar este conocimiento en diversas circunstancias.
- 3) La capacidad de realizar inferencias a partir de una información incompleta.

Gracias a las investigaciones en sistemas informáticos y biológicos, continuamente se descubren nuevas y prometedoras teorías para explicar el conocimiento visual humano. Sin embargo, las técnicas más avanzadas de análisis de imágenes por computadora se basan en su mayor parte en fórmulas heurísticas, adaptadas para la resolución de problemas específicos. [1]

En el análisis de imágenes existen tres áreas básicas:

- 1) Procesado de bajo nivel.  
El procesado de bajo nivel trata de un tipo de funciones a las que se puede considerar como reacciones automáticas, y que no requieren inteligencia por parte del sistema de análisis de imágenes, como los son la adquisición y el preprocesado de la imagen
- 2) Procesado de nivel intermedio.  
El procesado de nivel intermedio se refiere a la extracción y caracterización de los componentes de la imagen que se obtienen de un proceso de bajo nivel. El procesado de nivel intermedio abarca la segmentación y la descripción
- 3) Procesado de alto nivel  
El procesado de alto nivel comprende el reconocimiento y la interpretación. Estos dos procesos tienen un gran parecido con lo que normalmente se conoce como conocimiento inteligente. En el reconocimiento y especialmente en la interpretación, nuestro conocimiento y comprensión de los principios fundamentales llega a ser menos preciso y mucho más especulativo que muchos procesos de nivel inferior. Como lo son: los métodos de decisión teórica para el reconocimiento; los métodos estructurales y los métodos para la interpretación

### 5.1.2 Patrones y clases de patrones

Un patrón es una descripción estructural o cuantitativa de un objeto, en general, un patrón está formado por uno o más descriptores, en otras palabras es una disposición de descriptores. Una clase de patrones es una familia de patrones que comparten algunas propiedades comunes, representadas por  $w_1, w_2, w_3$ , donde  $M$  es el número de clases. Existen tres principales representaciones de patrones, los vectores (para descripciones cuantitativas); las cadenas y los árboles (para descripciones estructurales).

Un patrón vectorial  $\mathbf{x}$  depende de la técnica de medida utilizada para describir el patrón físico, cada componente,  $x_i$ , representa el  $i$ -ésimo descriptor y  $n$  es el número de descriptores.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

[Ecu. 5.1]

Por ejemplo, supongamos que queremos describir tres tipos de lirios (*Iris setosa*, *Iris virginica* e *Iris versicolor*) por la medida de longitud y la anchura de sus pétalos, en este caso se utilizarán un vector bidimensional, donde  $x_1$  y  $x_2$  corresponden a la anchura y a la longitud del pétalo respectivamente, y existen tres tipos de patrones, representados por  $w_1$ ,  $w_2$  y  $w_3$ , corresponden a las variedades setosa, virginica y versicolor, respectivamente.

Las técnicas de generación de patrones vectoriales producen clases de patrones caracterizadas por información cuantitativa. En algunas aplicaciones, las características de los patrones se describen mejor utilizando relaciones estructurales. Por ejemplo en la figura 5.1, se tiene una estructura de escalera, con este método de descripción se perdería la estructura básica, que consiste en la repetición de dos sencillos elementos primitivos. Tendría más sentido definir las primitivas  $a$  y  $b$  y hacer que el patrón sea la cadena de símbolos  $w = \dots abababa \dots$ .

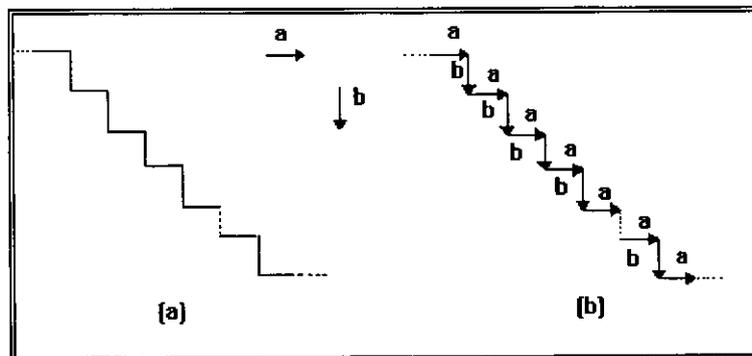


Figura 5.1 (a) una estructura simple de escalera; (b) estructura codificada en términos de los primitivos  $a$  y  $b$  para producir la representación en forma de cadena  $\dots abababab\dots$  [B]

Las representaciones en forma de cadena generan adecuadamente patrones de objetos y otras entidades cuya estructura se basa en conexiones relativamente sencillas de primitivos, normalmente asociadas a formas de bordes o contornos. Una técnica potente para muchas aplicaciones consiste en la utilización de descriptores en forma de árbol, esquemas basados en jerarquías que formaran un árbol. [1]

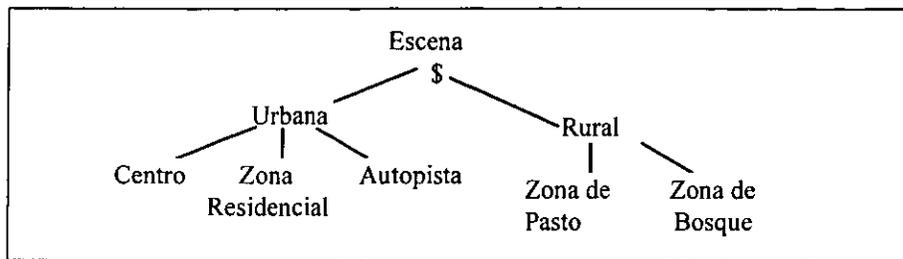


Figura 5. 2 Estructura de árbol generada de una escena que involucra varias características. [B]

Podemos seguir con estas subdivisiones hasta que alcancemos el límite de nuestra capacidad para discriminar diferentes regiones de la imagen.

### 5.1.3 Métodos de decisión teórica

Los métodos de decisión teórica para el reconocimiento se basa en funciones de decisión ( o discriminantes) Supongamos que  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  representa un patrón de dimensiones  $n$ , el problema básico en el reconocimiento de patrones por decisión teórica consiste en encontrar  $M$  funciones de decisión  $d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_M(\mathbf{x})$  que tengan la propiedad de que si un patrón  $\mathbf{x}$  pertenece a la clase  $w_i$ , entonces

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, 3, \dots, M; \quad j \neq i$$

[Ecu. 5. 2]

En otras palabras, un patrón desconocido  $\mathbf{x}$  pertenece a la  $i$ ésima clase de patrones si al sustituir la  $\mathbf{x}$  en todas las funciones de decisión  $d_i(\mathbf{x})$  toma el mayor valor numérico. La frontera de decisión que separa la clase  $w_i$  de  $w_j$  viene dada por valores de  $\mathbf{x}$  para los que  $d_i(\mathbf{x}) = d_j(\mathbf{x})$  o, expresarlo de forma equivalente, por los valores de  $\mathbf{x}$  para los que

$$d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

[Ecu. 5. 3]

## 5.2 METODOS SINTACTICOS

Los métodos sintácticos se encuentran entre las técnicas más utilizadas en gestión de problemas de reconocimiento de estructuras. Básicamente, la idea del reconocimiento sintáctico de patrones consiste en la especificación de un conjunto de *primitivas* de patrones, un conjunto de reglas (en forma de *gramáticas*) que gobierna su interrelación, y un *reconocedor* (denominado *autómata*) cuya estructura está determinada por el conjunto de reglas que impone la gramática. [9]

- **Aspecto del reconocimiento en el modelo sintáctico.**

Un problema importante para los métodos sintácticos es su sensibilidad a las variaciones en datos. En la mayoría de las aplicaciones existen variaciones en los datos. Esto es debido a las variaciones naturales de la escena, al proceso de adquisición para la automatización del proceso de reconocimiento, o a la extracción de primitivos. El lenguaje que representa una clase en un modelo, deducida de un conjunto de muestras, no es afectado por la estadística de las variaciones, cientos de cadenas idénticas y parecidas a la versión ideal no agregarán más información referente a las reglas de una muestra, para reescribir con relación a las variaciones que tienen con respecto a la referencia ideal. Esto es una diferencia significativa con respecto a los métodos estadísticos donde tales variaciones se modelan explícitamente. Si las variaciones en el resultado son grandes para un modelo existirán problemas en el proceso de comparación. En estos no es posible medir si las clases comparadas tienen algunas semejanzas significativas, por las variaciones abruptas en los datos. [7]

Los lenguajes estocásticos se han introducido como los medios para modelar las variaciones en datos. Una probabilidad es asociada a cada regla al reescribirla y definirá cómo será utilizada en un proceso de generación de cadenas para un lenguaje. El reescribir la regla debido a las variaciones propiciará una probabilidad baja. De acuerdo con la probabilidad de la producción ejercida, existirá una probabilidad para cada cadena y el lenguaje podrá ser computado.

Tres áreas importantes utilizadas para el modelado pueden ser definidas de la siguiente forma:

- i) Descripción de las variaciones del modelo.
- ii) Estrategias de la clasificación para la comparación de clases.
- iii) Control del análisis de la sintaxis de cadenas.

### 5.3 TEORIA DE LENGUAJES, GRAMATICAS Y AUTOMATAS

En el campo de la informática uno de los temas más importantes que la componen es la teoría matemática de lenguajes, gran parte de las tareas en este campo como son la implementación de compiladores, analizadores sintácticos, desarrollo de lenguajes de programación, etc., requieren un conocimiento profundo de esta teoría de lenguajes y por otro lado, muchos problemas computacionales que tienen un gran interés en la actualidad pueden plantearse también en términos lingüísticos. Algunos de forma inmediata, como la traducción entre idiomas. Otros, de forma indirecta: por ejemplo, el reconocimiento de objetos mediante la clasificación de cadenas de símbolos que codifican una imagen, un problema análogo al de identificar palabras de un lenguaje. [10]

- **Representación de un Automata Finito Determinista (AFD).**

Existen tres formas habituales de representar un autómata finito:

1. Gráficamente.
2. Como una tabla de transición.
3. Mediante una función de transición.

La figura 5.3 representa un *grafo dirigido etiquetado*. Un grafo etiquetado consta de:

1. Un conjunto de etiquetas o alfabeto  $\Sigma$ ;
2. Un conjunto  $Q$  de nodos;
3. Para cada símbolo de trabajo  $a \in \Sigma$  un subconjunto  $E_a \subset Q \times Q$  de pares de estados llamados arcos.

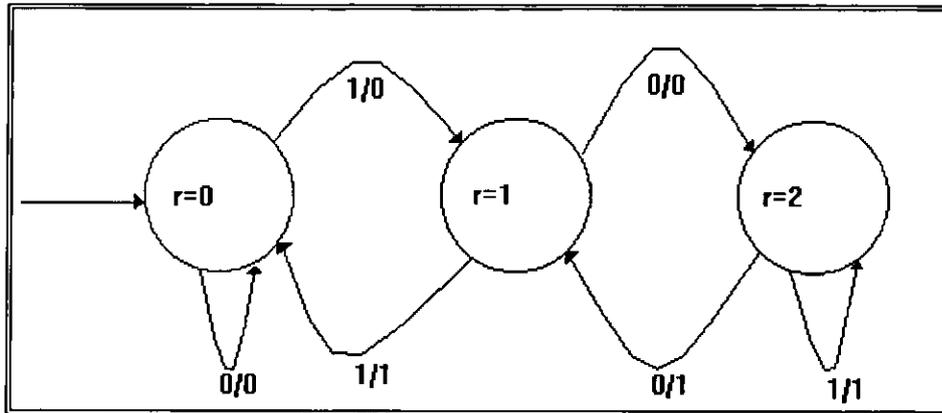


Figura 5.3 Grafo dirigido etiquetado. [G]

Por ejemplo,  $(q_1, q_2) \in E_1$  indica que existe un arco etiquetado con 1 que se dirige desde  $q_1$  hasta  $q_2$ .

La información que contiene este diagrama puede darse también de otra forma: a través de una tabla como la siguiente.

	0	1
$q_1$	$q_1$	$q_2$
$q_2$	$q_3$	$q_1$
$q_3$	$q_2$	$q_3$

La representación en forma de tabla es también una forma válida y utilizada con frecuencia para describir un autómata finito determinista. A este tipo de tablas se les denomina *tabla de transiciones* y contienen los cambios de estado que se produce cuando se procesa un símbolo de cadena de entrada. Para cada símbolo dado, cada estado de partida está relacionado con un estado destino. Por ello, la tabla anterior puede entenderse también como un conjunto de funciones  $\{\delta_0, \delta_1\}$  (una por cada columna) que asocian a  $q'$  también de  $Q$ . Matemáticamente, esto se escribe como  $\delta_k : Q \rightarrow Q$  y el resultado de aplicar la función  $\delta_k$  al estado  $q$  se representa como  $q' = \delta_k(q)$ . Aún es posible utilizar una notación más compacta y definir una función única  $\delta : Q \times \Sigma \rightarrow Q$  que a cada estado  $q$  y a cada símbolo de trabajo  $a \in \Sigma$ , le asocia un estado de llegada  $q' \in Q$ , de forma que  $\delta(q, a) = \delta_a(q)$ . Esta función  $\delta$  es conocida habitualmente como *función de transición*.

La función de transición del ejemplo es:

$$\delta(q_1, 0) = q_1$$

$$\delta(q_1, 1) = q_2$$

$$\delta(q_2, 0) = q_3$$

$$\delta(q_2, 1) = q_1$$

$$\delta(q_3, 0) = q_2$$

$$\delta(q_3, 1) = q_3$$

[Ecu. 5. 4]

Un *autómata finito determinista* funciona efectuando transiciones entre los estados de forma controlada por el símbolo que se está leyendo en cada momento de entrada, de forma que un estado de llegada es a su vez estado de partida para la siguiente transición. Para nuestro ejemplo, esto puede expresarse de forma matemática como  $\delta(q_1, 101) = \delta(q_2, 01) = \delta(q_3, 1) = q_3$ . Observamos que ésta propiedad permite considerar la función sobre cadenas de entrada.

Una última consideración hace referencia al adjetivo determinista, los autómatas que se han considerado son deterministas en el sentido habitual de que sólo existe una opción en cada momento: dada una cadena de entrada sólo existe un posible estado en el que se puede encontrar el autómata tras la lectura, exactamente igual que un ordenador sólo puede encontrarse en una configuración después de ejecutar un programa (secuencia de instrucciones).

- **Alfabetos y Lenguajes.**

Un *alfabeto*  $\Sigma$  es un conjunto finito y ordenado de símbolos. Por ejemplo:  $\Sigma = \{a, b, c, d, e\}$  o el alfabeto binario  $\Sigma = \{0, 1\}$ . El orden de sus elementos se denomina *ordenación alfabética*. Una secuencia de un número arbitrario (finito) de símbolos del alfabeto forma una *cadena, palabra o frase*. Por ejemplo,  $w = aca$  significa que la cadena  $w$  está formada por una secuencia de tres símbolos del alfabeto:  $a$ ,  $c$  y  $a$ . Dos secuencias compuestas de los mismos símbolos pero en diferentes orden son distintas:  $aca \neq caa$ . El número de símbolos que componen la longitud de la cadena se representa por  $|w|$ .

Al conjunto de todas las cadenas generables a partir del alfabeto  $\Sigma$  las llamaremos *lenguaje universal*  $\Sigma^*$  y a la operación por la que se genera una cadena a partir de símbolos que la componen, *concatenación*. La operación de concatenación se define como una operación interna y asociativa en  $\Sigma^*$ , de forma que si  $x$ ,  $y$  y  $z$  son cadenas de  $\Sigma$ , entonces:

$$x(yz) = (xy)z = xyz$$

[Ecu. 5. 5]

Es posible formar cadenas de longitud arbitraria, y en particular, cabe la posibilidad de que la cadena contenga cero símbolos. En este caso, se trata de una *cadena vacía*, que se representará de forma especial mediante el símbolo  $\epsilon$  y que satisface:

$$\forall w \in \Sigma^* : w\epsilon = \epsilon w = w$$

[Ecu. 5. 6]

Con la incorporación de  $\epsilon$ , cuyas características son las de un elemento neutro, la concatenación provee una estructura *monoide* a  $\Sigma^*$ , pues se trata de una operación interna asociativa, con elemento neutro pero no conmutativa. En algunos casos es conveniente considerar  $\epsilon$  como un símbolo especial del alfabeto (extendido) más que como una cadena. Finalmente se trata de un nuevo símbolo que representa algo difícil de escribir, la ausencia de símbolos. Cuando los lenguajes de programación representan cadenas de caracteres, utilizan comillas para marcar su comienzo y su final, en esta notación la cadena vacía sería “ ”.

Por último, un *lenguaje* es un subconjunto cualquiera de cadenas de  $\Sigma^*$ . Por ejemplo, las frases del castellano son un conjunto de todas las cadenas que se pueden formar a partir del alfabeto habitual, las letras especiales (ñ, á, ...), y los signos de puntuación. Pero también llamaremos, por analogía, lenguaje a conjuntos como por ejemplo el de las cadenas binarias que expresan un número par.

Un posible lenguaje puede ser aquél que no contiene ninguna cadena o *conjunto vacío*, que se representa por  $\emptyset$ . Importa distinguir entre este y  $\epsilon$  que es una cadena y como tal puede estar en un conjunto. Por ejemplo,  $\{\epsilon\}$  representa a un conjunto con un solo elemento (en este caso  $\epsilon$ ) y, por tanto, distinto del conjunto vacío.

Los lenguajes que se pueden generar a partir del alfabeto finito o de cardinal infinito *numerable*. En efecto, si un lenguaje es infinito, siempre es posible establecer una ordenación efectiva en él, de forma que a cada cadena se le asigne un número de orden siguiendo la llamada ordenación lexicográfica o canónica. Esta consiste en recorrer las palabras del lenguaje de menor a mayor longitud de cadenas y dentro de cadenas y de cada longitud por el orden alfabético definido para el alfabeto  $\Sigma$ . Por ejemplo, un alfabeto binario seguiría la sucesión  $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$ .

La clasificación de problemas en función de su complejidad (medida ésta en términos de qué mecanismos son necesarios para resolverlo), tiene una analogía con la teoría matemática de lenguajes. Noam Chomsky estableció cuatro categorías diferentes de lenguaje (desde el tipo 0 al tipo 3) basándose en la complejidad de las gramáticas que definían estos lenguajes. [10]

Tipo de lenguaje	Denominación habitual	Mecanismo de
3	regular	autómata finito
2	independiente de contexto	autómata con pila
1	dependiente del contexto	autómata lineal acotado
0	recursivo	máquina turing

Tabla 5.1. Jerarquía de Chomsky para los lenguajes. [G]

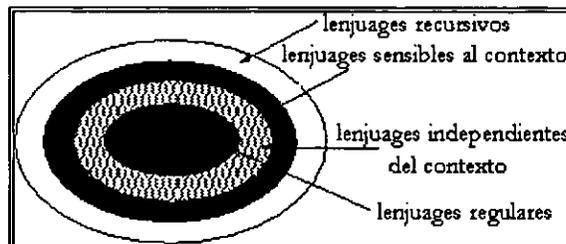


Figura 5.4 Relación de inclusión entre las clases de jerarquías de Chomsky. [G]

- *Autómata finito determinista como clasificador.*

Un autómata finito puede servir para discriminar palabras, observando el estado final en el cuál se encuentra. Por ejemplo, el autómata de la figura 5.5 permite distinguir entre las cadenas de  $\Sigma^* = \{0,1\}^*$  que empiezan con 0 de aquellas que empiezan con 1.

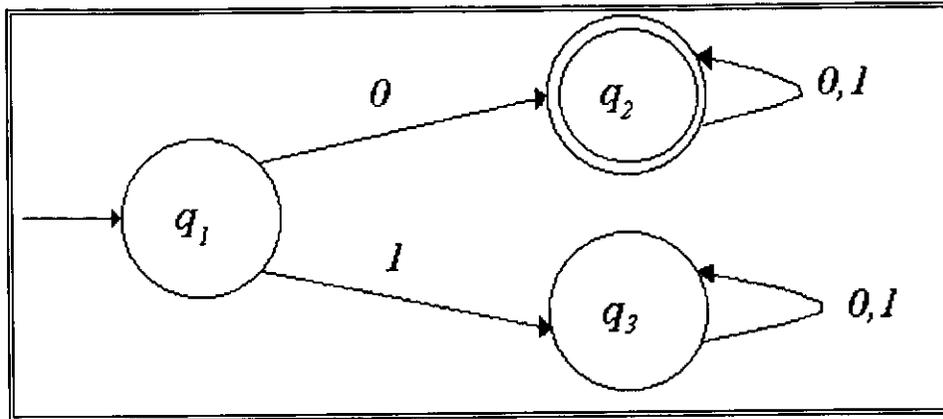


Figura 5. 5 Ejemplo de un autómata finito determinista como clasificador. [G]

El estado inicial (aquel en que se encuentra el autómata antes de analizar la cadena) ha sido señalado con una flecha. De este estado se pasa a otro  $q_2$  o  $q_3$ , al recibir el primer símbolo y permanecer en el mismo nodo independientemente de los símbolos que sean leídos a continuación. Se ha señalado el estado  $q_2$  con doble círculo indicando que el autómata seleccionará como aceptada la cadena que conduzca a dicho estado, que denominaremos *estado de aceptación*.

Siguiendo el planteamiento de un autómata finito, los elementos de un autómata finito determinista (AFD) son  $M = (Q, \Sigma, \delta, q_1, F)$ , siendo:

- $Q = \{q_i\}$ , el conjunto finito de estados posibles del autómata;
- $\Sigma$ , su alfabeto finito de entrada;
- $q_1 \in Q$ , su estado inicial;
- $F \subset Q$ , el subconjunto de estados de aceptación;
- $\delta$ , es una *función de transición* entre estados de la forma  $\delta: Q \times \Sigma \rightarrow Q$  que dicta el comportamiento del autómata.

Con las propiedades de los AFD, se establece un *homorfismo* entre la concatenación de cadenas del monoide  $\Sigma^*$  y la composición de las funciones  $\delta_w : Q \rightarrow Q$  que corresponden a cada cadena  $w$ . Puede comprobarse que con sólo haber exigido esta naturaleza de homorfismo se hubiesen cumplido todas las propiedades que se han indicado.

Como cabe esperar, se define el *lenguaje aceptado* por un AFD como:

$$L(M) = \{w \in \Sigma^* : \delta_w(q_1) \in F\}$$

[Ecu. 5.7]

La definición de lenguaje aceptado viene motivada por la utilización del AFD como reconocedor de lenguajes: el AFD nos permite determinar si una cierta cadena  $w$  pertenece o no a  $L(M)$  simplemente observando si el estado final al que se llega mediante  $w$  es uno de los estados del subconjunto  $F$ . Todos los lenguajes reconocidos mediante autómatas finitos deterministas reciben el nombre de *lenguajes regulares*

- ◆ *Autómata finito indeterminista (AFI)*. Un (AFI) es una generalización de los AFD donde se permite que de un nodo parta más de una transición etiquetada con el mismo símbolo, o que exista más de un estado inicial. Esta flexibilidad podría aumentar, aparentemente, la potencia de los autómatas y permitir el reconocimiento de conjuntos que no son analizables mediante un AFD. Sin embargo, veremos inmediatamente que esto no es así, y que la capacidad de análisis de los AFI es exactamente la misma que la de los AFD.
- ◆ *Autómatas finitos estocásticos (AFE)*. Un (AFD) asocia a cada cadena de  $\Sigma^*$  una cierta probabilidad de aparición, con lo que el lenguaje es generado en orden aleatorio, pudiendo aparecer las cadenas repetidas veces. Este tipo de comportamiento es importante si se quiere reproducir algún proceso real. Con frecuencia las cadenas con las que se trabaja son generadas por fuentes con comportamiento aleatorio, o sufren efectos de ruido, etc. Si no se da un tratamiento estadístico a estos lenguajes, su reconocimiento y análisis se vuelve difícil. Por ello conviene introducir en el modelo la probabilidad de que aparezca cada una de las cadenas. En el caso de los AFE, la probabilidad se asigna a cada una de las transiciones y a los estados de aceptación.

### 5.3.1 Expresiones regulares.

Una forma de representar algunos subconjuntos de  $\Sigma^*$  de forma precisa consiste en el uso de expresiones regulares. Una *expresión regular* no es más que un tipo de notación matemática para representar lenguajes. Los conjuntos representables mediante una expresión regular reciben el nombre de *conjuntos regulares*. Aunque la definición de estos conjuntos no parece guardar ninguna relación con la forma en que se introdujeron los lenguajes regulares, llegaremos a la interesante conclusión de que estamos hablando exactamente de los mismos objetos. [10]

Las expresiones regulares, que utilizaremos para caracterizar los conjuntos regulares, pueden ser definidas recursivamente. Si escribimos  $L(r)$  para el conjunto denotado por una expresión  $r$ , entonces:

- $\emptyset$  es una expresión regular que denota el lenguaje vacío ( $L(\emptyset) = \emptyset$ );
- $\varepsilon$  es una expresión regular que denota el lenguaje que sólo contiene la cadena vacía ( $L(\varepsilon) = \{\varepsilon\}$ );
- Si  $a$  es un símbolo de  $\Sigma$ , entonces  $a$  es una expresión regular que denota el lenguaje que contiene la cadena  $a$  ( $L(a) = \{a\}$ ).

Estas tres primeras reglas (cláusulas iniciales de la definición recursiva) definen las expresiones regulares básicas.

Podemos decir que los *conjuntos regulares* son los subconjuntos de  $\Sigma^*$  que pueden ser definidos con expresiones regulares. Es decir, un subconjunto de  $\Sigma^*$  es regular si y sólo si es  $\emptyset$ ,  $\{\varepsilon\}$ ,  $\{a\}$  para algún símbolo  $a \in \Sigma$ , o puede ser obtenido a partir de estos conjuntos mediante un número finito de aplicaciones de las operaciones de unión, concatenación o clausura de Kleene. [10]

Algunos ejemplos de expresiones regulares sobre  $\Sigma=\{0,1\}$  son:

- $0^*$  (conjunto de todas las cadenas de ceros)
- $(0+1)^*$  (conjunto de cadenas formadas por ceros y unos, es decir,  $\Sigma^*$ )
- $(10)^*$  (cadenas formadas por secuencias de 110, incluyendo  $\varepsilon$ )
- $(1+0)^*10^*$  (cadenas que contienen al menos un 1)
- $(0+1)^*00(0+1)^*$  (cadenas con dos ceros consecutivos.)
- $(1+10)^*$  (cadenas que no contienen dos ceros consecutivos y empiezan con uno)

Algunas propiedades de las expresiones regulares (ER) son:

- el operador  $+$  es asociado y conmutativo
- el operador concatenación es asociativo
- $r(s+t) = rs + rt$  (distributiva por la izquierda de la concatenación sobre la unión)
- $(rs+t)t = rt+st$  (distributiva por la derecha de la concatenación sobre la unión)
- $(r^*)^* = r^*$  (idempotencia).
- $r+r=r$
- $r+\emptyset=r$
- $r\varepsilon=\varepsilon r=r$
- $r\emptyset=\emptyset r=\emptyset$
- $\varepsilon^*=\varepsilon$
- $\emptyset^*=\varepsilon$
- $r^*r=rr^*=r^+$
- $r^++\varepsilon=r^*$
- $r^*+\varepsilon=r^*$
- $r\emptyset s \Rightarrow r^*\emptyset s^* \Rightarrow r^*+s^*=(r+s)^*=s^*$
- $(r+\varepsilon)^*=r^*$
- $s(rs)^*=(sr)^*s$
- $s(rs)^+(sr)^+s$
- $(r+\varepsilon)^*=(r^*s^*)^*$
- $(r+rs)^*r=r(r+sr)^*$

Se tienen implementaciones básicas para construir un algoritmo de búsqueda de cadenas que representan una expresión regular y por medio de estas comenzaremos a construir un autómata finito.

- Por cada símbolo de la cadena que no es un metacarácter (paréntesis, barra o estrella) añade dos estados conectados por un símbolo.

- Por cada concatenación de expresiones añade una transición vacía entre el último estado de la primera y el primero de la segunda, en analogía de como se realiza la conexión en serie de dos circuitos eléctricos.
- Por cada unión de expresiones añade un nuevo estado de entrada y otro de salida. Conecta el primero con dos transiciones vacías a los primeros de cada expresión. A su vez, los dos últimos de cada expresión deben conectarse mediante transiciones vacías con el nuevo estado de salida, en analogía con una conexión en paralelo de dos circuitos.
- Para cada clausura de Kleene añade un nuevo estado de entrada que también será el de salida. Conecta este estado con el inicial del autómata correspondiente a la expresión y el de salida de éste con el nuevo estado mediante transiciones vacías.

Ejemplos: Se quiere diseñar autómatas que aceptan  $\emptyset$ ,  $\{\epsilon\}$  y  $\{a \in \Sigma\}$ , siguiendo las reglas descritas tendremos:

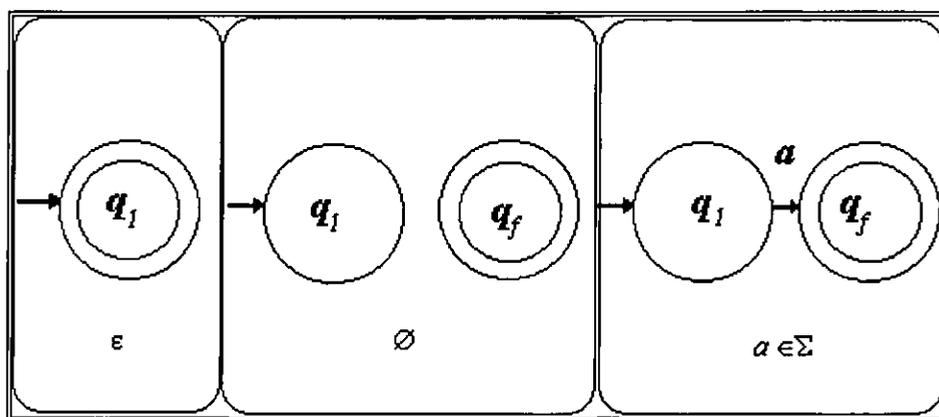


Figura 5. 6 Autómatas finitos para  $\epsilon$ ,  $\emptyset$ , y  $a \in \Sigma$ . [G]

Existen operaciones bajo cuya aplicación de los conjuntos regulares son estables, es decir, el conjunto que se obtiene como resultado es también regular, ejemplo de estas operaciones son:

- unión
- complementación
- intersección
- sustitución
- cociente
- inversión

### 5.3.2 Gramáticas.

Al definir una gramática esta debe dar reglas con las que se generan cadenas o frases pertenecientes al lenguaje que caracteriza, permitiendo, asimismo, el reconocimiento de una cadena arbitrariamente dada como perteneciente o no a ésta.

**Definición:** Una gramática es una cuádrupla:

$$G = (V_N, V_T, S, P)$$

[Ecu. 5. 8]

donde  $V_N$  es un conjunto finito de símbolos variables (no terminales);  $V_T = \Sigma$ , un conjunto finito de símbolos terminales;  $S \in V_N$ , el símbolo o variable inicial; y  $P$ , un conjunto finito de reglas de derivación.

Lógicamente se debe verificar que  $V_N$  y  $V_T$  sean disjuntos:  $V_N \cap V_T = \emptyset$ . Usaremos  $V$  para denotar al conjunto de todos los símbolos tanto variables como terminales:  $V = V_N \cup V_T$ . Las reglas de derivación  $r \in P$  indican como sustituir variables y pueden expresarse como aplicaciones,

$$r: V^* V_N V^* \rightarrow V^*,$$

[Ecu. 5. 9]

donde  $V^*$  denota al conjunto de todas las cadenas que se pueden formar con los símbolos de  $V$ . Es decir, las reglas transforman cadenas de  $V^*$  que contienen al menos una variable en cadenas de  $V^*$ . Por lo general las variables se representaran por letras mayúsculas,  $V_N = \{A, B, C, \dots\}$ , los terminales por letras minúsculas,  $V_T = \{a, b, c, \dots\}$  y las cadenas de  $V^*$  por letras griegas ( $\alpha, \beta, \gamma, \dots$ ).

Las cadenas generadas por la gramática son aquellas que se pueden obtener partiendo del símbolo inicial  $S$  mediante la aplicación sucesiva de reglas de  $P$ , este conjunto de cadenas derivables desde  $S$  se denomina *lenguaje generado por la gramática  $G$* . Simbólicamente,

$$L(G) = \{w \in V_T^* : S \xrightarrow{*} w\},$$

[Ecu. 5. 10]

donde el asterisco indica que la derivación se puede realizar en un número arbitrario de pasos. Concretamente, el símbolo  $\xrightarrow{*}$  representa a la relación reflexiva y transitiva más pequeña que contiene a  $\rightarrow$ , esto es,  $\xrightarrow{*}$  satisface, para cualquier cadena  $\alpha, \beta, \gamma \in V^*$ :

$$\begin{aligned} \alpha \rightarrow \beta &\Rightarrow \alpha \xrightarrow{*} \beta \\ \alpha &\xrightarrow{*} \alpha \\ \alpha \xrightarrow{*} \beta \wedge \beta \rightarrow \gamma &\Rightarrow \alpha \xrightarrow{*} \gamma \end{aligned}$$

[Ecu. 5. 11]

Diremos que dos gramáticas son equivalentes si generan el mismo lenguaje, es decir,

$$G \equiv G' \Leftrightarrow L(G) = L(G')$$

[Ecu. 5. 12]

Según la complejidad de las reglas que definen una gramática, ésta puede ser uno de los cuatro tipos ( $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ ) que constituyen la llamada *jerarquía de Chosmsky*:

- ◆ Gramáticas estructuradas por frase o de tipo 0: son las más generales. Sus reglas no tienen ninguna restricción. Generan los llamados *lenguajes recursivamente numerables* o *lenguajes sin restricciones*, aunque a pesar de este nombre no son capaces de representar a los lenguajes naturales.
- ◆ Gramáticas sensibles al contexto o de tipo 1: sus reglas son de la forma  $\alpha \rightarrow \beta$  ( $\alpha, \beta \in V^*$ ,  $|\alpha| \leq |\beta|$ ,  $\alpha$  contiene al menos una variable), es decir, las derivaciones sucesivas no pueden disminuir la longitud de la cadena de símbolos anteriores. Se puede demostrar que este tipo de reglas es equivalentes a las de la forma  $\alpha A \beta \rightarrow \alpha \gamma \beta$ , ( $\gamma \in V^+$ ), de ahí el nombre de este tipo de gramática ( la sustitución de  $A$  por  $\gamma$  depende del contexto en el que aparece  $A$  ).
- ◆ Gramáticas independientes del contexto o de tipo 2: sus reglas tienen la forma  $A \rightarrow \alpha$ , ( $A \in V_N$ ,  $\alpha \in V^*$ ). A diferencia del tipo anterior, la sustitución de  $A$  por  $\alpha$  se realiza independientemente del lugar en el que aparezca  $A$ . La mayor parte de los lenguajes de programación están generados por gramáticas de este tipo (aumentadas con algunos elementos contextuales necesarios para la semántica del lenguaje); de ahí la importancia de su estudio.
- ◆ Gramáticas regulares o de tipo 3: son las más simples, sus reglas son de uno de los siguientes tipos:  $A \rightarrow aB$ ,  $A \rightarrow a$ ,  $A \rightarrow \varepsilon$ , ( $A, B \in V_N$ ,  $a \in V_T$ ), y a continuación generan los mismos lenguajes que son reconocidos por un autómata finito. [10]

• *Gramática regular asociada a un autómata.*

Partiendo del autómata de la figura 5.7 se puede comprobar que el lenguaje reconocido por este autómata es  $(0+1)^*10$ , el conjunto de cadenas sobre el alfabeto  $0,1$  que acaban en  $10$ .

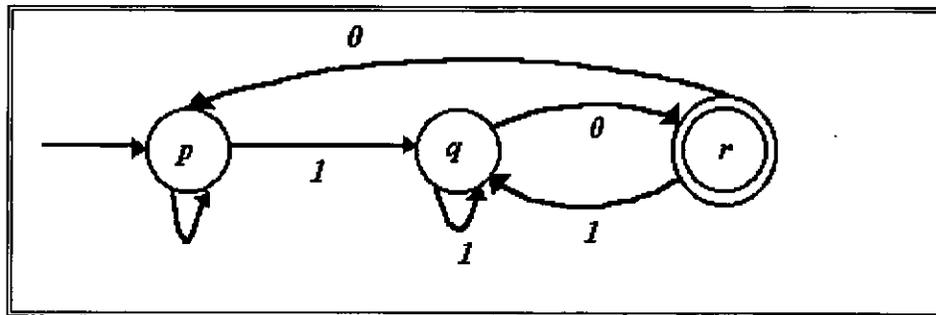


Figura 5.7 Autómata que reconoce el lenguaje  $(0+1)^*10$  [G]

Supongamos que tomamos una cadena  $1001110$  de este lenguaje, recorriendo la secuencia podemos describir el estado del autómata, escribiendo la subcadena leída en cada instante. Obtendremos la secuencia:

$$p \rightarrow 1p \rightarrow 10r \rightarrow 100p \rightarrow 1001q \rightarrow 10011q \rightarrow 100111q \rightarrow 100110$$

[Ecu. 5.13]

que puede interpretarse como una generación de la cadena en una gramática. Para obtener la gramática especificamos las variables como los estados del AFD, y a partir de los arcos obtenemos las reglas de reescritura:

$$p \rightarrow 0p|1q$$

$$q \rightarrow 0r|1q$$

$$r \rightarrow 0p|1q$$

[Ecu. 5. 14]

El símbolo inicial de la gramática representa al estado inicial del autómata. Para completar el proceso hay que conseguir que la derivación acabe cuando en el autómata se alcanza un estado de aceptación esto se logra en nuestro ejemplo añadiendo la producción,

$$q \rightarrow 0$$

[Ecu. 5. 15]

o alternativamente,

$$r \rightarrow \varepsilon$$

[Ecu. 5. 16]

Observemos que todas las producciones que hemos necesitado son una de las formas que corresponden a una gramática regular.

Resumiendo. Dado un autómata  $M = (Q, \Sigma, \delta, q_1, F)$ , obtenemos una gramática  $G = (V_N, V_T, P, S)$  tal que  $L(M) = L(G)$ .

- **Gramáticas independientes de contexto (GIC).**

La gran importancia de las *gramáticas independientes del contexto* o *gramáticas incontextuales* reside en el hecho de que la mayor parte de los lenguajes de programación se representan mediante gramáticas regulares y se caracterizan porque sus reglas son de la forma:

$$r: V_T \rightarrow V^*$$

[Ecu. 5. 17]

es decir, la parte derecha de la regla puede ser cualquier cadena de símbolos terminales y no terminales, si bien la parte izquierda está restringida a una sola variable. Un ejemplo de GIC estaría dado por:

$$S \rightarrow 0B|1A$$

$$A \rightarrow 0|0s|1AA$$

$$\bar{B} \rightarrow 1|1s|0BB$$

[Ecu. 5. 18]

- **Derivación y árbol de derivación.**

A la secuencia de reglas que nos permite obtener una cadena del lenguaje generado por la gramática a partir de su símbolo inicial, se le conoce con el nombre de *derivación de la cadena*. Por ejemplo si se tiene la cadena,

$$E \rightarrow E + E | E^* E | (E) \quad a$$

[Ecu. 5. 19]

la secuencia

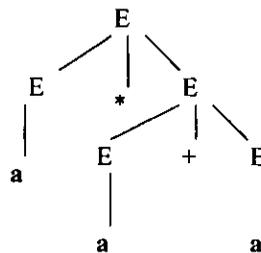
$$E \rightarrow E^* E \rightarrow E^* E + E \rightarrow E^* E + a \rightarrow E^* a + a \rightarrow a^* a + a$$

[Ecu. 5. 20]

constituye una derivación de la cadena  $a^*a+a$ . La cadena  $De V^*$  obtenida en cada paso de la derivación recibe el nombre de *forma sentencial*.

Un *árbol* de derivación es una representación gráfica de la secuencia de reglas aplicadas (es fácil comprobar que las gramáticas tipo 0 no admiten esta representación). Más formalmente, un árbol de derivación se define como un *árbol ordenado* que cumple con los siguiente puntos y diagrama:

- cada nodo interior (con descendencia) está etiquetado con un símbolo de  $V_N$ ;
- el símbolo inicial  $S$  de la gramática ocupa el nodo raíz;
- los nodos terminales (sin descendencia) u *hojas* sólo contienen símbolos de  $V_T$  tal vez, la cadena vacía  $\epsilon$ ;
- si un nodo contiene el símbolo  $A \in V_N$  y presenta como descendientes en este orden, a los nodos  $X_1, X_2, \dots, X_n$ , entonces existe una regla en la gramática  $G$  en la que la variable  $A$  es sustituida por  $X_1, X_2, \dots, X_n$ .



A la hora de reconstruir la derivación a partir del árbol, se nos presentan varias alternativas según el orden en el que efectuemos las sustituciones de las diversas variables; para evitar la ambigüedad en la construcción del árbol se utilizan dos de las reglas anteriores, llamadas *derivaciones por la izquierda* y *derivación por la derecha*. La diferencia entre ambas reside en decidir que variables es sustituida en cada paso de la derivación, la que está más a la izquierda o la que está más derecha en la forma sentencial. Para el árbol anterior estas derivaciones serían:

1. Por la izquierda:  $E \rightarrow E^* E \rightarrow a^* E \rightarrow a^* E + E \rightarrow a^* a + E \rightarrow a^* a + a$
2. Por la derecha:  $E \rightarrow E^* E \rightarrow E^* E + E \rightarrow E^* E + a \rightarrow E^* a + a \rightarrow a^* a + a$

- *El análisis sintáctico.*

Para decidir si una cadena dada pertenece al lenguaje generado por una GIC particular, debemos ser capaces de reconstruir el árbol de derivación de la cadena. A este proceso se le llama análisis sintáctico, y su dominio es, por otra parte, el principal objetivo que debemos plantearnos en el estudio de la GIC. Un algoritmo de uso general, válido tanto para gramáticas ambiguas como no ambiguas, es llamado Algoritmo de *Coke, Younger y Kasami (CYK)*. [10]

### 5.3.3 Función del análisis sintáctico.

Analizar sintácticamente una tira o cadena de pruebas no es más que encontrar para ella el árbol sintáctico o de derivación que tiene como raíz el axioma de la gramática, y como nodos terminales la sucesión ordenada de símbolos que componen la cadena analizada. En caso de no existir este árbol sintáctico, la cadena no pertenecerá al lenguaje, y el analizador sintáctico ha de emitir el correspondiente mensaje de error. Existen dos formas de analizar sintácticamente una cadena:

- **Análisis descendente:** Partiendo del axioma inicial de la gramática se va descendiendo utilizando las derivaciones izquierdas, hasta llegar a construir la cadena analizada.
- **Análisis ascendente:** Se va construyendo el árbol desde sus nodos terminales. Es decir, se construye desde los símbolos de la cadena hasta llegar al axioma de la gramática. En este caso, se emplean normalmente las derivaciones más a la derecha hasta la localización de la raíz.

Los principales métodos de análisis sintáctico son:

- **Análisis descendente:**
  - Análisis descendente con retroceso.
  - Análisis de gramáticas LL (*Left-to-right, Leftmost derivation*).
- **Análisis ascendente:**
  - Análisis ascendente con retroceso.
  - Análisis de gramáticas de precedencia simple.
  - Análisis de gramáticas de precedencia generalizada.
  - Análisis de gramáticas por el método mixto.
  - Análisis de gramáticas de precedencia de operador.
  - Análisis de gramáticas LR (*Left-to-right, Rightmost derivation*). [15]

Los análisis con retroceso se basan en la prueba sistemática de todas las alternativas posibles, dando marcha atrás tan pronto como se detecte que el camino seguido es erróneo. Pueden usarse para cualquier gramática de contexto libre, aunque tienen tres grandes inconvenientes: Primero, emplean mucho más tiempo para el análisis que los demás analizadores, dependiendo éste incluso de la ordenación de las reglas gramaticales; Segundo, no dan un buen diagnóstico de los errores que encuentran; Tercero, complican la generación del código cuando ésta se realiza paralelamente al análisis sintáctico.

Los métodos más eficientes de análisis (tanto ascendente como descendente) no funcionan para todas las gramáticas de contexto libre, sino sólo para las gramáticas que cumplen unas determinadas condiciones.

Afortunadamente, en la mayoría de los casos, pueden encontrarse para los lenguajes de programación gramáticas de tipo LL o LR que los generen.

Para representar el árbol sintáctico que conduce hasta una cadena se asigna a cada regla de la gramática un número. Se define el *parse* como la secuencia ordenada de números (reglas que describen la gramática de un lenguaje) aplicada para construir dicho árbol. [15]

Hay dos tipos de *parse*, que son:

1. El *parse-izquierdo*: Son los números de las reglas de derivación izquierda utilizadas para generar la cadena a partir del axioma, por tanto, correspondiente a un análisis descendente.
2. El *parse-derecho*: Son los números de las reglas de derivación derecha utilizadas para generar la cadena a partir del axioma en orden inverso. El tomar el orden inverso viene condicionado por ser el análisis ascendente el que normalmente utiliza las reglas de derivación derecha, con lo que el orden en el que aparecen al realizar el análisis es invertido.

Ejemplos:

Dada la gramática

1.  $E \rightarrow T$
2.  $E \rightarrow T+E$
3.  $T \rightarrow F$
4.  $T \rightarrow F*T$
5.  $F \rightarrow a$
6.  $F \rightarrow b$
7.  $F \rightarrow (E)$

y la sentencia "a\*(a+b)", el parse izquierdo es: 1-4-5-3-7-2-3-5-1-3-6 y el derecho: 5-5-3-6-3-1-2-7-3-4-1.

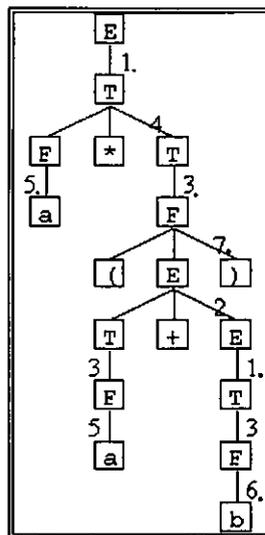


Figura 5. 8 [G]

Simultáneamente a la fase de análisis sintáctico, además de reconocer las secuencias de pruebas, y analizar su estructura, pueden realizarse una serie de tareas adicionales, como:

- Recopilar información de los distintos pruebas y almacenarla en la tabla de símbolos.
- Realizar algún tipo de análisis semántico, tal como la comprobación de tipos.
- Generar código intermedio.
- Avisar de los errores que se detecten.

### 5.3.3.1 Tipos de análisis sintáctico.

#### Análisis descendente con retroceso.

Si bien en la práctica difícilmente se usa, se estudia a continuación el análisis descendente con retroceso, por ser el más general de los análisis descendentes, y por su valor didáctico. Para la realización del análisis descendente con retroceso contamos con una gramática, cuyas reglas se enumeran:

(1)  $A \rightarrow \alpha_1$  (2)  $A \rightarrow \alpha_2 : : : B \rightarrow \beta_1 : B \rightarrow \beta_2 : B \rightarrow \beta_3 : : : C \rightarrow \gamma_1 : : :$

y de una cadena, supuestamente perteneciente al lenguaje definido por la gramática:  $a_1 a_2 a_3 a_4 \dots$

- Paso 1. Inicialmente se parte de la forma sentencial  $S$ , que contiene únicamente el axioma de la gramática.
- Paso 2. Dada una forma sentencial generada anteriormente, se sustituye el símbolo no terminal situado más a la izquierda por la primera de las alternativas posibles para dicho símbolo, generando así una nueva forma sentencial. Anotamos en una pila el número de la regla empleada. En caso de que no existiera ningún símbolo no terminal en la forma sentencial, se habría llegado a producir la sentencia que estábamos analizando, estando almacenado su parse izquierdo en la pila; si no, se continúa en el paso 4.
- Paso 3. Se anula la última producción utilizada (situada en la cima de la pila), volviendo a obtener la forma sentencial anterior, y se retira de la pila el último número. En esta forma sentencial se utiliza ahora la siguiente de las alternativas disponibles para el símbolo no terminal situado más a la izquierda, anotando en la pila el número de la regla empleada. En caso de que no haya más alternativas, se repite el paso 3 para la forma sentencial obtenida.

Si llega el momento en que se agotan todas las posibles alternativas del axioma inicial, (la pila se habría quedado vacía), entonces se puede garantizar que la cadena analizada no pertenece al lenguaje.

- Paso 4. Se comprueba si todos los símbolos terminales consecutivos que hayan aparecido a la izquierda de la forma sentencial encajan con alguna subcadena izquierda de la tira analizada. En caso afirmativo, se va al paso 2. Si no, se va al paso 3.

Ejemplo:

Sea la gramática:

(1)  $S \rightarrow cAd$

(2)  $A \rightarrow bcB$

(3)  $A \rightarrow a$

(4)  $B \rightarrow b$

A continuación, se va a realizar el análisis (paso a paso) de la sentencia cad:

- Paso 1: Inicialmente se parte de la forma sentencial  $S$ .
- Paso 2: El símbolo no terminal situado más a la izquierda es  $S$ , en las reglas de la gramática la primera (y única) alternativa para el símbolo  $S$  es  $S \rightarrow cAd$ . Empleando esta regla se obtiene la forma sentencial  $cAd$ . La pila contiene el número 1. Ir al paso 3.
- Paso 3: La subcadena de símbolos terminales consecutivos que han aparecido a la izquierda de la forma sentencial es  $c$  y la tira a analizar es  $cad$ . Por tanto, se supone válida esta producción y se continúa por el paso 2.
- Paso 2: Ahora la forma sentencial es  $cAd$ , cuyo primer símbolo no terminal es  $A$ , y la primera de las producciones para  $A$  es  $A \rightarrow bcB$ . Aplicándola se obtiene la nueva forma sentencial  $cbcBd$ . La pila de análisis contiene ahora el parse 1-2.
- Paso 4: La subcadena izquierda de símbolos terminales es ahora  $cbc$ , que no encaja con el principio de la tira analizada. Por tanto, vamos al paso 3.
- Paso 3: Se anula la producción anterior, obteniendo nuevamente la forma sentencial  $cAd$ . Retiramos el símbolo superior de la pila de análisis, quedando ésta como antes (es decir, 1). Se aplica ahora la segunda de las alternativas del símbolo no terminal  $A$ , que es  $A \rightarrow a$ , obteniéndose la forma sentencial  $cad$ . En la pila de análisis se introduce el número de la regla aplicada quedando ésta como 1-3.
- Paso 4: La subcadena izquierda de símbolos terminales es  $cad$ , que coincide exactamente con la tira analizada.
- Paso 2: La forma sentencial no incluye ningún símbolo no terminal, por lo que el proceso termina satisfactoriamente. El parse izquierdo de la cadena está contenido en la pila.

Es evidente que una condición necesaria para realizar este tipo de análisis, tal como se ha descrito anteriormente, es que la gramática no sea recursiva por la izquierda y, en general, que no tenga ciclos por la izquierda. Por ejemplo, supongamos la gramática:  $A \rightarrow Aa \mid a$

Según el método que se sigue para el análisis, se generaría la siguiente secuencia de formas sentenciales :  $A \Rightarrow Aa \Rightarrow Aaa \Rightarrow Aaaa \Rightarrow \dots$  Igualmente sucede si la gramática tiene ciclos, ya que por definición de ciclo:  $A \Rightarrow^+ A\dots$  Para solucionar este problema podemos hallar una gramática equivalente no recursiva por la izquierda y sin ciclos, o bien modificar ligeramente el algoritmo.

La implementación de este algoritmo, mediante el uso de un procedimiento recursivo para la gramática:

1.  $E \rightarrow T$
2.  $E \rightarrow T+E$
3.  $T \rightarrow F$
4.  $T \rightarrow F*T$
5.  $F \rightarrow a$
6.  $F \rightarrow b$
7.  $F \rightarrow (E)$

y para el reconocimiento de la cadena,  $a^*(a+b)$ .

da como resultado la siguiente secuencia.

Pila	Formas Sentenciales
1-	T
1-3-	F
1-3-5-	a
1-3-6-	b
1-3-7-	(E)
1-4-	F*T
1-4-5-	a*T
1-4-5-3-	a*F
1-4-5-3-5-	a*a
1-4-5-3-6-	a*b
1-4-5-3-7-	a*(E)
1-4-5-3-7-1-	a*(T)
1-4-5-3-7-1-3-	a*(F)
1-4-5-3-7-1-3-5-	a*(a)
1-4-5-3-7-1-3-6-	a*(b)
1-4-5-3-7-1-3-7-	a*((E))
1-4-5-3-7-1-4-	a*(F*T)
1-4-5-3-7-1-4-5-	a*(a*T)
1-4-5-3-7-1-4-6-	a*(b*T)
1-4-5-3-7-2-	a*(T+E)
1-4-5-3-7-2-3-	a*(F+E)
1-4-5-3-7-2-3-5-	a*(a+E)
1-4-5-3-7-2-3-5-1-	a*(a+T)
1-4-5-3-7-2-3-5-1-3-	a*(a+F)
1-4-5-3-7-2-3-5-1-3-5-	a*(a+a)
1-4-5-3-7-2-3-5-1-3-6-	a*(a+b)

En la columna derecha se muestran las formas sentenciales que se van obteniendo durante el análisis, y en la izquierda, los números de las reglas utilizadas para producirlas (es decir, el parse izquierdo de cada una de estas formas senteciales). [15]

El último valor obtenido en la columna izquierda es el parse izquierdo de la cadena analizada.

- **Procedimientos de análisis LL.**

Análisis descendente sin retroceso.

En el análisis descendente con retroceso se generan formas sentenciales a partir del axioma, dando marcha atrás en cuanto se detecta que la forma generada no es viable, es decir, no conduce a ninguna sentencia del lenguaje. Este proceso de vuelta atrás es lento. Para mejorar la eficiencia del mismo, sería muy útil saber a priori qué alternativa del símbolo no terminal es más conveniente usar.

Veamos de nuevo el ejemplo del apartado anterior:

Ejemplo: Sea la gramática

- (1)  $S \rightarrow cAd$
- (2)  $A \rightarrow bcB$
- (3)  $A \rightarrow a$
- (4)  $B \rightarrow b$

y la sentencia "cad". Partiendo del axioma, sólo se puede aplicar la regla 1, obteniendo la forma sentencial "cAd". Si se compara con la sentencia "cad", se observa que ambas comienzan con el carácter "c". Por tanto, la subcadena "Ad" ha de generar el resto de la sentencia, o sea, "ad". En este instante existen dos alternativas que se pueden emplear para modificar la forma sentencial, que corresponden a la aplicación de las reglas 2 y 3.

La aplicación de la regla 2 provoca la aparición del carácter b al principio de la subcadena restante, mientras que la regla 3 provoca la aparición del carácter a. Por tanto, como la subcadena que falta por generar para producir la sentencia final es "ad" (empieza por "a"), puede deducirse que en este instante la regla que debe emplearse es la regla 3, y no la 2.

Desafortunadamente, hay casos en los que este procedimiento no sirve. Supóngase, por ejemplo, que la gramática fuese:

- (1)  $S \rightarrow cAd$
- (2)  $A \rightarrow aB$
- (3)  $A \rightarrow a$
- (4)  $B \rightarrow b$

Al analizar la tira de entrada "cad", tras realizar la primera producción obtendríamos la forma sentencial "cAd", quedando como subcadena a analizar ad (que comienza con "a"). Pero ahora hay dos reglas aplicables que comienzan por "a" (las reglas número 2 y 3). Por tanto, no es posible decidir de forma automática qué regla debe emplearse. Por otra parte, si se pretende que el análisis sea sin retroceso, es indispensable que la gramática no tenga ciclos por la izquierda (y consiguientemente, que no sea recursiva por la izquierda).

No todas las gramáticas admiten un análisis descendente sin retroceso en el que se pueda predecir la alternativa que debe usarse.

## 5.4 RECONOCIMIENTO SINTACTICO DE CADENAS

- *Gramática de cadenas.* Supóngase que tenemos dos clases,  $w_1$  y  $w_2$ , cuyos patrones son cadenas de primitivas. podemos interpretar que cada primitiva es un símbolo permitido del *alfabeto* de una determinada gramática, siendo una gramática de conjuntos de reglas sintácticas (de aquí viene el nombre de reconocimiento sintáctico) que controla la generación de sentencias generadas por la gramática,  $G$ , se denomina *lenguaje* de dicha gramática, y se indica mediante  $L(G)$ . Aquí, las sentencias son cadenas de símbolos (que, a su vez, representan patrones), y los lenguajes se corresponden con las clases de patrones.

Considérese dos gramáticas,  $G_1$  y  $G_2$ , cuyas reglas sintácticas son tales que  $G_1$  sólo permite la generación de sentencias correspondientes a los patrones de la clase  $w_1$ , y que  $G_2$  sólo permite la generación de sentencias correspondientes a los patrones de la clase  $w_2$ . Después de definir dos gramáticas con estas propiedades, en principio, el proceso sintáctico de reconocimiento es directo.

En la clasificación multiclase, un patrón pertenece a la clase  $w_i$  si representa una sentencia válida sólo en  $L(G)$ . Cuando trabajamos con cadenas, podemos definir una gramática como la 4-tupla:

$$G = (N, \Sigma, P, S)$$

[Ecu. 5. 21]

donde:

- $N$ =Conjunto finito de variables denominadas no terminales, las letras mayúsculas, A,B, ... , S, ....., representan *no terminales*.
- $\Sigma$ =conjunto finito de constantes denominadas *terminales*, las letras minúsculas  $a,b,c,\dots$ , del inicio del alfabeto se utilizan para representar terminales y las letras finales del alfabeto,  $v,w,x,y,z$ , indican las cadenas de terminales. Las letras griegas minúsculas  $\alpha,\beta,\theta,\dots$ , representan cadenas terminales y no terminales mezcladas.
- $P$ = Conjunto de reglas de reescritura denominadas producciones.
- *La sentencia vacía ( la sentencia sin símbolos )* está representada por  $\lambda$ .
- Por último, para un conjunto de símbolos  $V$ , la notación  $V^*$  representa el conjunto de todas las sentencias compuestas de elementos de  $V$ .

Una restricción es que  $N$  y  $\Sigma$  deben de ser conjuntos disjuntos. Las gramáticas de cadenas se caracterizan por la forma de sus producciones. En el reconocimiento sintáctico de patrones son de especial interés las gramáticas regulares y las gramáticas libres de contexto. [1]

Ejemplo: Supongamos que se representa el objeto de la figura 5.9 mediante su esqueleto, y definiendo las primitivas, para describir la estructura de este esqueleto ( y de otros similares). Consideremos la gramática  $G=(N,\Sigma,P,S)$ , con  $N=\{A,B,S\}$ ,  $\Sigma=\{a,b,c\}$  y  $P=\{S \rightarrow aA, A \rightarrow bA, A \rightarrow bB, B \rightarrow c\}$ , donde los terminales  $a, b$  y  $c$  se corresponden con las primitivas, "S" es el símbolo inicial a partir del cual se generan las cadenas de  $L(G)$ . Por ejemplo, si se aplica la primera producción y después se aplica dos veces la segunda, se obtiene  $S \Rightarrow aA \Rightarrow abA \Rightarrow abbA$ , donde ( $\Rightarrow$ ) indica una derivación de cadenas empezando por la S y utilizando

producciones del conjunto  $P$ . La primera producción permite reescribir  $S$  como  $aA$ , y la segunda permite reescribir  $A$  como " $bA$ ". Como hay un no terminal en la cadena " $abbA$ ", es posible continuar la derivación. Por ejemplo, si se aplica dos veces más la segunda producción, y a continuación se aplica la tercera y cuarta, se obtiene la cadena " $abbbbc$ ", correspondiente a la estructura de la figura 5.9 Después de aplicar la cuarta producción no queda ningún no terminal, por lo que la derivación finalizada al utilizar esta producción. El lenguaje generado por las reglas de esta gramática es  $L(G)=\{ab^nc \mid n \geq 1\}$ , donde  $b^n$  representa " $n$ " repeticiones del símbolo " $b$ ". En otras palabras, " $G$ " sólo es capaz de generar esqueletos como el de este ejemplo, pero de longitud arbitraria.

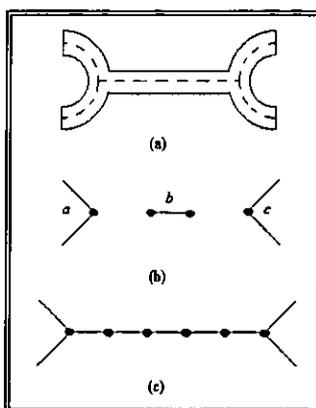


Figura 5. 9 (a) Objeto representado por su esqueleto; (b) primitivas; (c) estructuras generada al utilizar una gramática de cadenas regulares. [B]

- *Uso de la semántica.* En situaciones más complicadas, es necesario indicar explícitamente las reglas de conectividad, así como la información relativa a otros factores (como la longitud y dirección de las primitivas), y el número de veces que se puede aplicar una producción. Básicamente, la sintaxis inherente a las reglas de producción define la estructura de un objeto, mientras que la semántica define si es correcto o no. Si utilizamos la información semántica, podremos emplear unas pocas reglas sintácticas para definir una amplia (pero tan limitada como se desee) clase de patrones. Por ejemplo, al especificar la dirección  $\theta$ , ver al siguiente tabla:

Producción	Información semántica
$S \rightarrow aA$	Las conexiones con " $a$ " sólo se realizan en el punto. La dirección de " $a$ ", indicada por $\theta$ , está dada por la dirección de la bisectriz perpendicular a la línea que une los extremos de los dos segmentos sin puntos. Los segmentos son de 3 cm cada uno.
$A \rightarrow bA$	Las conexiones con " $b$ " sólo se realizan en los puntos. No se permiten múltiples conexiones. La dirección " $b$ " debe ser la misma que la de " $a$ ". la longitud de $b$ es de 0.25 cm. Esta producción no se puede aplicar más de 10 veces.
$A \rightarrow bB$	Las direcciones " $a$ " y " $b$ " deben ser iguales. las conexiones deben ser sencillas y sólo se pueden realizar en los puntos.
$B \rightarrow c$	La dirección de " $a$ " debe ser igual a la de " $c$ ". Las conexiones deben ser sencillas y sólo se puede realizar en los puntos.

## 5.5 AUTOMATAS PARA RECONOCIMIENTO

∅ *Autómatas como reconocedor de cadenas.* Los conceptos básicos que subyacen en el reconocimiento sintáctico se pueden ilustrar desarrollando modelos matemáticos de máquinas de cálculo, denominadas *autómatas*. Dada una cadena correspondiente a un patrón de entrada, un autómata es capaz de reconocer si dicho patrón pertenece al lenguaje al que está asociado el autómata. Los *autómatas finitos* son los reconocedores de lenguajes generados por la gramáticas regulares. Se define un autómata finito como la 5-tupla:

$$A_f = (Q, \Sigma, \delta, q_0, F)$$

[Ecu. 5. 22]

donde:

- $Q$  es un conjunto finito de *estados*, no-vacío,  $\Sigma$  es un *alfabeto* de entrada finito.
- $\delta$  es una *correspondencia* de  $Q \times \Sigma$  ( el conjunto de pares ordenados compuestos por elementos de  $Q$  y de  $\Sigma$ ) con la colección de todos los subconjuntos de  $Q$ .
- $q_0$  es el estado inicial y
- $F$  (un subconjunto de  $Q$ ) es un conjunto de *estados finales*.

Ejemplo: Considérese un autómata dado por:

$$A_f = (Q, \Sigma, \delta, q_0, F)$$

[Ecu. 5. 23]

Con  $Q = \{q_0, q_1, q_2\}$ ,  $\Sigma = \{a, b\}$ ,  $F = \{q_0\}$ , y las correspondencias dadas por:

$$\delta(q_0, a) = \{q_2\},$$

$$\delta(q_0, b) = \{q_1\},$$

$$\delta(q_1, a) = \{q_2\},$$

$$\delta(q_1, b) = \{q_0\},$$

$$\delta(q_2, a) = \{q_0\}$$

[Ecu. 5. 24]

Si, por ejemplo, el autómata está en el estado “ $q_0$ ”, y recibe una entrada “ $a$ ”, su estado cambia a  $q_2$ . De forma análoga, si “ $b$ ” es la siguiente entrada, el autómata cambia al estado “ $q_1$ ” y así sucesivamente. En este caso, los estados inicial y final son el mismo.

El *diagrama de estados* del autómata consta de un nodo para cada estado y de arcos dirigidos que muestran las posibles transiciones entre estados. el estado final aparece con una doble circunferencia, y cada arco tiene una etiqueta que indica el símbolo que provoca la transición entre los estados unidos por dicho arco, se dice que una cadena  $w$  de símbolos terminales es *aceptada* o *reconocida* por un autómata,  $s$ , comenzando en el estado  $q_0$ , dicha secuencia de símbolos (leída de izquierda a derecha en  $w$ ) hace que el autómata pase al estado final tras la lectura del último símbolo de  $w$ . [1]

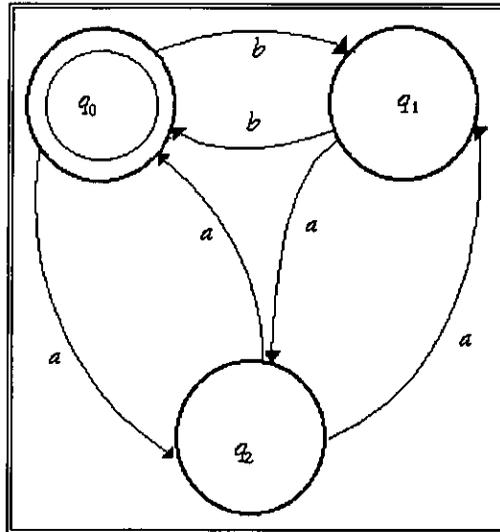


Figura 5. 10 Un autómata finito, este autómata reconoce la cadena  $w=abbabb$ , pero rechaza la cadena  $w=aabab$ . [B]

Existe una correspondencia biunívoca entre las gramáticas regulares y los autómatas finitos. Es decir, un autómata finito reconocerá un lenguaje sí, y sólo sí, está generado por una gramática regular. Supongamos que se tiene una gramática  $G=(N,\Sigma,P,X_0)$ , donde  $X_0 \in S$ , supongamos también que  $N$  se compone de  $X_0$  y de  $n$  no terminales adicionales,  $X_1, X_2, \dots, X_n$ . El conjunto de  $Q$  del autómata se constituye introduciendo  $n+2$  estados  $\{q_0, q_1, \dots, q_n, q_{n+1}\}$ , de modo que  $q_i$  se corresponde con  $X_i$  para  $0 \leq i \leq n$ , y que  $q_{n+1}$  es el estado final. El conjunto de símbolos de entrada es idéntico al conjunto de terminales de  $G$ . Las correspondencias de  $\delta$  se obtienen utilizando dos reglas basadas en las producciones de  $G$ ; a saber, para cada  $i$  y  $j$ ,  $0 \leq i \leq n$ ,  $0 \leq j \leq n$ :

1. Si  $X_i \rightarrow aX_j$  está en  $P$ , entonces  $\delta(q_i, a)$  contiene el estado  $q_j$ .
2. Si  $X_i$  está en  $P$ , entonces  $\delta(q_i, a)$  contiene el estado  $q_{n+1}$ .

A la inversa, dado un autómata finito,  $A_f=(Q,\Sigma,\delta,q_0,F)$ , podemos obtener la correspondiente gramática regular,  $G=(N,\Sigma,P,X_0)$  haciendo que  $N$  contenga los elementos de  $Q$ , suponiendo que el símbolo inicial  $X_0$  se corresponde con  $q_0$ , y donde las producciones de  $G$  se obtienen de la siguiente forma:

1. Si  $q_j$  está en  $\delta(q_i, a)$  existe una producción  $X_i \rightarrow aX_j$  en  $P$ .
2. Si un estado de  $F$  está en  $\delta(q_i, a)$ , existe una producción  $X_i \rightarrow a$  en  $P$ .

El conjunto de terminales,  $\Sigma$ , es el mismo en ambos casos.

### 5.6 RECONOCIMIENTO SINTACTICO DE ARBOLES

Suponiendo que las regiones de la imágenes o los objetos de interés se han expresado en forma de árboles utilizando los elementos primitivos apropiados.

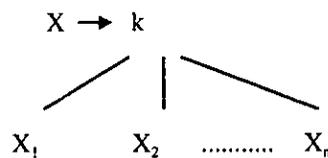
- Gramática de árboles. Se define la gramática de árboles como la 5-tupla:

$$G = (N, \Sigma, P, r, S)$$

[Ecu. 5. 25]

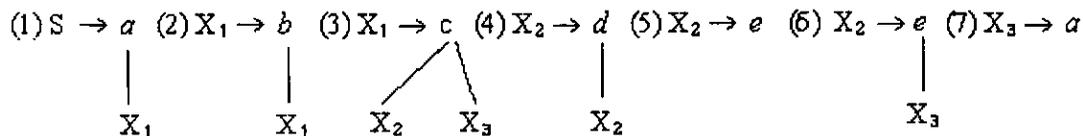
donde:

- $N$  y  $\Sigma$  son conjuntos de no terminales y de terminales, respectivamente
- $S$  contenido en  $N$ , es el símbolo inicial, que en general puede ser un árbol;  $P$  es un conjunto de producciones de la forma  $T_i \rightarrow T_j$ , donde  $T_i$  y  $T_j$  son árboles; y
- $r$  es una *función de categoría*, que indica el número de descendientes directos de un nodo cuya leyenda sea un terminal de la gramática. En este estudio son de especial relevancia las gramáticas expansivas, que tienen producciones de la forma:



donde  $X_1, X_2, \dots, X_n$  son no terminales y  $k$  es un terminal.

Ejemplo: el esqueleto de la figura 5.11 se puede generar utilizando una gramática de árbol con  $N=\{X_1, X_2, X_3, S\}$ , y  $\Sigma=\{a, b, c, d, e\}$ , donde los terminales representan las primitivas. Suponiendo una conectividad de arriba abajo de las primitivas correspondientes a las líneas, y conexiones arbitrarias a los círculos en cualquier punto de la circunferencia, la gramática considerada posee producciones de la forma:



En este caso, las funciones de categoría son  $r(a)=\{0,1\}$ ,  $r(b)=r(d)=r(e)=\{1\}$  y  $r(c)=\{2\}$ . Restringiéndose las aplicaciones 2,4 y 6 a un mismo número de veces, se generaría una estructura en la que las tres ramas tendrían la misma longitud análogamente, si se impone la restricción de que las producciones 4 y 6 se deben aplicar el mismo número de veces, se obtendría una estructura simétrica con respecto al eje vertical.

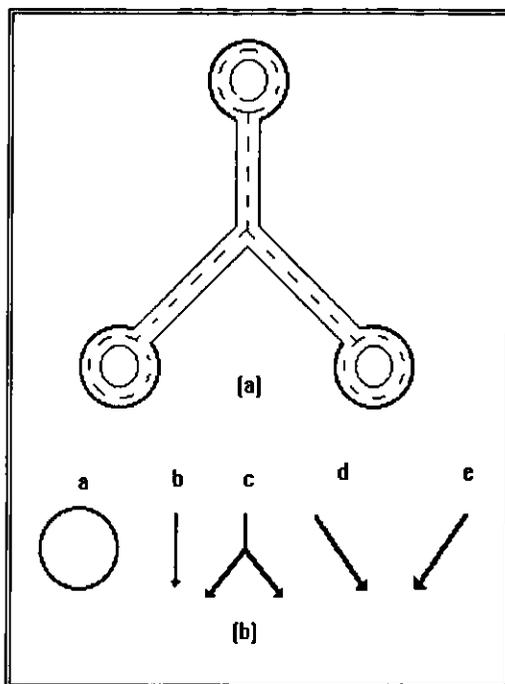


Figura 5. 11(a) Un objeto y (b) las primitivas utilizadas para representar el esqueleto en términos de una gramática de árbol. [B]

#### Ø *Autómatas como reconocedor de cadenas.*

Mientras que un autómata finito convencional debe explorar su cadena de entrada símbolo a símbolo, y de izquierda a derecha, un autómata de árbol debe empezar simultáneamente en cada nodo de la frontera (las hojas, consideradas de izquierda a derecha) de un árbol de entrada, y continuar por caminos paralelos hacia la raíz. Específicamente, se define un autómata de frontera a raíz como:

$$A_f = (Q, F \{f_k | k \in \Sigma\})$$

[Ecu. 5. 26]

donde:

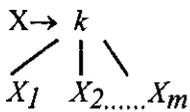
$Q$  es un conjunto finito de estados,

$F$ , subconjunto de  $Q$ , es un conjunto de estados finales, y

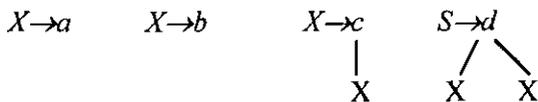
$f_k$  es una relación  $Q^m \times Q$  tal que  $m$  es una categoría de  $k$ .

la notación  $Q^m$  representa el producto cartesiano de  $Q$  aplicado  $m$  veces:  $Q^m = Q \times Q \times Q \times \dots \times Q$ . Considerando la definición de producto cartesiano, sabemos que esta expresión hace referencia al conjunto ordenado de todas las  $m$ -tuplas con elementos de  $Q$ . Por ejemplo, si  $m=3$ ,  $Q^3 = Q \times Q \times Q = \{x, y, z | x \in Q, y \in Q, z \in Q\}$ . Se recuerda que una relación  $R$  entre un conjunto de  $A$  y un conjunto  $B$  es un subconjunto del producto cartesiano de  $A$  y  $B$ ; es decir,  $R \subseteq A \times B$ . Por lo tanto, una relación en  $Q^m \times Q$  es simplemente un subconjunto del conjunto  $Q^m \times Q$ .

Para una gramática del árbol expansiva,  $G=(N, \Sigma, P, r, S)$ , el autómata de árbol correspondiente se construye haciendo  $Q=N$ , con  $F= \{S\}$  y definiendo para cada símbolo  $a$  de  $\Sigma$  una relación  $f_k$  tal que  $(X_1, X_2, \dots, X_m, X)$  pertenezca a  $f_k$  si y sólo si en  $G$  existe una producción

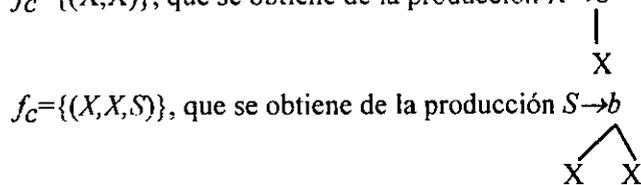


Por ejemplo, si consideramos una sencilla gramática de árbol  $G=(N, \Sigma, P, r, S)$ , con  $N=\{S, X\}$ ,  $\Sigma=\{a, b, c, d\}$ , las producciones



y las categorías  $r(a)=\{0\}, r(b)=\{0\}, r(c)=\{1\}$  y  $r(d)=\{2\}$ . El correspondiente autómata de árbol,  $A_f=(Q, F, \{f_k \mid k \in \Sigma\})$  queda especificado haciendo  $Q= \{S, X\}$ ,  $F= \{S\}$  y  $\{f_k \mid k \in \Sigma\}=\{f_a, f_b, f_c, f_d\}$ , donde las relaciones se definen de la siguiente forma:

$f_a=\{(\emptyset, X)\}$ , que se obtiene de la producción  $X \rightarrow a$   
 $f_b=\{(\emptyset, X)\}$ , que se obtiene de la producción  $X \rightarrow b$   
 $f_c=\{(X, X)\}$ , que se obtiene de la producción  $X \rightarrow c$



La interpretación de la relación  $f_a$  es que a un nodo de leyenda  $a$ , sin descendiente (de ahí el símbolo nulo  $\emptyset$ ) se le asigna el estado  $X$ . La interpretación de  $f_c$  es que a un nodo de leyenda  $c$ , que tenga un solo descendiente de estado  $X$ , se le asigna el estado  $X$ . La interpretación de  $f_d$  es un nodo de leyenda  $d$ , con dos descendientes, ambos de estado  $X$ , se le asigna el estado  $S$ .

Para ver cómo puede este autómata de árbol reconocer un árbol generado por esta gramática, teniendo un árbol de entrada, en primer lugar, el autómata de árbol  $A_f$  asigna los estados de los nodos frontera  $a$  y  $b$  mediante las relaciones  $f_a$  y  $f_b$ , respectivamente. En este caso, de acuerdo con estas dos relaciones, se asigna el estado  $X$  a ambas hojas. A continuación, el autómata sube un nivel desde la frontera y realiza una asignación de estado al nodo  $c$ , teniendo en cuenta  $f_c$  y el estado del descendiente de este nodo. De nuevo, la asignación de estado obtenida con  $f_c$  es  $X$ . Al ascender un nivel más, el autómata encuentra el nodo  $d$  y como se han asignado ya los estados de sus descendientes, se utiliza la relación  $f_b$ , que asigna el estado  $S$  al nodo  $d$  puesto que se trata del último nodo, y el estado  $S$  está en  $F$ , el autómata acepta (reconoce) el árbol como miembro válido del lenguaje de la gramática de árbol definida anteriormente. La figura 5.12 muestra la representación final de las secuencias de estados establecidas en los caminos que van desde la frontera hasta la raíz.

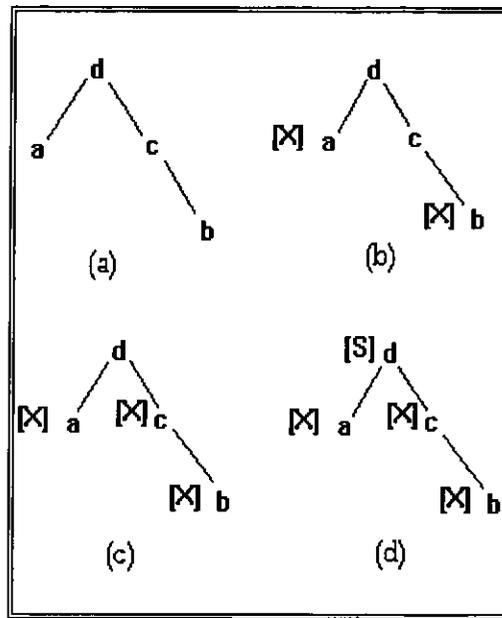


Figura 5. 12. Procesamiento de estados de un autómata de árbol del tipo frontera a raíz: (a) Árbol de entrada; (b) asignación de estados a los nodos fronteras; (c) asignación de estados a los nodos intermedios; (d) asignación de estados al nodo raíz. [B]

## 5.7 APRENDIZAJE

La aplicación del reconocimiento de patrones tiene dos enfoques principales. En el más tradicional, el investigador determina por medio de un análisis estadístico aquellas características que permiten diferenciar un patrón de otro, así como el nivel de confiabilidad. En contraste, el enfoque moderno se basa en otros principios y no requiere tanto de los procesos de abstracción y discernimiento del investigador. Aquí se trata de emular la manera como llegamos a distinguir objetos y formar categorías; esto es, mediante un proceso de aprendizaje.

En sistemas artificiales, “aprender” consiste en generar una sucesión de representaciones, este último comienza con una muestra burda o concepto inicial, que se transformara hasta lograr una representación adecuada del concepto o patrón. Para ello, se favorecen las transformaciones que nos acercan al ideal y se inhiben aquellas que nos alejan, utilizando un conjunto de objetos típicos, llamado conjunto de entrenamiento. La aplicación del reconocimiento de patrones empleando aprendizaje resulta apropiado en casos donde es difícil encontrar una parametrización adecuada de los patrones y en donde hay muchas variaciones y excepciones, como en la identificación de instancias de rostros. Algunos beneficios de esta aplicación -aprendizaje supervisado- facilitan la determinación automática de fronteras de decisión entre clases de objetos, permitiendo llegar a un concepto más general e identificar objetos que se asemejan a los que participaron en el entrenamiento, pero que no necesariamente son iguales; y que la eficacia de un sistema puede aumentar progresivamente al considerar nuevos casos y encontrar ejemplos. Esto permite refinar el concepto que se tiene de los patrones. Otro tipo de aprendizaje es el no supervisado, que busca encontrar regularidades y tendencias en una población de datos.

En la medida que esto pudiera identificarse con ciertas capacidades intelectuales de abstracción y generalización, podría hablarse de cierta inteligencia o creatividad artificial para descubrir nuevos conceptos. El área de redes neuronales artificiales es representativa de este enfoque, las redes neuronales

son otra forma de emular características propias de los humanos: la capacidad de memorizar y asociar hechos. Las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto de un sistema del cual disponemos capaz de adquirir conocimiento a través de la experiencia. Una red neuronal es un nuevo sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano, la neurona. Si examinamos con atención aquellos problemas que no pueden expresarse a través de un algoritmo nos daremos cuenta que todos ellos tienen una característica común: la experiencia. El aprendizaje de una red se puede producir de tres formas:

- a) Aprendizaje supervisado: consiste en introducir una serie de patrones de entrada a la red y a su vez mostrar la salida que se quiere tener. La red es capaz de ajustar los pesos de las neuronas de tal forma que a la presentación posterior de esos patrones de entrada a la red responde con salida memorizada.
- b) Aprendizaje no supervisado: se presentan los patrones de entrada a la red y ésta los clasifica en categorías según sus rasgos más sobresalientes.
- c) Aprendizaje autosupervisado: la propia red corrige los errores en la interpretación empleando una realimentación.

Estas redes emplean las arquitecturas de la Teoría de la Resonancia Adaptiva (ART). Entre otras aplicaciones se encuentran el reconocimiento de caracteres manuscritos, autorización de descubiertos bancarios y clasificación de cromosomas.

Las técnicas de reconocimiento sintáctico analizadas requieren la especificación de los autómatas (reconocedores) apropiados para cada clase en consideración. En situaciones sencillas, con la inspección se puede generar el autómata necesario. En situaciones más complejas, puede ser necesario un algoritmo que defina (aprenda) el autómata partiendo de patrones muestra (como, por ejemplo, cadenas o árboles). Debido a la correspondencia biunívoca existente entre autómatas y gramáticas, a veces se plantea el problema del aprendizaje en términos de aprendizaje directo de gramáticas a partir de patrones de muestra, un proceso al que normalmente se le denomina *inferencia gramática*. [1]

Supongamos que una gramática desconocida,  $G$ , genera todos los patrones de una clase, y que disponemos de un conjunto finito de muestras  $R^+$  que posee la siguiente propiedad:

$$R^+ \subseteq \{\alpha \mid \alpha \text{ en } L(G)\}$$

[Ecu. 5. 27]

Denominado *conjunto positivo de muestras*, es simplemente un conjunto de patrones de entrenamiento de la clase asociada a la gramática  $G$ . Se dice que el conjunto de muestras  $R^+$  es *estructuralmente completo* si se utiliza cada producción de  $G$  para generar al menos un elemento de  $R^+$ . Se desea aprender (sintetizar) un autómata finito  $A_f$  que acepte las cadenas de  $R^+$ , y de ser posible, algunas cadenas que se parezcan a las de  $R^+$ .

Sobre la base de la definición de autómata finito y según la correspondencia existente entre  $G$  y  $A_f$ , se deduce que  $R^+ \subseteq \Sigma^*$ , siendo  $\Sigma^*$  el conjunto de todas las cadenas compuestas de elementos de  $\Sigma$ .

Supongamos que “ $z$ ”, incluida en  $\Sigma^*$ , es una cadena tal que, para algunos “ $w$ ” de  $\Sigma^*$ , “ $zw$ ” pertenece a  $R^+$ . Si  $k$  es un entero positivo, definimos la cola  $k$  de “ $z$ ” con respecto a  $R^+$ .

$$h(z, R^+, k) = \{w | zw \text{ en } R^+, |w| \leq k\}$$

[Ecu. 5. 28]

En otras palabras, la cola  $k$  de “ $z$ ” es el conjunto de cadenas  $w$  que tienen las propiedades de que

- 1) “ $zw$ ” pertenece a  $R^+$
- 2) la longitud de  $w$  es menor o igual que  $k$ .

Un procedimiento para aprender un autómata  $A_f (R^+, k) = (Q, \Sigma, \delta, q_0, F)$  a partir de un conjunto de muestras  $R^+$  y para un determinado valor de  $k$  consiste en hacer

$$h(z, R^+, k) = \{w | zw \text{ en } R^+, |w| \leq k\}$$

[Ecu. 5. 29]

y, para cada “ $a$ ” perteneciente a  $\Sigma$ ,

$$\delta(q, a) = \{q' \text{ en } Q | q' = h(za, R^+, k), \text{ con } q = h(z, R^+, k)\}$$

[Ecu. 5. 30]

Además, hacemos que

$$q_0 = h(\lambda, R^+, k)$$

[Ecu. 5. 31]

y

$$F = \{q | q \text{ en } Q, \lambda \text{ en } q\}$$

[Ecu. 5. 32]

donde  $\lambda$  es la cadena vacía (una cadena sin símbolos). Podemos observar que  $A_f(R^+, k)$  posee como estados los subconjuntos del conjunto de todas las  $k$  que se pueden construir a partir de  $R^+$ .

Ejemplo: Supongamos que  $R^+ = \{a, ab, abb\}$  y  $k=1$ . de la definición anterior se obtiene:

$$\begin{aligned} z = \lambda, \quad h(\lambda, R^+, 1) &= \{w|\lambda w \text{ en } R^+, |w| \leq 1\} \\ &= \{a\} \\ &= q_0; \end{aligned}$$

$$\begin{aligned} z = a, \quad h(a, R^+, 1) &= \{w|aw \text{ en } R^+, |w| \leq 1\} \\ &= \{\lambda, b\} \\ &= q_1; \end{aligned}$$

$$\begin{aligned} z = a, \quad h(a, R^+, 1) &= \{\lambda, b\} \\ &= q_1; \end{aligned}$$

$$\begin{aligned} z = a, \quad h(a, R^+, 1) &= \{\lambda\} \\ &= q_2; \end{aligned}$$

[Ecu. 5. 33]

En este caso, de otras cadenas  $z$  de  $\Sigma^*$  se obtienen cadenas "zw" que no pertenecen a  $R^+$ , haciéndose necesario un cuarto estado, representando por  $q_\emptyset$ , y que corresponde a la circunstancia de que  $h$  sea el conjunto nulo. Los estados son, por tanto,  $q_0 = \{a\}$ ,  $q_1 = \{\lambda, b\}$ ,  $q_2 = \{\lambda\}$  y  $q_\emptyset$ , y forman el conjunto  $Q = \{q_0, q_1, q_2, q_\emptyset\}$ . Aunque los estados se obtienen como conjuntos de símbolos (colas  $k$ ), sólo se utilizan las leyendas de estado  $q_0, q_1, \dots$  para construir el conjunto  $Q$ .

El siguiente paso consiste en obtener las funciones de transición. Puesto que,

$$q_0 = h(\lambda, R^+, 1),$$

[Ecu. 5. 34]

se deduce que,

$$\delta(q_0, a) = h(\lambda a, R^+, 1) = h(a, R^+, 1) = q_1$$

[Ecu. 5. 35]

y

$$\delta(q_0, b) = h(\lambda b, R^+, 1) = h(b, R^+, 1) = q_\emptyset.$$

[Ecu. 5. 36]

De forma análoga,

$$q_1 = h(a, R^+, 1) = h(ab, R^+, 1)$$

[Ecu. 5. 37]

y

$$\delta(q_1, a) = h(aa, R^+, 1) = h(aba, R^+, 1) = q_\emptyset$$

[Ecu. 5.38]

Además,

$$\delta(q_1, b) \supseteq h(ab, R^+, 1) = q_1$$

[Ecu. 5.39]

y

$$\delta(q_1, b) \supseteq h(abb, R^+, 1) = q_2;$$

[Ecu. 5.40]

esto es,

$$\delta(q_1, b) = \{q_1, q_2\}.$$

[Ecu. 5.41]

Si se sigue el procedimiento descrito, se obtiene

$$\delta(q_2, a) = \delta(q_2, b) = \delta(q_\emptyset, a) = \delta(q_\emptyset, b) = q_\emptyset$$

[Ecu. 5.42]

El conjunto de estados finales contiene los estados que tienen la cadena vacía  $\lambda$  en su representación de cola  $k$ . En este caso,  $q_1 = \{\lambda, b\}$  y  $q_2 = \{\lambda\}$ , por lo que  $F = \{q_1, q_2\}$ .

Con los resultados, el autómata inferido viene dado por  $A_f(R^+, 1) = (Q, \Sigma, \delta, q_0, F)$ , donde  $Q = \{q_0, q_1, q_2, q_\emptyset\}$ ,  $\Sigma = \{a, b\}$ ,  $F = \{q_1, q_2\}$ , y las funciones de transición son las antes indicadas.

El autómata acepta cadenas de la forma  $a, ab, abb, \dots, ab^n$ . En otras palabras, el procedimiento ha logrado identificar la regularidad iterativa del símbolo  $b$ .

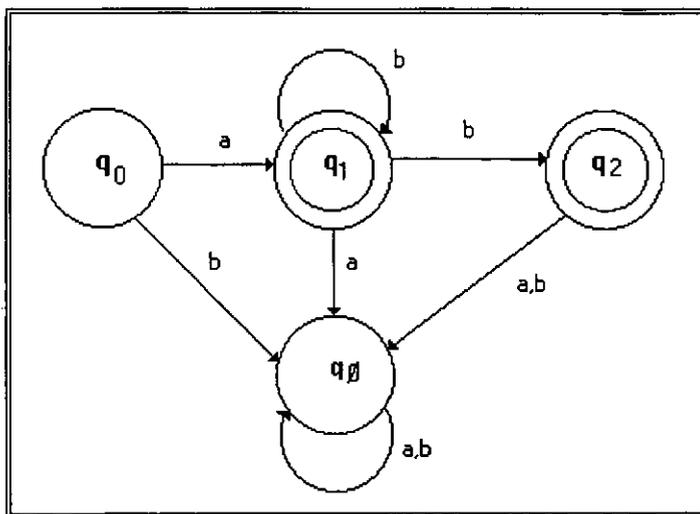


Figura 5.13 Diagrama de estado del autómata finito  $A_f(R^+, 1)$  inferido del conjunto de muestras  $R^+ = \{a, ab, abb\}$ . [B]

El ejemplo anterior muestra que el valor de  $k$  controla la naturaleza del autómata resultante. Las siguientes propiedades ponen de manifiesto la dependencia de  $A_f(R^+, k)$  respecto a este parámetro.

- *Propiedad 1.*  $R^+ \subseteq L[A_f(R^+, k)]$  para todo  $k \geq 0$ , donde  $L[A_f(R^+, k)]$  es el lenguaje aceptado por  $A_f(R^+, k)$ .
- *Propiedad 2.*  $L[A_f(R^+, k)] = R^+$  si  $k$  es mayor o igual que la longitud de cadena más larga de  $R^+$ ;  $L[A_f(R^+, k)] = \Sigma^*$  si  $k=0$ .
- *Propiedad 3.*  $L[A_f(R^+, k+1)] \subseteq L[A_f(R^+, k)]$ .

La propiedad 1 garantiza que  $A_f(R^+, k)$  aceptará, como mínimo, las cadenas del conjunto de muestras  $R^+$ . entonces, por la propiedad 2, el autómata sólo aceptará las cadenas de  $R^+$ . Si  $k=0$ ,  $A_f(R^+, 0)$  constará de un solo estado,  $q_0 = \{\lambda\}$ , que actuará como estado inicial y final. Las funciones serán entonces de la forma  $\delta(q_0, a) = q_0$ , para  $a$  perteneciente a  $\Sigma$ . Por tanto,  $L[A_f(R^+, 0)] = \Sigma^*$ , y el autómata aceptará la cadena vacía  $\lambda$  y todas las cadenas compuestas de símbolos de  $\Sigma$ . Por último, la propiedad 3 indica que el ámbito del lenguaje aceptado por  $A_f(R^+, k)$  se reduce conforme se incrementa el valor de  $k$ . [1]

Estas tres propiedades permiten controlar la naturaleza de  $A_f(R^+, k)$  variando simplemente el parámetro  $k$ . Si  $L[A_f(R^+, 0)]$  es una conjetura del lenguaje  $L_0$  a partir del que se han elegido la muestra  $R^+$  y si  $k$  es muy pequeño, esta conjetura de  $L_0$  constituirá una inferencia liberal que puede incluir la mayor parte o la totalidad de las cadenas de  $R^+$ . Sin embargo, si  $k$  es igual a la longitud de las cadenas más larga de  $R^+$ , la inferencia será conservadora en el sentido de que  $A_f(R^+, k)$  sólo aceptará las cadenas incluidas en  $R^+$ .

Ejemplo: consideré el conjunto  $R^+ = \{caaab, bbaab, caab, bbab, cab, bbb, cb\}$ . Para  $k=1$ , si se sigue el mismo procedimiento del ejemplo anterior se obtienen:

1.  $z = \lambda$ ,  $h(\lambda, R^+, 1) = \{\emptyset\} = q_0$ ;
2.  $z = c$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
3.  $z = ca$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
4.  $z = cb$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
5.  $z = caa$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
6.  $z = cab$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
7.  $z = caaa$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
8.  $z = caab$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
9.  $z = caaab$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
10.  $z = b$ ,  $h(z, R^+, 1) = \{\emptyset\} = q_0$ ;
11.  $z = bb$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
12.  $z = bba$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
13.  $z = bbb$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
14.  $z = bbaa$ ,  $h(z, R^+, 1) = \{b\} = q_1$ ;
15.  $z = bbab$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;
16.  $z = bbaab$ ,  $h(z, R^+, 1) = \{\lambda\} = q_0$ ;

[Ecu. 5. 43]

El autómata es  $A_f(R^+,1)=(Q,\Sigma,\delta,q_0,F)$ , con  $Q=\{q_0q_1q_2\}$ ,  $\Sigma=\{a,b,c\}$ ,  $F=\{q_0\}$ , y las transiciones que aparecen en el diagrama de estado. Para que una cadena sea aceptada por el autómata, debe comenzar por  $a,b$ , o  $c$ , y finalizar con  $b$ . Además, las cadenas con recursividad en  $a,b$ , o  $c$  son aceptadas por  $A_f(R^+,1)$ .

La principal ventaja del método anterior es la simplicidad de implementación. El procedimiento de síntesis se puede simular en una computadora con un pequeño esfuerzo. El principal inconveniente es la decisión del valor apropiado para  $k$ , aunque las tres propiedades anteriores simplifican hasta cierto punto este problema.

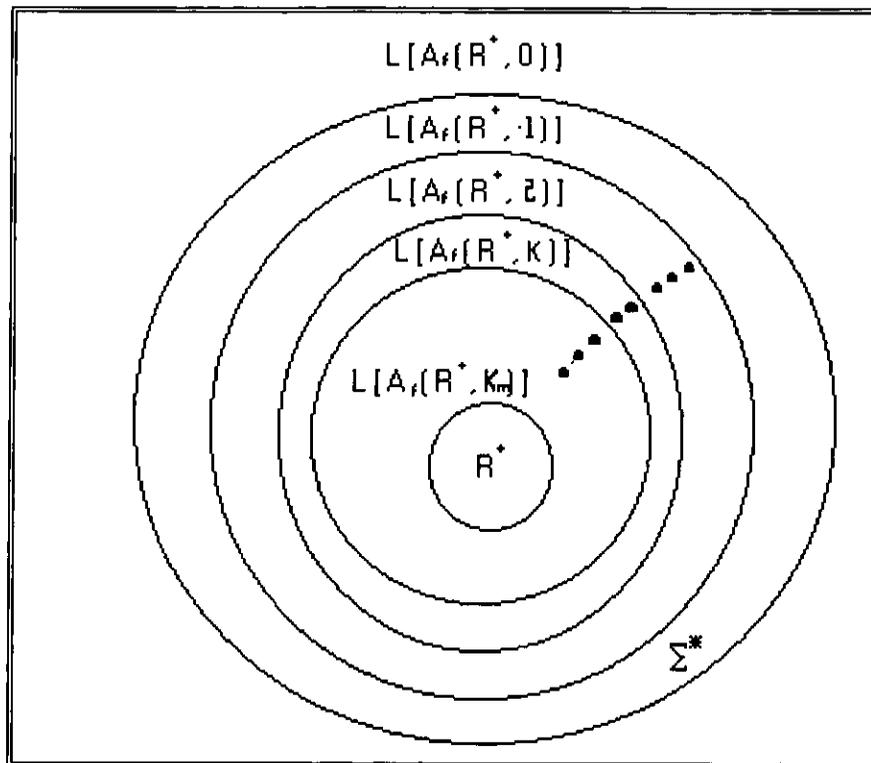


Figura 5. 14 Relación entre  $L[A_f(R^+,k)]$  y  $k$ . El valor de  $k_m$  es tal que  $k_m \geq$  (la longitud de la cadena mas larga de  $R^+$ ). [B]

## II COMENTARIO AL CAPITULO II

La identificación de patrones en imágenes ha tenido una gama de enfoques muy amplia, el tratar de imitar un proceso inteligente que tiene como objetivo la identificación visual y el poder de discriminación entre objetos es un proceso sumamente ambicioso. Estos procesos están comprendidos en lo que se conoce como inteligencia artificial. En este trabajo de tesis no se abocó a generar un estudio amplio y comparativo entre todos los métodos, por la complejidad y especialización que tiene cada uno. Y esto va más allá del objetivo de este trabajo.

Una de las áreas que actualmente está sobresaliendo en el reconocimiento de patrones es el análisis estructural por medio de métodos sintácticos, este enfoque tiene como base la teoría informática llamada teoría de lenguajes, utilizada en el diseño de software para compiladores. Estas teorías fueron llevadas al ámbito de reconocimiento de patrones por su efectividad para ser programadas en una computadora y su adecuado algoritmo para identificar cadenas de caracteres, cadenas que en una imagen representan

*primitivas* que estructuran un objeto en la imagen. El proceso de reconocimiento sintáctico puede definirse como la identificación correcta de una secuencia de caracteres que estructuran un objeto contenido en la escena de una imagen digital. Este proceso fue elegido por la naturaleza y primicias de este trabajo de tesis, donde se ha mantenido la teoría de poder representar y finalmente reconocer patrones en imágenes complejas, que a través de un tratamiento digital que finalmente se han representado por una serie de objetos simples como son líneas, círculos y contornos irregulares.

En el siguiente capítulo se implementarán experimentos con la finalidad de mostrar la validez de estas teorías y poder hacer un análisis completo de los métodos sintácticos, comprobar su efectividad y enmarcar sus limitaciones para pasar finalmente a plantear las conclusiones finales de esta tesis.

· C · A · P · I · T · U · L · O ·

VI

IMPLEMENTACION DE  
EXPERIMENTOS Y  
EVALUACION DE  
RESULTADOS

Este último capítulo tiene como objetivo desarrollar una serie de experimentos que ejemplifiquen el funcionamiento de los algoritmos para el reconocimiento de patrones descritos en el capítulo 5 de esta tesis, los experimentos ilustran las funciones básicas que soportan la filosofía del reconocimiento sintáctico de patrones estructurados; para este fin, en los experimentos se hace uso de todos los elementos de preprocesamiento descritos durante el desarrollo de este trabajo de tesis, al integrar las técnicas de preprocesamiento, los experimentos tienen la finalidad de darnos una idea aun más cierta de la importancia que tienen este proceso dentro del esquema para el reconocimiento de patrones, tan importante como lo es la implementación eficiente del algoritmo para el reconocimiento.

Los experimentos parten desde las funciones básicas en los primeros casos hasta su adecuación para ser utilizados en aplicaciones particulares, observando de esta manera su desempeño y eficiencia en diferentes áreas del reconocimiento de patrones, así sus particularidades que pueden presentarse para su implementación y análisis de resultados.

En cada experimento se hace una estructuración detallada de los objetivos y alcances de cada ejemplo, los algoritmos son implementados en Matlab™, con la ventaja que se podrá hacer un seguimiento paso a paso del desarrollo de las diferentes fases comprendidas en el experimento.

## DISEÑO DE EXPERIMENTOS

### 6.1 EXPERIMENTO 1

**Título:** RECONOCIMIENTO SINTACTICO DE CADENAS.

**Objetivo:** Diseñar un experimento que ejemplifique el reconocimiento sintáctico de una estructura en función de una cadena de elementos sencillos.

**Metodología:** Partimos de una estructura sencilla en la que no existe un preprocesamiento para su diseño, construida en función de segmentos de recta, este experimento tiene la finalidad de ilustrar los conceptos de reconocimiento sintáctico de cadenas y por tanto no se hace un énfasis en el preprocesamiento de la imagen. Se seleccionó el formato *bmp* (mapa de bits) para generar los archivos gráficos por la facilidad que ofrece este formato para trazar dibujos sencillos.

**Proceso:**

- Se define la estructura patrón.
- Se definen las primitivas como un vector de direcciones en función de un código de freeman.
- Se generan las reglas que describen la secuencia de la estructura.
- Se agregan algunas reglas para crear una familia de patrones.
- Se diseña el autómata para reconocer la secuencia de cadenas válidas.

#### 6.1.1 Definición de la estructura

La estructura seleccionada para este experimento, es generada por una secuencia de segmentos de línea recta con diferentes direcciones de pendiente, a la cual nombraremos "estrella". La imagen tiene el formato *bmp* (mapa de bits) donde los pixeles que describen la estructura son codificados con un "1" y fuera de los trazos se codifica el fondo de la imagen como una matriz de ceros "0". Ver figura 6.1

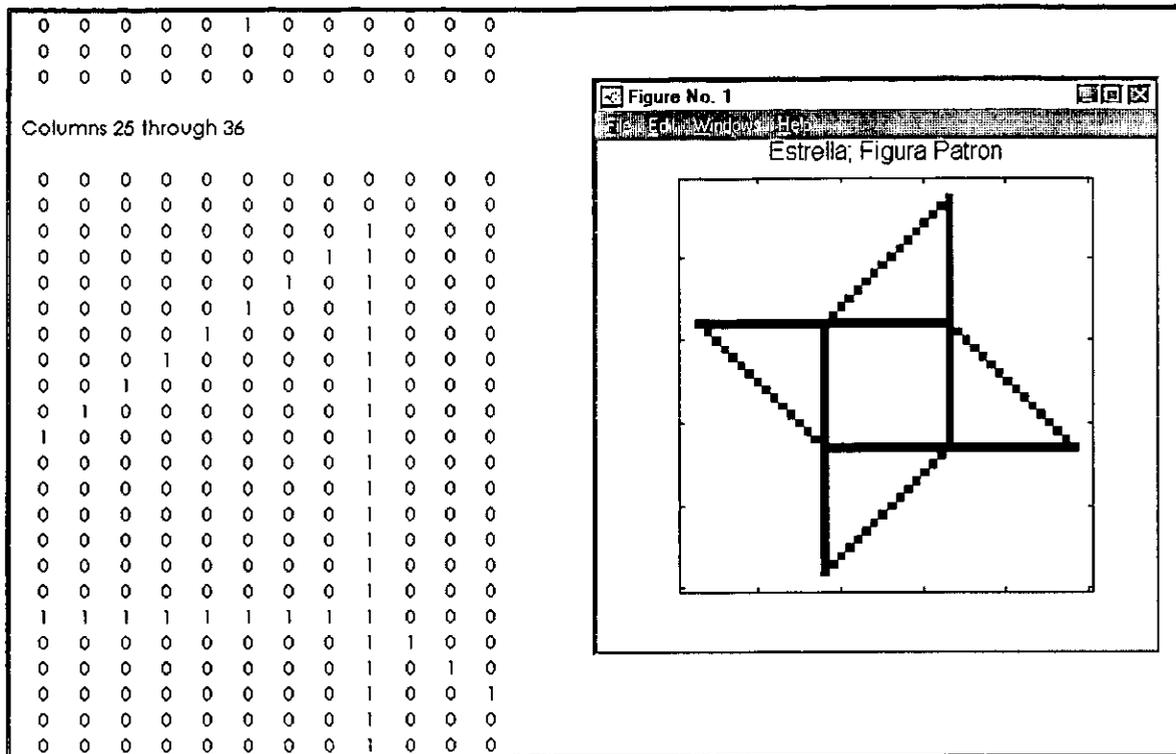


Figura 6. 1 A la derecha se muestra la forma en que se despliega la figura llamada “estrella”; Del lado izquierdo se muestra una sección del despliegue numérico de la matriz que codifica a la “estrella”.[A]

### 6.1.2 Definición del código de freeman, y extracción de primitivas

La definición del código de freeman servirá como sistema de orientación y codificación para las primitivas, para este caso seleccionamos un código de 8 elementos y de esta manera tener una conectividad 8, cubriendo en detalle el radio de acción en el que se puede generara una nueva cadena, se inicia en la dirección uno y se incrementan las direcciones en sentido contrario al de las manecillas del reloj hasta llegar a un valor 8. Ver figura 6.2.

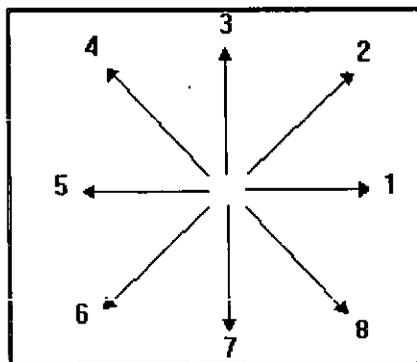


Figura 6. 2 Código de freeman seleccionado como referencia para la extracción de primitivas.. [A]

Una vez definido el código para formar los vectores de dirección, se procede a barrer la imagen (“estrella”) hasta encontrar un valor distinto a cero, una vez encontrado y siguiendo el barrido de la imagen orientado esta vez por la conectividad de unos y de esa manera ir formando el vector de direcciones. El número de

elementos que formen este vector describirá una longitud medida en pixeles, todos los elementos que formen al vector de direcciones tendrán el mismo valor ya que se trata de líneas rectas que siguen una misma orientación. En la Figura 6.3 se observa como se inicia el proceso de extracción y como se inicia la asignación de direcciones para este primer vector.

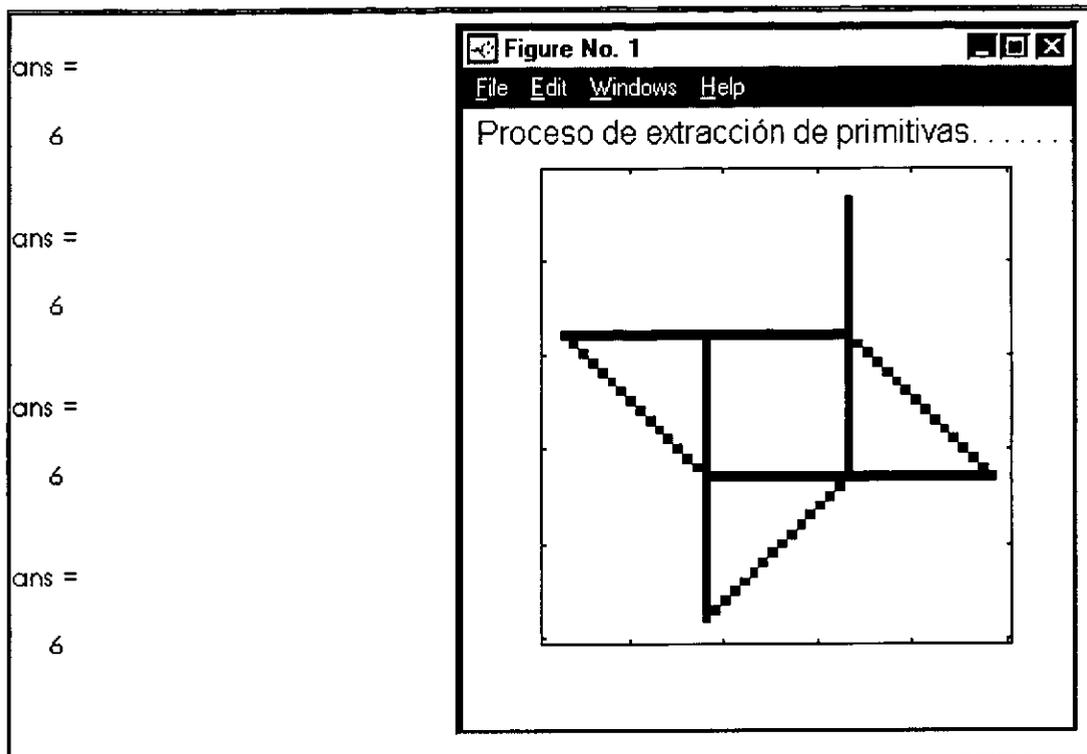


Figura 6. 3 A la derecha se muestra la extracción de un segmento de la figura; A la izquierda la secuencia de direcciones 6 que definen el vector para ese segmento de recta. [A]

Una vez encontrado un vector de direcciones estos puntos (pixeles) tomaran el valor cero para que no sean tomados en cuenta al iniciar una nueva búsqueda.

Una vez obtenido un primer vector de direcciones, el proceso de búsqueda se reinicia hasta encontrar la siguiente secuencia de direcciones, el proceso de eliminación de segmentos es igualmente reiterativo hasta que son codificados todos los elementos de la "estrella". En la Figura 6.4, se muestra el proceso de barrido paso a paso para extraer los elementos fundamentales, en la figura se observa la secuencia de barrido iniciando en el lado superior izquierdo de la imagen y termina en la parte inferior derecha, otra observación que se puede hacer es que al terminar de hacer la caracterización de un elemento de la estructura el proceso de barrido se reinicia en el punto en el que se inicio la caracterización y de esta forma se garantiza haber tomado en cuenta todos los puntos de la imagen.

Al diseñar este experimento se tomo como base encontrar una estructura que abarcara todas las opciones de dirección posible para que la ejemplificación fuera más completa. Al hacer el barrido completo de la estructura algún vector de direcciones podría haberse repetido, siguiendo el criterio para la definición de primitivas hacemos especificaciones en el programa para que al final solo sea considerada una sola vez un vector de direcciones característico, de esta manera no somos redundantes en la definición de elementos básicos de una estructura y de esta manera los vectores resultantes podrán ser nombrados como primitivas.

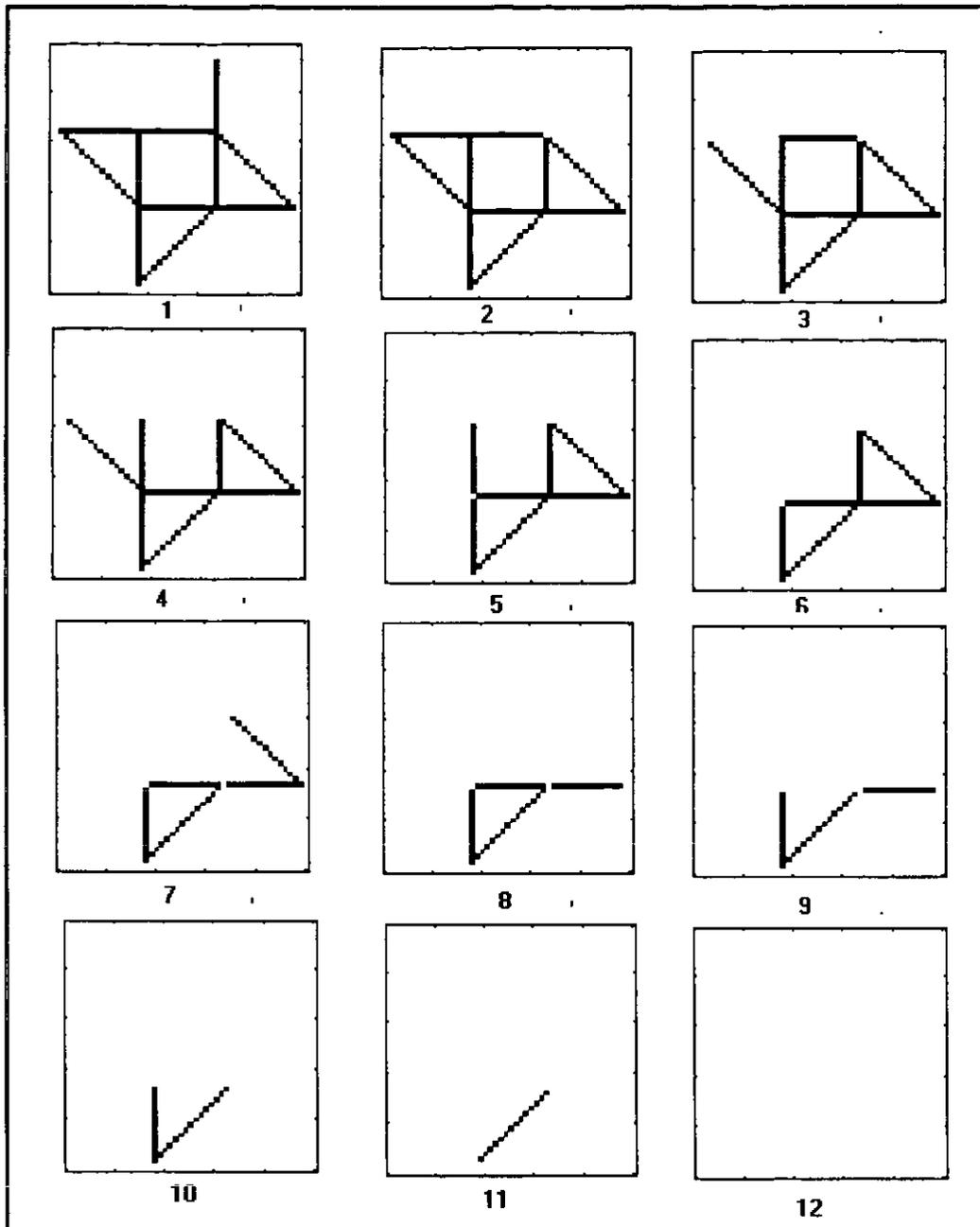


Figura 6. 4 Secuencia de barrido y conformación de segmentos de recta como vectores de direcciones. [A]

Una vez obtenidos todos los vectores de direcciones procedemos a definir las primitivas en función de dichos vectores, las primitivas se conformaran gráficamente como un vector con magnitud y dirección, los vectores contienen 16 elementos de la misma dirección; Para efecto de las siguientes tareas en este experimento se hacen las siguiente adecuaciones, el primer elemento del vector es asignado como (0) para indicar el inicio del vector y el último elemento es nombrado con (-1) y los demás elementos indicaran la dirección característica de ese vector. En la Figura 6.5 se muestra la representación gráfica de las primitivas como una serie de vectores.

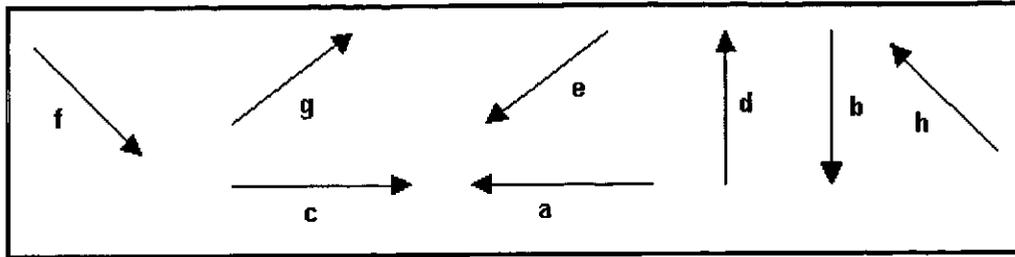


Figura 6. 5 Representación gráfica de las primitivas. [A]

Las primitivas se representaran mediante letras minúsculas del alfabeto, de la letra *a* a la letra *h*. A continuación se muestran el resultado de la obtención de las primitivas para este experimento:

los vectores de direcciones, de las primitivas son:

a =

Columns 1 through 12

0 5 5 5 5 5 5 5 5 5 5 5

Columns 13 through 16

5 5 5 -1

b =

Columns 1 through 12

0 7 7 7 7 7 7 7 7 7 7 7

Columns 13 through 16

7 7 7 -1

c =

Columns 1 through 12

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

Columns 13 through 16

1 1 1 -1

d =

Columns 1 through 12

0 3 3 3 3 3 3 3 3 3 3 3

Columns 13 through 16

3 3 3 -1

e =

Columns 1 through 12

0 6 6 6 6 6 6 6 6 6 6 6

Columns 13 through 16

6 6 6 -1

f =

Columns 1 through 12

0 8 8 8 8 8 8 8 8 8 8 8

Columns 13 through 16

8 8 8 -1

g =

Columns 1 through 12

0 2 2 2 2 2 2 2 2 2 2 2

Columns 13 through 16

2 2 2 -1

h =

Columns 1 through 12

0 4 4 4 4 4 4 4 4 4 4 4

Columns 13 through 16

4 4 4 -1

### 6.1.3 Generación de las reglas que describen la secuencia de la estructura.

Para generar las reglas que nos garanticen la identificación y reconstrucción de la estructura a partir de sus primitivas, seguimos el procedimiento descrito en la sección 5.4 para generar la gramática de lenguaje  $L(G)$  correspondiente. [1]

$$G = (N, \Sigma, P, S)$$

[Ecu. 6. 1]

donde:

- $N$ =Conjunto finito de variables denominadas *no terminales* y las nombraremos con las letras mayúsculas, A,B,C,D,S.
- $\Sigma$ = Conjunto finito de constantes denominadas *terminales* y las nombraremos con las letras minúsculas,  $a,b,c,d,e,f,g,h$ .
- $P$ = Conjunto de reglas de reescritura denominadas producciones.
- $S$ =Es el símbolo inicial a partir del cual se generan las cadenas de  $L(G)$ .

	<i>Producción</i>	<i>Información semántica</i>
1.	$S \rightarrow aA$	Se inicia una secuencia en $S$ donde el elemento que la inicia es una primitiva $a$ y llega hasta un punto no terminal $A$ .
2.	$A \rightarrow bB$	Se agrega la primitiva $b$ desde un punto no terminal $A$ , llegando a un punto no terminal $B$ .
3.	$B \rightarrow cC$	Se agrega la primitiva $c$ desde un punto no terminal $B$ , llegando a un punto no terminal $C$ .
4.	$C \rightarrow dD$	Se agrega la primitiva $d$ desde un punto no terminal $C$ , llegando a un punto no terminal $D$ .
5.	$D \rightarrow dD$	Se agrega la primitiva $d$ desde un punto no terminal $D$ , llegando a un punto no terminal $D$ .
6.	$D \rightarrow eA$	Se agrega la primitiva $e$ desde un punto no terminal $D$ , llegando a un punto no terminal $A$ .
7.	$A \rightarrow aA$	Se agrega la primitiva $a$ desde un punto no terminal $A$ , llegando a un punto no terminal $A$ .
8.	$A \rightarrow fB$	Se agrega la primitiva $f$ desde un punto no terminal $A$ , llegando a un punto no terminal $B$ .
9.	$B \rightarrow bB$	Se agrega la primitiva $b$ desde un punto no terminal $B$ , llegando a un punto no terminal $B$ .
10.	$B \rightarrow gC$	Se agrega la primitiva $g$ desde un punto no terminal $B$ , llegando a un punto no terminal $C$ .
11.	$C \rightarrow cC$	Se agrega la primitiva $c$ desde un punto no terminal $C$ , llegando a un punto no terminal $C$ .
12.	$C \rightarrow hS$	Se agrega la primitiva $h$ desde un punto no terminal $C$ , llegando y terminando en el punto de inicio $S$ indicando que termina la secuencia de primitivas para construir la estructura.

Si aplicamos estas reglas en la secuencia en que están enumeradas en la tabla, comprobamos que su aplicación regenera la estructura original, ver Figura 6.6. También en esta figura se observa que al ir

aplicando las reglas se va generando una cadena de primitivas por la aplicación de las producciones que definen la gramática de este experimento. Al aplicar la primera regla o producción se obtiene  $S \Rightarrow aA$ , donde  $(\Rightarrow)$  indica una derivación de cadenas empezando por  $S$ , aplicando la segunda producción se obtiene  $S \Rightarrow aA \Rightarrow abB$ , siguiendo esta lógica se obtiene la siguiente secuencia de derivaciones,  $S \Rightarrow aA \Rightarrow abB \Rightarrow abcC \Rightarrow abcdD \Rightarrow abcddD \Rightarrow abcdddeA \Rightarrow abcdddeaA \Rightarrow abcdddeafB \Rightarrow abcdddeafbB \Rightarrow abcdddeafbgC \Rightarrow abcdddeafbgcC \Rightarrow abcdddeafbgchS$ . Obteniéndose una cadena de primitivas que representa la estructura original "estrella", a partir de las reglas gramaticales.

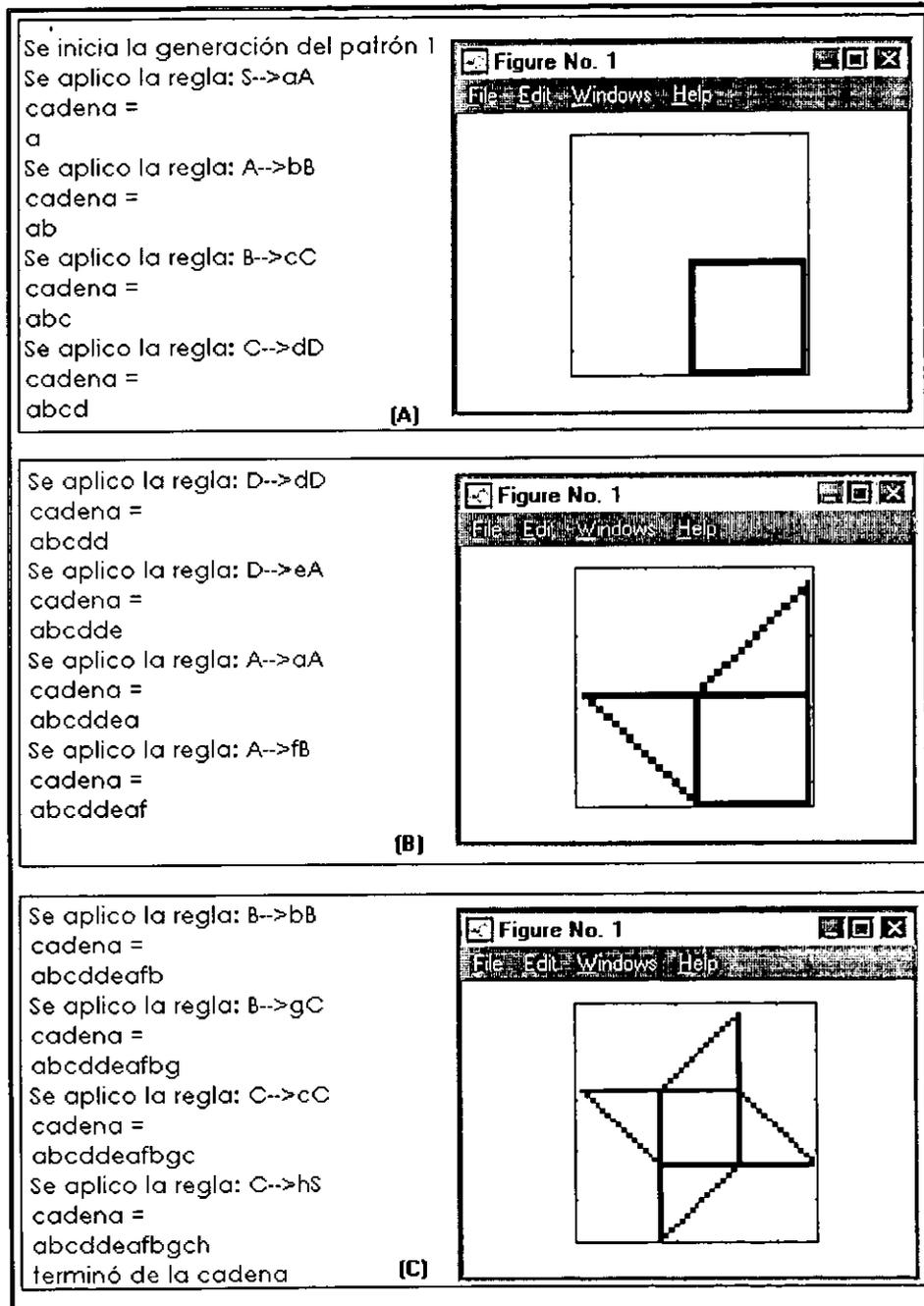


Figura 6. 6 (A) Se aplican las primeras 4 reglas para generar la estructura original; (B) Se aplican las reglas 5,6,7,8 de la gramática generada; y (C) Finalmente se aplican las restantes 4 reglas llegando al fin de la secuencia para generar la estructura original "estrella". [A]

### 6.1.4 Generación de una familia de patrones

Para los fines de este experimento es necesario generar además del patrón “estrella” otros patrones con los que se pueda entrar a un proceso de reconocimiento. Un proceso de reconocimiento para que tenga validez debe tener como objetivo discernir entre uno y otro patrón en sus pequeñas diferencias aunque su forma sea muy similar, es decir, aunque pertenezcan a una misma familia de patrones de características similares, para tal efecto adicionamos una regla sin que con ello se afecte la gramática.

	Producción	Información semántica
13.	$S \rightarrow dD$	Se inicia una secuencia en $S$ donde el elemento que la conforma es una la primitiva $d$ llegando hasta un punto no terminal $D$ .

Igual que hicimos con la estructura “estrella” aplicando una secuencia de reglas para poder generar patrones adicionales al que ya tenemos.

- Aplicando la secuencia de producciones: 13,6,2,9,10,11,12; Obtenemos la cadena: *debbgch*. Y la estructura de la Figura 6.7, patrón número 2.
- Aplicando la secuencia de producciones: 13,6,7,8,9,10,11,12; Obtenemos la cadena: *deafbgch*. Y la estructura de la Figura 6.8, patrón número 3.

Es posible generar un número mayor de patrones con base a las reglas descritas, siempre y cuando estos se apeguen a las reglas comprendidas en la gramática, en caso contrario en el proceso de reconocimiento se generará un rechazo.

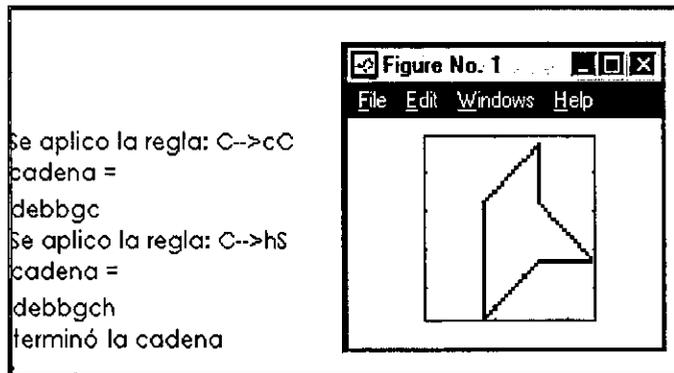


Figura 6. 7 Proceso para la generación del patrón número 2. [A]

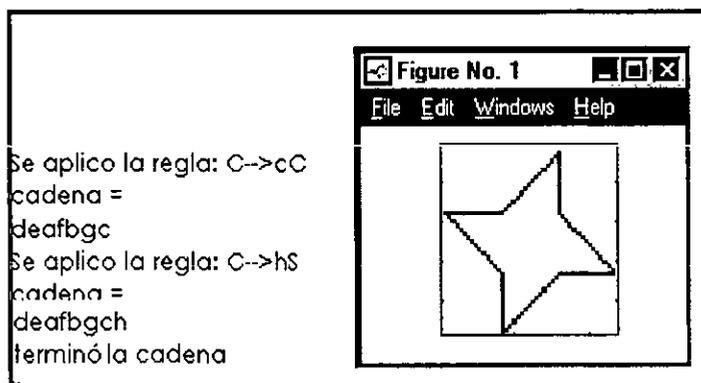


Figura 6. 8 Proceso para la generación del patrón número 3. [A]

### 6.1.5 Diseño de un Automata Finito para el reconocimiento de cadenas

Finalmente procederemos a diseñar el autómata para el reconocimiento de cadenas y por lo tanto un reconocedor de patrones estructurados. Dada una cadena correspondiente a un patrón de entrada, el autómata será capaz de reconocer si dicho patrón pertenece al lenguaje al que está asociado el autómata. Los *autómatas finitos* son los reconocedores de lenguajes generados por las gramáticas regulares. Definimos el autómata de la siguiente manera:

$$A_f = (Q, \Sigma, \delta, q_0, F)$$

[Ecu. 5. 1]

donde:

- $Q$  es un conjunto finito de *estados*, no-vacío,  $\Sigma$  es un *alfabeto* de entrada finito ( $a, b, c, d, e, f, g, h$ ).
- $\delta$  es una *correspondencia* de  $Q \times \Sigma$  ( el conjunto de pares ordenados compuestos por elementos de  $Q$  y de  $\Sigma$ ) con la colección de todos los subconjuntos de  $Q$ .
- $q_0$  es el estado inicial y
- $F$  (un subconjunto de  $Q$ ) es un conjunto de *estados finales*.

Existe una correspondencia biunívoca entre la gramática regular y el autómata finito. Es decir, el autómata finito reconocerá el lenguaje al ser generado por la gramática regular. Partiendo de la definición descrita en el punto 6.3 de este capítulo hacemos las siguientes consideraciones:

- $X_0 \equiv S$
- $N$  se compone de  $X_0$  y de  $n=4$  no terminales adicionales,  $X_1, X_2, X_3, X_4$ .
- El conjunto de  $Q$  del autómata se constituye introduciendo  $n+1$  estados  $\{q_0, q_1, q_2, q_3, q_4\}$ ,
- Si  $X_0 \rightarrow aX_1$  está en  $P$ , entonces  $\delta(q_0, a)$  contiene el estado  $q_1$ .

Siguiendo el mismo procedimiento generamos las siguientes relaciones para generar las siguientes correspondencias para generar el autómata.

1.  $\delta\{q_0, a\} = \{q_1\}$
2.  $\delta\{q_1, b\} = \{q_2\}$
3.  $\delta\{q_2, c\} = \{q_3\}$
4.  $\delta\{q_3, d\} = \{q_4\}$
5.  $\delta\{q_4, d\} = \{q_4\}$
6.  $\delta\{q_4, e\} = \{q_1\}$
7.  $\delta\{q_1, a\} = \{q_1\}$
8.  $\delta\{q_1, f\} = \{q_2\}$
9.  $\delta\{q_2, b\} = \{q_2\}$
10.  $\delta\{q_2, g\} = \{q_3\}$
11.  $\delta\{q_3, c\} = \{q_3\}$
12.  $\delta\{q_3, h\} = \{q_0\}$
13.  $\delta\{q_0, d\} = \{q_4\}$

[Ecu. 6. 2]

Tomando estas correspondencias se genera el siguiente autómata; figura 6.9, para el reconocimiento de cadenas:

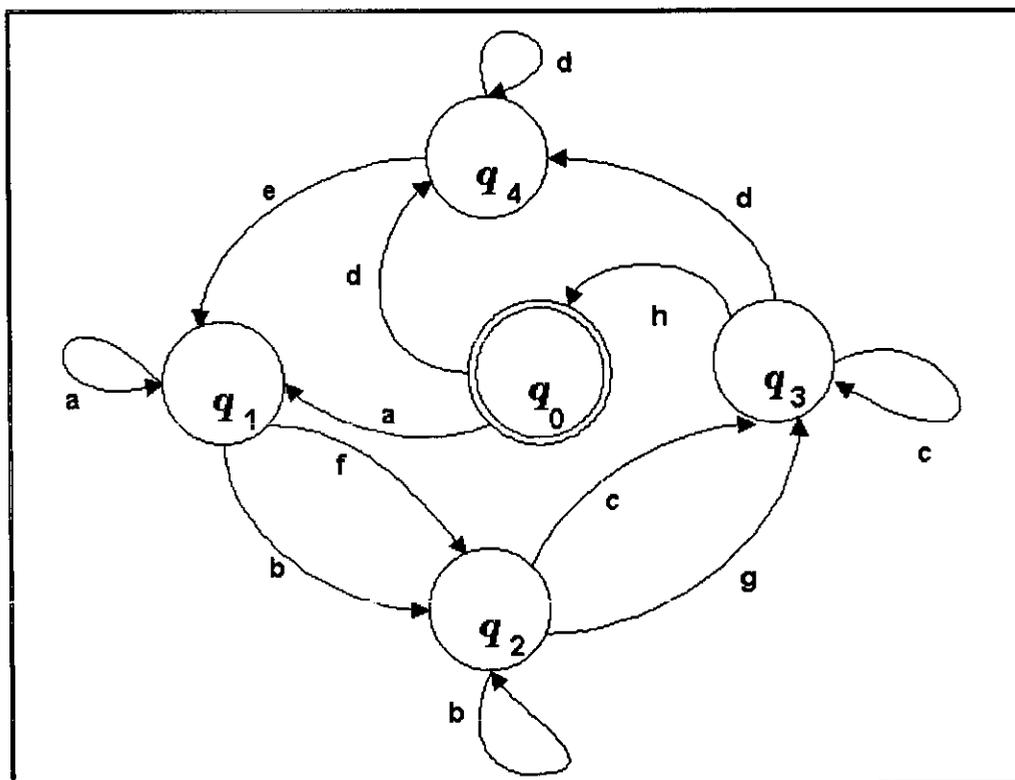


Figura 6. 9 Autómata Finito diseñado para el reconocimiento de cadenas. [A]

Ahora en el siguiente paso introducimos las tres cadenas que hemos obtenido en los procesos anteriores y que corresponden a tres diferentes patrones, esto para verificar que son aceptadas; Así mismo se introducirá una cadena que no corresponde a las reglas del experimento, comprobando con esto que el autómata rechazara las cadenas fuera de la gramática regular propuesta. En la Figura 6.10 observamos la imagen de los tres patrones propuestos para que sean recogidos por autómata finito, dichos patrones se construyeron con los elementos básicos a lo que hemos llamado primitivas, en la construcción de estos patrones son evidentes las similitudes en sus rasgos pero con características propias, un rasgo importante que tienen estos patrones es que son estructuras cerradas es decir que inician y terminan en el mismo lugar siguiendo una secuencia de aplicación de producciones.

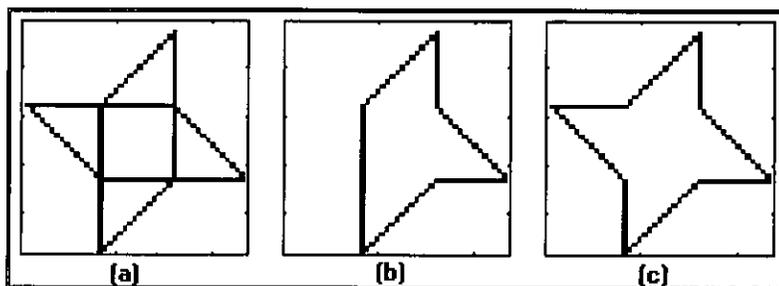


Figura 6.10 (a) Patrón 1 con la cadena: abcddeafbgch; (b) Patrón número 2 con la cadena: debbgch; (c) Patrón número 3 con la cadena: deafbgch. [A]

A continuación se muestra la mecánica y aplicación del programa que simula las acciones del autómata propuesto para este experimento, en su primer paso se definen las cadenas que serán introducidas en el proceso de reconocimiento, enseguida se inicia la ejecución del programa para iniciar la búsqueda, el programa solicita la definición de los valores de la cadena, siguiendo la lógica del autómata (figura 6.9) probará uno a uno los elementos bajo la secuencia de la cadena, si esta se rige por este autómata y es completada la aprobación de los elementos que componen la cadena se indicará que es una secuencia válida y que el patrón está representado por esa secuencia, este procedimiento se sigue para las cadenas, 1, 2, 3. Para comprobar su confiabilidad se somete a prueba una secuencia fuera de la lógica del autómata simulado en el programa, se ingresa esta cadena siguiendo el mismo procedimiento de los anteriores casos, pero ahora llega a un punto en que su secuencia se rompe quedando bajo una indeterminación esto indica que esta fuera de la gramática regular propuesta, siendo rechazada.

```
» cadv
la cadena 1 es: abcddeafbgch
la cadena 2 es: debbgch
la cadena 3 es: deafbgch
» cader
la cadena error propuesta es: afgeddee
;cadena 4
» autoi
```

```
introduce los valores de la
cadena.....cadena1
z =
abcddeafbgch
ans =
a
elemento identificado.....
ans =
b
elemento identificado.....
ans =
c
```

```
elemento identificado.....
ans =
d
elemento identificado.....
ans =
d
elemento identificado.....
ans =
e
elemento identificado.....
a
elemento identificado.....
ans =
f
elemento identificado.....
ans =
b
elemento identificado.....
ans =
```

```
g
elemento identificado.....
ans =
c
elemento identificado.....
ans =
h
cadena correcta e identificada =
z =
abcddeafbgch
y corresponde al patrón 1
» autoi

introduce los valores de la
cadena.....cadena2
z =
debbgch
ans =
d
ans =
d
```

```

elemento identificado.....
ans =
e
ans =
e
ans =
e
ans =
e
elemento identificado.....
ans =
b
elemento identificado.....
ans =
b
elemento identificado.....
ans =
g
elemento identificado.....
ans =
c
elemento identificado.....
ans =
h
cadena correcta e identificada =
z =
debbgch
y corresponde al patrón 2
» autoi

introduce los valores de la
cadena.....cadena3
z =
    
```

```

deafbgch
ans =
d
ans =
d
elemento identificado.....
ans =
e
ans =
e
ans =
e
elemento identificado.....
a
elemento identificado.....
ans =
f
elemento identificado.....
ans =
b
elemento identificado.....
ans =
g
elemento identificado.....
ans =
c
elemento identificado.....
ans =
h
cadena correcta e identificada =
z =
    
```

```

deafbgch
y corresponde al patrón 2
» autoi

introduce los valores de la
cadena.....cadena4
z =
afcddee
ans =
a
elemento identificado.....
ans =
f
elemento identificado.....
ans =
c
elemento identificado.....
ans =
d
elemento identificado.....
ans =
d
elemento identificado.....
ans =
e
elemento identificado.....
ans =
e
ans =
e
error, cadena no valida
    
```

### 6.1.5 Comentario al experimento

En este pequeño experimento aunque no presenta tal vez una aplicación muy practica en el reconocimiento de patrones, si es un experimento didáctico que muestra todos los aspectos que fundamental el reconocimiento sintáctico de patrones y como es aplicada a una estructura sencilla, se hace una clara descripción de cómo una figura estructura en función de sus bordes puede ser fácilmente descrita por elementos simples, ofreciendo con ello grandes posibilidades para su manipulación, demostrando de manera sencilla en los términos en que pueden quedar definidas las primitivas, la ayuda que proporciona la información semántica para la programación de las producciones y la sencillez a que queda reducida la codificación de un patrón, una simple secuencia cadena de primitivas. Esto ofrece una gran cantidad de ventajas las posibilidades de confusión o de error se reducen considerablemente ya que las cadenas generadas son únicas, y están respaldadas por una lógica construida a partir de las producciones descritas por al gramática regular y ofrece también facilidades para transformar esa secuencia sencilla de caracteres en una imagen con un sentido gráfico.

Tal vez las desventajas de este experimento son que no es sometido a muestreo muy gran y aleatorio que nos pudiera arrojar algunas dudas, la simplicidad de la figura nos ahorra considerablemente un error acumulado ya que al no haber preprocesamiento, está garantizada una clara descripción de la imagen de igual manera la generación de las demás imágenes patrón también están sustentadas por las producciones generadas en los primeros pasos del experimento.

El experimento cumple con el objetivo de ayudarnos y mostrarnos detalladamente los elementos básicos del reconocimiento sintáctico de patrones, elementos necesarios para generar un proceso con imágenes mas complicadas y poder hacer uso de nuevos elementos y otros enfoques para particularizar una nueva aplicación.

## 6.2 EXPERIMENTO 2

**Título:** RECONOCIMIENTO SINTACTICO DE ARBOLES.

**Objetivo:** Diseñar un experimento que ejemplifique el reconocimiento sintáctico de una estructura expresada en función de una un árbol utilizando los elementos primitivos apropiados.

**Metodología:** Partimos una imagen donde los elementos que la conforman y por su propia naturaleza son objetos que contienen ramificaciones, sin embargo será necesario hacer una serie de aplicaciones de preprocesamiento para extraer la estructura que se definirá como un árbol. Remarcando que nos enfocaremos fundamentalmente en la construcción de su estructura.

**Proceso:**

- Se hace una análisis de la escena y de sus particularidades.
- Se hace un preprocesamiento, mejora y segmentación de los objetos la imagen para extraer la estructura.
- Se describe el esqueleto del objeto que se definirá como un árbol.
- Se generan los elementos para crear el árbol y las reglas que describen su gramática

### 6.2.1 Análisis de la escena



Figura 6. 11 Imagen, Tormenta Eléctrica[A]

En la figura 6.11 se observa la escena de una tormenta eléctrica, la escena es oscura donde resalta la luz de los relámpagos y la luz del poblado, las características de los relámpagos que aparecen en la escena describen una secuencia continua de un punto a otro en la imagen, salvo el caso del tercero, de izquierda a derecha, el cual presenta ramificación de la base principal e inclusive de esas ramificación se derivan otras más pequeñas. Los objetos de la escena que nos servirán para este experimento serán los relámpagos incluidos en la escena, por lo que como primer paso será el aplicar un preprocesamiento para extraer las estructuras ha manejar.

### 6.2.2 Mejora y Segmentación de la imagen

La imagen de la figura 6.11, tiene un tamaño de 171x111 pixeles con un mapa de color de 256 colores, para la escena será favorable establecerla como una imagen de 64 tonos de gris para acentuar más los contrastes entre la luz y la sombra. El segundo lugar se tratará de acentuar las imágenes de los relámpagos y oscurecer el fondo de la escena y de esta manera segmentar adecuadamente la imagen para extraer y definir los relámpagos como objetos, y posteriormente poder obtener su esqueleto. En la figura 6.12 se observa en las primeras ilustraciones, la imagen original a color y enseguida la imagen convertida a tonos de gris en esa misma figura se incluye la gráfica de su histograma de la imagen en tonos de gris, en esta gráfica se observa que la imagen tiene predominantemente elementos en tonos oscuros y realmente la información luminosa es mínima; en la siguiente imagen de la misma figura se muestra la aplicación de una herramienta para el mejoramiento de la imagen, la cual fundamentalmente consiste en hacer una normalización de los valores de luminosidad en la imagen, tomando el intervalo de valores que están más cercanos al valor máximo de

luminosidad y convertir ese rango en a un valor absoluto de "1" y el restante intervalo de valores fijarlos en cero, es decir en un fondo oscuro, con esto la imagen que se obtiene es una imagen en fondo oscuro y una serie de regiones que llamaremos objetos, logrando con ello una segmentación adecuada, los objetos corresponden a los relámpagos contenidos en la imagen original. Ver figura 6.12.

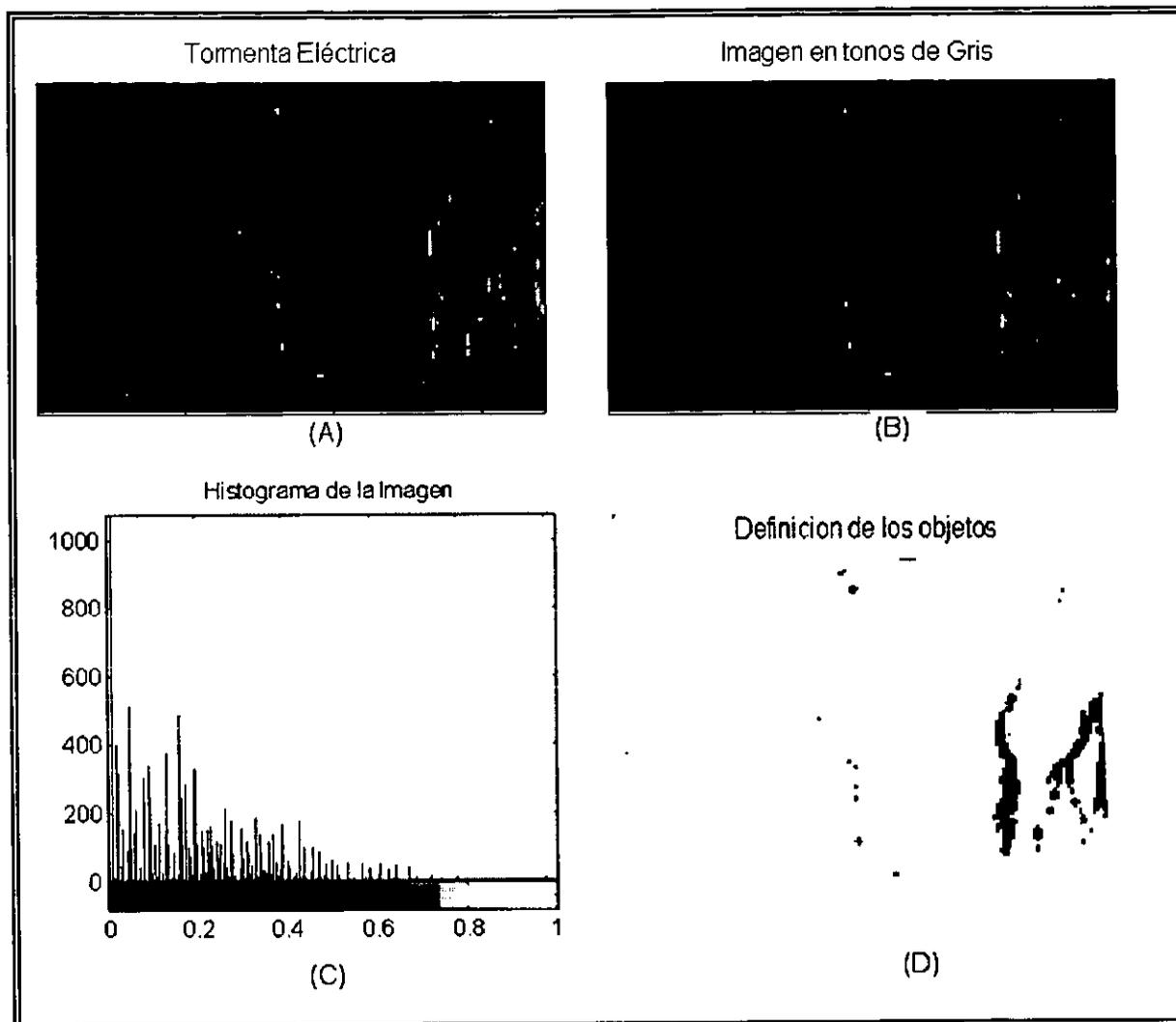


Figura 6. 12 (A) Imagen Original con un mapa de 256 colores; (B) Imagen Convertida en mapa de 64 tonos de gris; (C). Histograma de la imagen en tonos de gris y (D) La definición de los objetos de la imagen después de la segmentación.. [A]

### 6.2.3 Descripción del objeto.

Una vez obtenidos los objetos de la escena se procede a generar sus esqueletos, para obtener el esqueleto simplemente se aplicará una función de adelgazamiento de los objetos hasta obtener la descripción del objeto mediante una línea o sucesión de puntos y de esta manera definir su estructura mediante un árbol, conformado por una raíz y sus ramificaciones. Para los fines de este experimento y como se había comentado en la descripción de la escena el objeto que nos interesa es el relámpago que presenta ramificaciones en su descripción y este se tomará como objeto para el reconocimiento. En la Figura 6.13, la figura muestra la codificación de la imagen con base a grupos de "unos" y "ceros".

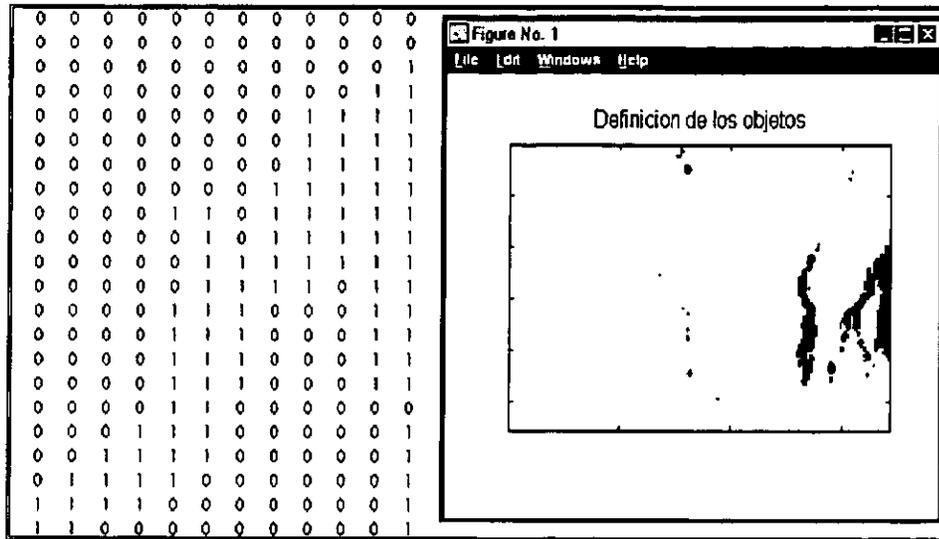


Figura 6.13 A la derecha la definición de los objetos; A la su izquierda la condición digital de la imagen. [A]

En la figura 6.14 las ilustraciones muestran el proceso de adelgazamiento, por medio del cual logramos la definición de los esqueletos, y finalmente se muestra el esqueleto del objeto que servirá para ejemplificar la descripción de objetos mediante un árbol.

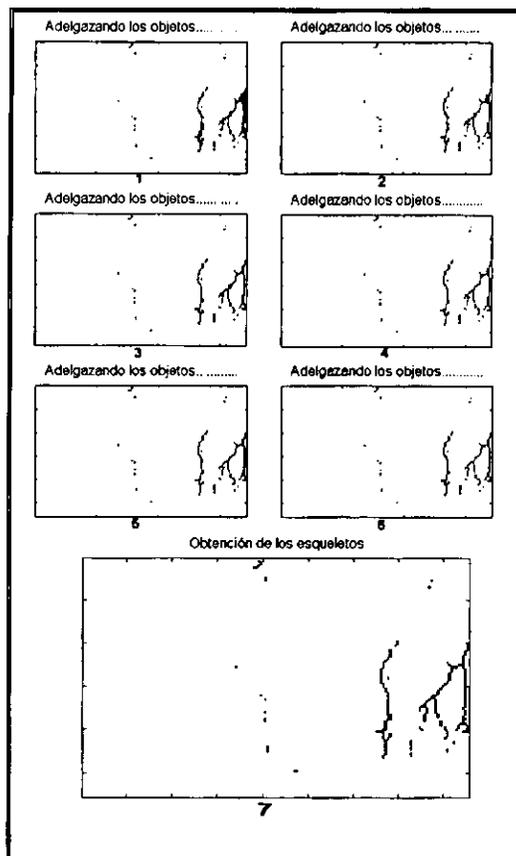


Figura 6.14 De (1 a 6), El proceso paso a paso de adelgazamiento de los objetos; y (7) la obtención de los esqueletos después del adelgazamiento. [A]

Si hace un comparativo entre la imagen original que se ilustra en la figura 6.11 y la imagen en que se detallan los esqueletos, figura 6.14, observamos desde la en la imagen original que los objetos que la componen pueden, por su propia naturaleza, ser definidos mediante una estructura de árbol, sin embargo es necesario someterla a un proceso minuciosos para adecuar y aislar correctamente los objeto que nos interesan de la imagen, requerimiento indispensable para poder seguir con los procesos de codificación y reconocimiento. Si pudiéramos hacer una definición coloquial de este experimento podríamos decir que el objetivo es “De una noche de relámpagos, obtener la estructura de los rayos contenidos en la imagen”.

#### 6.1.4 Definición del árbol y generación de las reglas que lo describen

Como se había establecido al hacer el análisis de la imagen, la estructura que nos interesa después de obtener los esqueletos de los objetos, es la estructura que se encuentra a la derecha de la imagen ya que presenta derivaciones y puede ser configurado como árbol. Ver Figura 6.15.

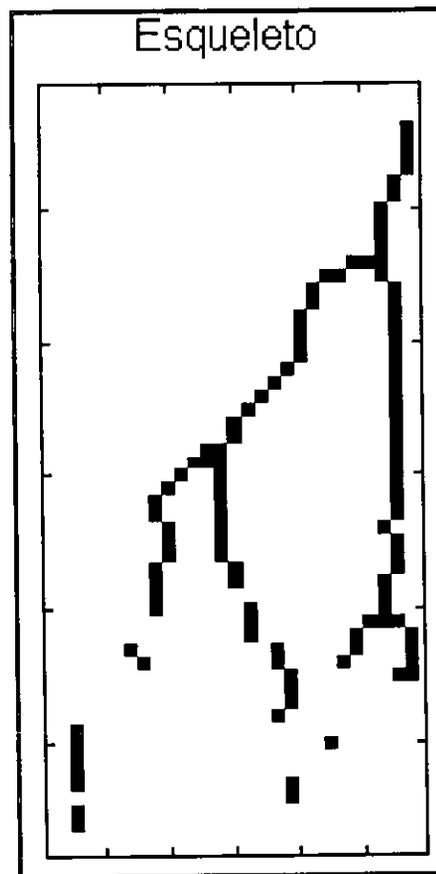


Figura 6. 15. Esqueleto del "relámpago" que aparece del lado derecho en la imagen original. [A]

Una vez aislado el objeto que servirá para la ejemplificación del reconocimiento sintáctico de arboles, procedemos hacer una descripción del esqueleto obtenido. Como vimos, después de hacer la adecuación de los objetos los dos primeros objetos "rayos", quedan descritos como una línea que cruza la escena, la adecuación de estos objetos se reducirían sencillamente a una o dos primitivas y utilizando solo una regla, por ello nos enfocamos al "rayo" que presenta ramificaciones en su estructura.

Para generar el árbol correspondiente tenemos que definir puntos básicos de la estructura, primero definiremos el punto de partida de la estructura, es decir, se encontrará el punto en el se iniciara la raíz del árbol, posteriormente encontramos los puntos donde se generan derivaciones, aquí es importante señalar que los puntos de derivación generaran posteriormente *las funciones de categoría*, de una derivación igualmente puede generarse una nueva derivación, generando así, una nueva *función de categoría* y un nuevo nivel en la estructura del árbol, finalmente definimos los puntos terminales de las ramificaciones para tener una idea cierta de la estructura total.

El reconocimiento sintáctico no es un método ajeno al reconocimiento sintáctico de cadenas, sus fundamentos son esencialmente los mismos, la diferencia radica principalmente en el que cuando uno hace una cadena la forma de las primitivas y el contenido de las mismas nos servirá para tener una adecuada reconstrucción de la figura, en cambio al utilizar una descripción basada en un árbol sintáctico, el objetivo principal es la de definir los niveles en que se ramifica la estructura y no poner tanto énfasis en sus dimensiones y forma de los objetos. Es decir, el objetivo es poder hacer una codificación del comportamiento que sigue la estructura; por esto, es importante hacer una adecuada descripción de los puntos fundamentales que describen el árbol. En la figura 6.16, se observa el proceso de búsqueda de puntos claves en el esqueleto para poder generar un árbol.

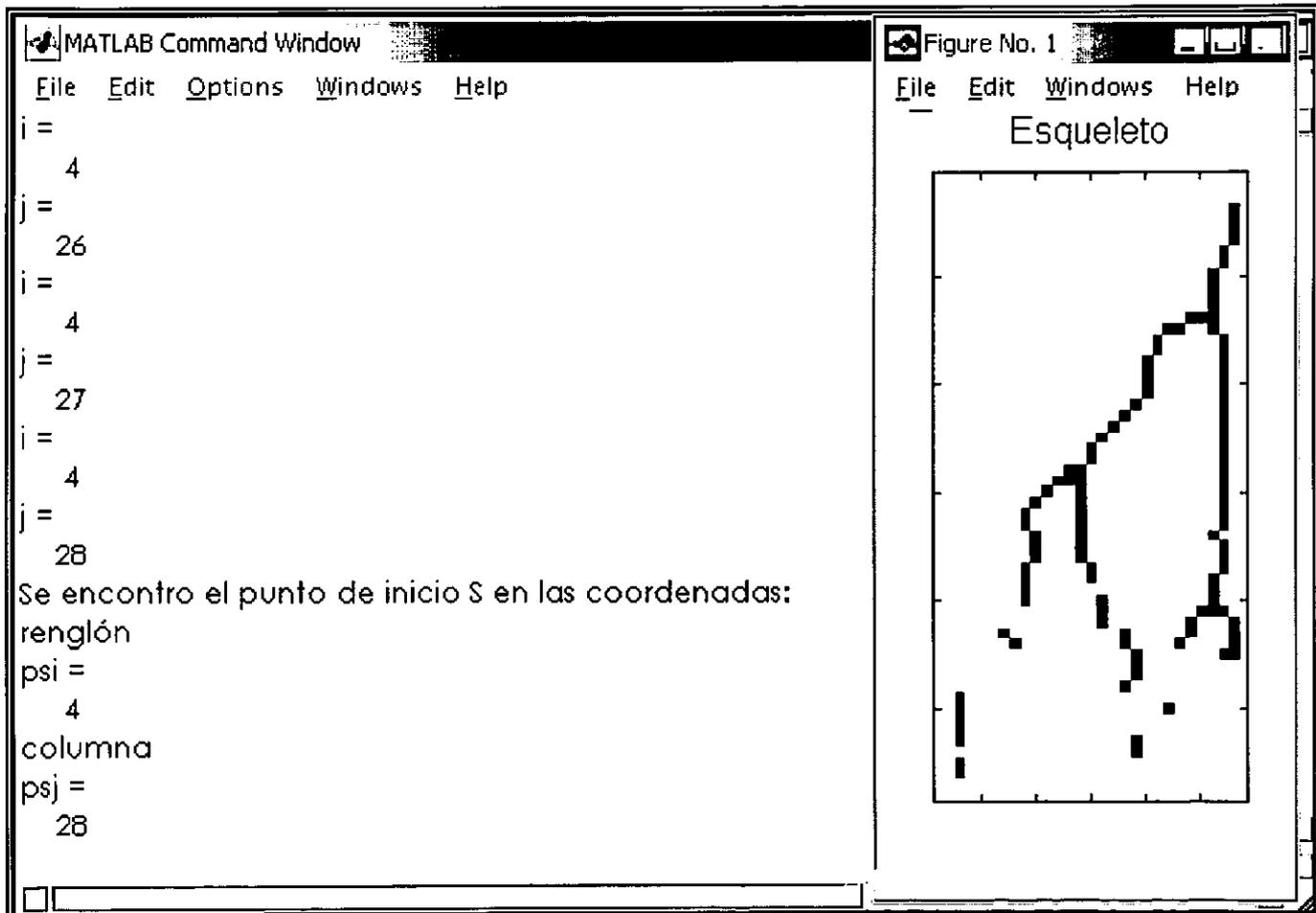


Figura 6. 16 Proceso de búsqueda de puntos básicos. [A]

Una vez finalizado el proceso de búsqueda obtenemos los siguientes puntos:

Los puntos de básicos de la estructura son:

El punto de inicio es:

renglón

psi =

4

columna

psj =

28

Los puntos de derivación son:

G =

14 25

29 13

41 26

Los puntos terminales por la izquierda son:

F =

41 9

45 22

Los puntos terminales por la derecha son:

C =

39 15

46 28

Una vez finalizado el proceso de búsqueda de los puntos básicos, usaremos estos para genera el árbol correspondiente:

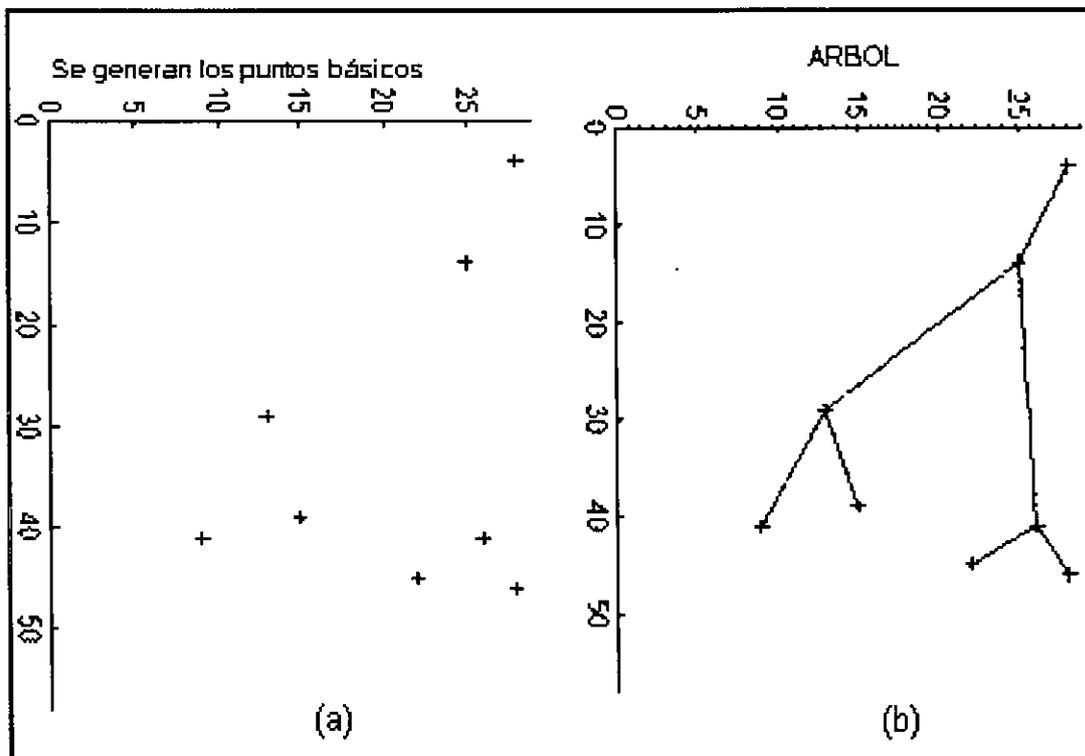


Figura 6. 17 (a) Puntos básicos del árbol; (b) traza del árbol [A]

El primer paso para empezar a definir la gramática que describe el árbol, será definir las primitivas que la integraran. Ver figura 6.18

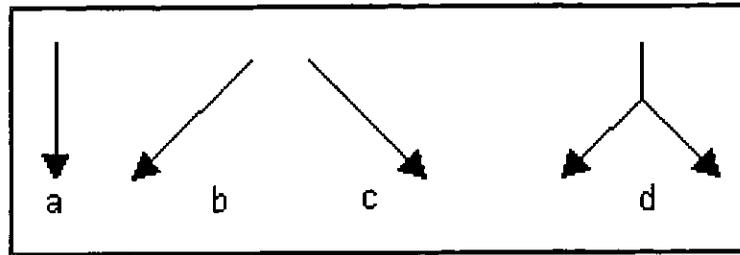


Figura 6. 18 Primitivas para la generación de producciones

Una vez definidas las primitivas cabe hacer algunas observaciones, en general las primitivas no representan estrictamente la longitud ni una orientación en una dirección tan exacta como en el experimento uno, las primitivas para la descripción del árbol tienen la finalidad de traducir su comportamiento, las primitivas reflejaran el punto de inicio ó raíz, el punto en que existe una derivación y si es una derivación por la derecha o por la izquierda.

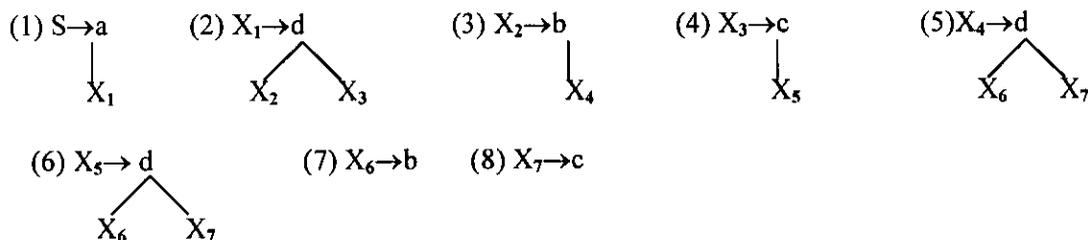
- Siguiendo el procedimiento descrito en la sección 5.6 de este trabajo de tesis, La Gramática del árbol la definiremos mediante la 5-tupla: [1]

$$G = (N, \Sigma, P, r, S)$$

[Ecu. 6. 3]

donde:

- $N$  y  $\Sigma$  son conjuntos de no terminales y de terminales, respectivamente:
  - ♦  $N = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, S\}$
  - ♦  $\Sigma = \{a, b, c, d\}$
- $S$  contenido en  $N$ , es el símbolo inicial,  $P$  es un conjunto de producciones de la forma:



- Las funciones de  $r$  (*función de categoría*), serían:  $r(a) = \{0, 1\}$ ,  $r\{b\} = r\{c\} = \{1\}$  y  $r\{d\} = 2$
- Uso de la semántica, una vez definidas las producciones o reglas gramaticales, es importante hacer una descripción semántica de estas y continuar así, con el proceso de validación de estas reglas:

<i>Producción</i>	<i>Información semántica</i>
(1)	Se inicia el árbol en el punto S (raíz) con la primitiva <i>a</i> y seguimos a un punto no terminal $X_1$
(2)	Del punto no terminal $X_1$ encontramos un punto de derivación <i>d</i> , de donde se ramifica en dos no terminales $X_2$ y $X_3$
(3)	Del punto no terminal $X_2$ agregamos la primitiva <i>b</i> y seguimos a un punto no terminal $X_4$
(4)	Del punto no terminal $X_3$ agregamos la primitiva <i>c</i> y seguimos a un punto no terminal $X_5$
(5)	Del punto no terminal $X_4$ encontramos un punto de derivación <i>d</i> , de donde se ramifica en dos no terminales $X_6$ y $X_7$
(6)	Del punto no terminal $X_5$ encontramos un punto de derivación <i>d</i> , de donde se ramifica en dos no terminales $X_6$ y $X_7$
(7)	Del punto no terminal $X_6$ agregamos la primitiva <i>b</i> llegando a un punto final o frontera.
(8)	Del punto no terminal $X_7$ agregamos la primitiva <i>c</i> llegando a un punto final o frontera

Al aplicar la secuencia de reglas: 1-2-3-4-5-6-7-7-8-8, se garantizará la construcción del árbol, al conjunto de reglas gramaticales se le conoce como "parse", esta secuencia de reglas construirá el árbol en forma descendente por la izquierda en la figura 6.19 se observa en forma gráfica la forma en que se aplican las reglas y como es congruente con la estructura del árbol.

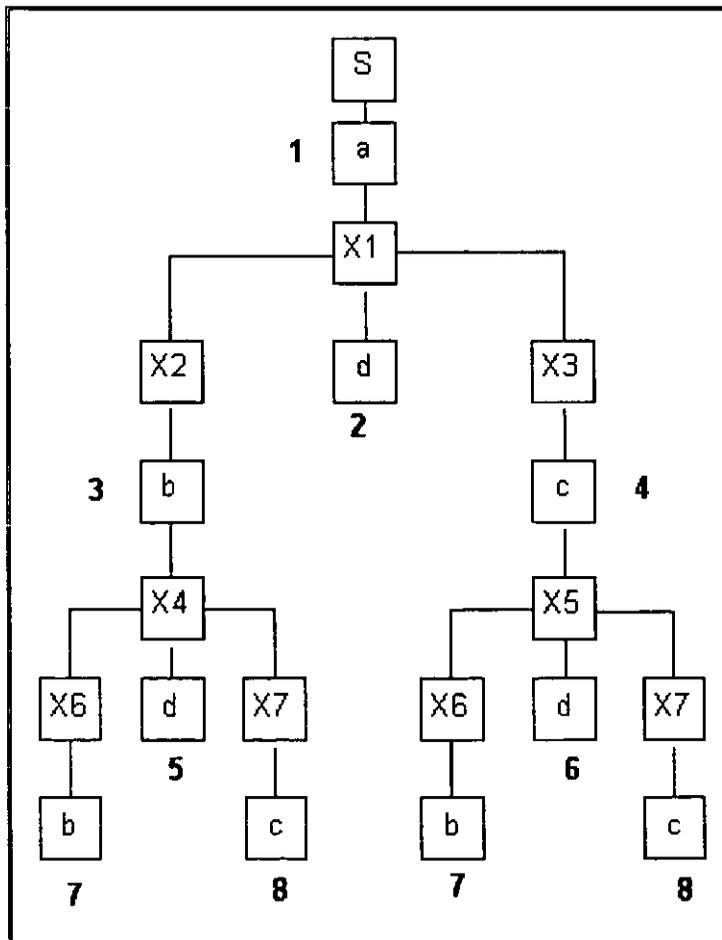


Figura 6. 19 Esquema de aplicación de reglas gramaticales del lenguaje del árbol. [A]

Como vemos en la figura anterior existe un punto inicial y dos niveles de categoría definidos por los puntos en cuales se presentan las derivaciones en el árbol. A diferencia del reconocimiento de cadenas este conjunto de reglas no es el único que garantiza la reconstrucción e identificación del árbol, las ramificaciones nos pueden dar varias opciones de construcción, el árbol también puede estar representado por los siguientes grupos de reglas:

1. 1-2-4-3-6-5-8-8-7-7 (Construcción descendente por la derecha)
2. 8-8-7-7-6-5-4-3-2-1 (Construcción ascendente por la izquierda)
3. 7-7-8-8-5-6-3-4-2-1 (Construcción ascendente por la derecha)

El autómata para el reconocimiento tiene la capacidad para validar cualquiera de estos conjuntos de reglas. Como hemos venido recalcando; para el reconocimiento sintáctico de árboles lo importante es tener una correcta definición de las reglas o producciones, la exactitud de la forma de los patrones quedara encomendada al reconocimiento sintáctico de cadenas, basado en los detalles contenidos en las primitivas. La combinación de ambas técnicas reconocimiento sintáctico de cadenas y arboles generará una mayor precisión en su estructura y forma de los objetos que necesiten una presión grande, esto dependerá de los objetivos de la aplicación y de la precisión que se quiera en el reconocimiento.

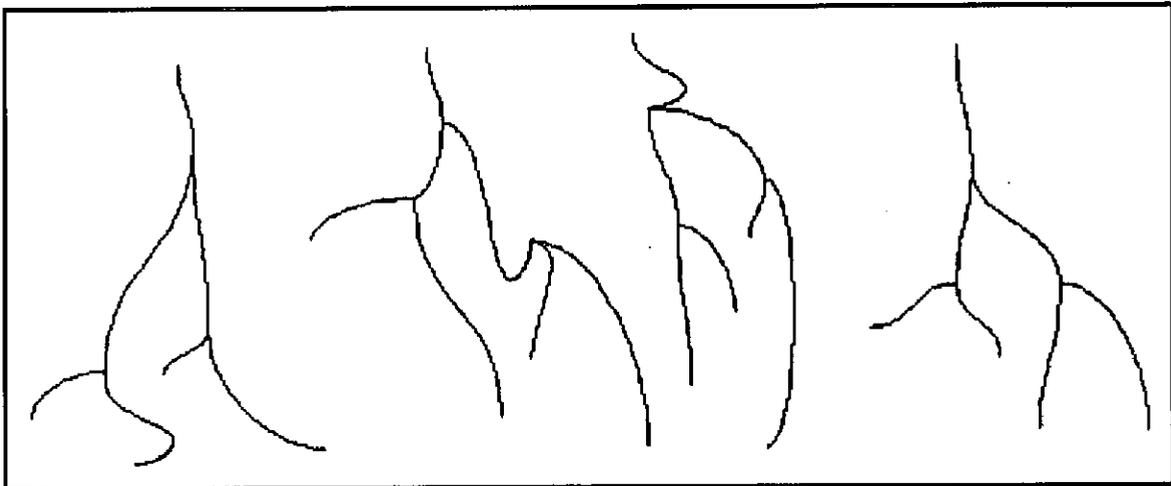


Figura 6. 20 Diversos esquemas de "rayos" que pueden ser descritos mediante las reglas generadas para este experimento [A]

### 6.2.5 Comentario al experimento

La estructura obtenida ejemplifica la aplicación del análisis sintáctico para una estructura de árbol, el análisis de una estructura mediante un árbol nos facilita en mucho la exactitud para reconocer la estructura de los objetos y olvidarnos de sus dimensiones, teniendo un comportamiento que se pueda implementado con las reglas gramaticales, la estructura será reconocida; Además ofrece la posibilidad de manejar mas niveles de caracterización de la imagen, manejar formas que no son cerradas, es decir, que no empiecen y terminen en el mismo punto.

Por otro lado también aportó información valiosa y ejemplificó algunas técnicas descritas en los temas de mejoramiento y segmentación, la correcta adecuación de la imagen abrió el camino para que desde ese punto empezáramos a discriminar objetos que estaban fuera de los objetivos de este experimento, la imagen original contenía una pequeña porción de información que se pudiera interpretar, por ello era necesario someterla a un proceso que aislara la información conveniente para el experimento. Después de los procesos

de segmentación y descripción la información para este experimento fue mínima pero consistente en su interpretación visual, cumpliendo una de las premisas de este trabajo, cualquier objeto o figura se puede ser representado con una porción básica de información al ser describirla mediante una estructura basada en líneas o bordes sin perder su interpretación visual.

### *II COMENTARIO AL CAPITULO II*

Con los experimentos descritos en este capítulo se ejemplificó el reconocimiento sintáctico de patrones y en cada uno de ellos nos planteamos los objetivos y al final de cada uno de ellos comentamos sus alcances y resultados. Como menciono al principio de este capítulo los programas están escritos para ser ejecutados en el programa de simulación matemática Matlab™, los requerimientos de máquina para que puedan visualizarse estos programas son los mismos que se requieren para la instalación del simulador la velocidad de los procesos dependerá de la velocidad de la máquina en que se ejecute el programa, además que cuente con las herramientas para procesamiento digital de imágenes; en cuando a memoria no requiere grandes recursos las imágenes que se procesaron no son de grandes dimensiones, la imagen para el segundo experimento necesita solo 18K[bytes] de memoria y no contienen mapas de colores complejos, 256 colores máximo.

Finalmente pasaremos a las conclusiones generales de este trabajo de Tesis.

· C · A · P · I · T · U · L · O ·

VII

CONCLUSIONES

## CONCLUSIONES

En un sistema de Procesamiento Digital de Imágenes (PDI) que tiene como fin el reconocimiento de patrones, involucra varias etapas en su proceso, donde se establece la secuencia de aplicación entre ellas. Para caracterizar un objeto contenido en una imagen, esta tendrá que pasar por un preprocesamiento para que el objeto sea viable de ser reconocido.

Tener una clara definición en el contraste de luminosidad y una buena definición de colores en los objetos contenidos en una imagen, garantiza la existencia de bordes que definen la forma de los objetos; la ambigüedad en la definición de los objetos por deficiencias de luminosidad incorpora errores en las siguientes etapas del proceso.

Una de las premisas sostenidas en este trabajo es la viabilidad de conservar la comprensión visual de los objetos al ser descritos por sus bordes, trabajando con los bordes reduciremos considerablemente la información visual ahorrando tiempo de procesamiento en máquina.

Los métodos del dominio espacial efectúan una manipulación directa de los píxeles, usando los niveles de luminosidad para conseguir un mejoramiento en la imagen. Los filtros digitales en el dominio espacial, con una estructura muy sencilla ejecutan un minucioso proceso en cada pixel de la imagen.

Un filtro paso bajo elimina el ruido contenido en la imagen, suavizándola; un filtro paso alto tiene el efecto de acentuar las líneas que definen los objetos, ambos procesos valiosos para los fines de este trabajo.

Después de resaltar los bordes es necesario aislarlos para formar la estructura, para esto la segmentación de frontera se aboca a la detección confiable de bordes asociada a los cambios umbrales en el mapa de la imagen.

La segmentación bajo los operadores de segunda derivada garantizan una correcta definición de los bordes, además de eliminar la textura de los objetos.

El proceso de descripción aporta información analítica a partir de la información visual de la estructura con el objetivo de tomar elementos básicos de la estructura y generar un conjunto de descriptores, llamados *primitivas* en el proceso de reconocimiento.

La definición de primitivas es una etapa compleja que demanda un proceso detallado de codificación, esto puede aminorarse utilizando y combinando descriptores de forma, para generar una estructura más simple y fácil de codificar ocupando un número pequeño de datos.

El estudio de los métodos sintácticos nos da la posibilidad de trabajar en función de la estructura de los objetos, este método para el reconocimiento de patrones ofrece varias facilidades para su implementación:

- No necesariamente se debe describir toda la estructura, definiendo elementos básicos se garantiza su reconstrucción.
- Las reglas gramaticales son fáciles de describir.
- Cada regla aporta información de sintaxis, orientación importante al momento de programarla en la computadora.
- Los patrones son descriptores en función de elementos y relaciones sencillas, como elementos del alfabeto y metacaracteres.

- El proceso de reconocimiento se reduce a un análisis de una cadena de elementos que describen la estructura, en función de un autómata.
- Al aplicar una secuencia correcta de reglas se garantiza la reestructuración de la estructura ya que los descriptores contienen la información para volver a trazarla.

Sobre la base de los experimentos descritos en el último capítulo de este trabajo nos permite establecer las siguientes conclusiones.

Los experimentos que se diseñaron en el último capítulo tuvieron como objetivo principal el mostrar las técnicas y algoritmos que se describen en el capítulo 5 de este trabajo. El **experimento uno** tuvo el objetivo de ilustrar todos los aspectos que involucra el reconocimiento sintáctico de cadenas, este experimento demuestra claramente una de las premisas de este trabajo: poder representar una figura contenida en una imagen en función de una cadena de elementos simples, elementos básicos que llamamos primitivas, la representación de objetos mediante cadenas reduce el proceso de reconocimiento a un proceso informático bien conocido en materia de estructuras de datos, encontrar una secuencia de caracteres contenidos en una cadena de datos, el reconocimiento mediante un autómata en un método fácil de implementar por su disposición visual para entender el proceso de búsqueda y aceptación de la cadena.

En PDI, el proceso de encontrar e identificar una secuencia de valores conlleva usar esta información con implicaciones visuales muy interesantes, como poder establecer un método de decisión visual efectivo para el reconocimiento de patrones, siempre y cuando esté respaldado por un preprocesamiento y descripción que presente deficiencias mínimas en la estructura de los objetos. La correcta generación de reglas o producciones que conforman la gramática del lenguaje de reconocimiento garantizará la regeneración y reconocimiento de los objetos que sean definidos mediante cadenas de primitivas.

En el segundo experimento aunque es un experimento básico para el reconocimiento sintáctico de árboles, nos aporta otros aspectos importantes que se deben tomar en cuenta al trabajar con imágenes. Hay que tener siempre claro que el PDI es un proceso de varias etapas en donde regularmente la imagen necesitará de una adecuación en sus características de color o de formato, estos procesos son igualmente importantes que el de reconocimiento, pues en ellos caracterizaremos los objetos que se encuentran dentro de la imagen en líneas o bordes, acción fundamental para las etapas de descripción y reconocimiento que se estudiaron en este trabajo.

En el **experimento dos** el principal objetivo es hacer una correcta definición de los puntos que describen su estructura, esto garantiza la reconstrucción de la misma y nos servirá como base para escribir las reglas gramaticales del lenguaje del árbol que describen el mecanismo de construcción de la estructura. Cuando la descripción de objetos se enfoca solo a la estructura, nos evita problemas con la identificación en sus dimensiones y en su orientación, si se quiere una mayor precisión es necesario incorporar las características descritas en el proceso para el reconocimiento sintáctico de cadenas.

El reconocimiento sintáctico de árboles, aporta nuevas herramientas para un mayor número o complejidad de patrones, al poder establecer varios niveles de caracterización. En cada uno de esos niveles puede definir un proceso particular de reconocimiento y de análisis, que podrá interactuar en un sistema de varias áreas definidas con base a un árbol sintáctico.

La primicia de este trabajo nos remite a establecer que muchos objetos pueden ser caracterizados por sus bordes y contornos sin perder su comprensión visual por compleja que sea la escena representada en una imagen digital.

Existen muchos enfoques y algoritmos que facilitan la extracción de este tipo de rasgos, el filtrado digital acondiciona bastante bien los rasgos importantes aún cuando estos estén contenidos en una imagen densa de

información. Los patrones pueden tener desde su propia naturaleza una descripción basada en líneas, como lo son las huellas dactilares, o estructuras celulares simples. En el caso de huellas dactilares la estructura de la impresión gráfica de los dedos forma una figura "concéntrica" de líneas que presentan pequeñas derivaciones llamadas "minucias", puntos que describen las características individuales de cada dedo, estos puntos pueden servir para crear una estructura de árbol aunque no sea una estructura vertical, con el reconocimiento sintáctico de árboles puede reconocerse el patrón de las huellas dactilares con base en los puntos llamados "minucias"; con la ventaja de que la imagen por su naturaleza está construida con líneas, ahorrando tiempos de preprocesamiento.

Los experimentos propuestos en el capítulo 6 de este trabajo, tienen un enfoque demostrativo y los requisitos de respuesta no son relevantes para estas aplicaciones, ya que son experimentos controlados y de alcances básicos, y aunque en el segundo experimento se toma una imagen que puede aportar información de fenómenos físico de la meteorología, solo con base en conocimientos de un experto podrían aportarse conclusiones adicionales al experimento, para los fines de nuestro trabajo lo importante fue implementar un proceso que pueda generalizarse para otras aplicaciones.

Existe una gran variedad de aplicaciones en diferentes ámbitos en los que la identificación visual por medio de imágenes es una herramienta básica para la toma de decisiones, este trabajo tuvo como objetivo describir un método eficiente en su algoritmo y filosofía que presente una alternativa para una variedad de imágenes. El método es sencillo en su descripción y formulación, el trabajo exhaustivo se presenta en el proceso de programación donde se requiere un programador hábil que pueda traducir las descripciones gramaticales a un programa de alto nivel que obviamente tenga una interface visual que vinculen todos los procesos que intervienen en el reconocimiento sintáctico. Aún con esto no es necesario incrementar su demanda de recursos de máquina, la demanda dependerán del entorno de programación.

El reconocimiento sintáctico de patrones al tener como filosofía la teoría de lenguajes, sumamente conocida en procesos informáticos, ofrece varias alternativas y métodos de programación. El reconocimiento estructural tiene muchos matices que lo hacen muy completo, puede fácilmente regenerar las estructuras, los lenguajes pueden crecer y combinarse, de una estructura simple puede generarse una gran red de éstas, y crear un sistema complejo de reconocimiento visual. Por esto el reconocimiento sintáctico es un método importante en el reconocimiento de patrones estructurados.



·A· ·N· ·E· ·X· ·O·

A

NOCIONES BASICAS DE  
TEORIA DE CONJUNTOS

## ANEXO A

## NOCIONES BASICAS DE TEORIA DE CONJUNTOS

Partiremos de forma acrítica, de la idea intuitiva de *conjunto* como reunión de objetos o elementos, y supondremos que dado cualquier objeto siempre es posible determinar si este pertenece o no al conjunto.

Se suele denotar con las letras minúsculas a los elementos y con mayúsculas a los conjuntos. Por ejemplo,  $A=\{a,e\}$  denota un conjunto  $A$  formado por los elementos  $a$  y  $e$ . Diremos que  $a \in A$ . En este punto, aceptaremos que los conjuntos pueden quedar definidos bien por *extensión* (enumeración de todos los elementos) o por *comprensión* (enunciación de una propiedad característica). A partir de aquí, utilizaremos las siguientes definiciones y propiedades:

- $A$  es *parte* o subconjunto de  $B$  *está incluido en*  $B$  ( $A \subset B$ )  $\Leftrightarrow \forall x : x \in A \Rightarrow x \in B$ . La relación entre  $A$  y  $B$  se denomina relación de inclusión y diremos también que  $B$  *contiene* a  $A$  ( $B \supset A$ ).
- $A$  es *idéntico* a  $B$  ( $A=B$ )  $\Leftrightarrow A \subset B \wedge B \subset A$ .
- $A$  es *parte propia* o *subconjunto propio* de  $B$ , o *está propiamente incluido en*  $B$   $\Leftrightarrow A \subset B \wedge A \neq B$ .
- Se llama *conjunto vacío* ( $\emptyset$ ) al conjunto que no contiene ningún elemento y que satisface  $\forall A : \emptyset \subset A$ .
- El *conjunto de las partes* de un conjunto ( $P(U)$ ) es el conjunto formado por todos los subconjuntos de  $U$  :  $A \in P(U) \Leftrightarrow A \subset U$ .
- El conjuntos *unión* de  $A$  y  $B$  :  $A \cup B = \{x : x \in A \vee x \in B\}$
- El conjunto *intersección*  $A$  y  $B$  :  $A \cap B = \{x : x \in A \wedge x \in B\}$
- $A + B = A \cup B$
- $A - B = A \cap B$  (*diferencia*).
- $A \oplus B = (A - B) + (B - A)$  (*diferencia simétrica*).
- $A \times B = \{(x,y):x \in A \wedge y \in B\}$  (*producto cartesiano*)

## A.1 Correspondencia y Relaciones

Dados dos conjuntos  $A$  y  $B$ , se llama *correspondencia*  $f$  de  $A$  en  $B$  ( $f:A \rightarrow B$ ) a un subconjunto de dado  $F$  del producto cartesiano  $A \times B$ , y se denota:

$$f(x) = \{y \in B : (x,y) \in F \subset A \times B\}$$

Al conjunto  $A$  se llama *conjunto inicial* ( $A=\text{in}(f)$ ) y al  $B$  *conjunto final* ( $B=\text{fin}(f)$ ). También se definen:

- $or(f) = \{x \in A : (x,y) \in F \text{ para algún } y \in B\}$  (conjunto original).
- $im(f) = \{y \in B : (x,y) \in F \text{ para algún } x \in A\} = \cup_{x \in A} f(x)$  (conjunto imagen).

Una correspondencia se denomina:

- *unívoca*  $\Leftrightarrow \forall x \in A : f(x)$  contiene como máximo un elemento
- *aplicación*  $\Leftrightarrow \forall x \in A : f(x)$  contiene un único elemento. Si  $y$  es el único elemento de  $f(x)$ , escribiremos  $y = f(x)$ .

A su vez, una aplicación puede calificarse como:

- *inyectiva*  $\Leftrightarrow \forall x, y \in A : f(x) = f(y) \Rightarrow x = y$
- *suprayectiva*  $\Leftrightarrow im(f) = fin(f)$
- *biyección*  $\Leftrightarrow f$  es aplicación inyectiva y suprayectiva. Una biyección establece un “emparejamiento” de cada elemento del conjunto inicial con uno del final y viceversa, y se denota  $f: A \leftrightarrow B$
- Una *aplicación interna* en  $A$  es una aplicación  $c: A \times A \rightarrow A$  de forma que  $z = c(x,y) \in A$ . Si  $c$  es asociativa se dice que  $A$  es un *semigrupo*. Si, además, existe un elemento neutro, entonces  $A$  es un monoide.
- *homomorfismo*  $\Leftrightarrow f: A \rightarrow B$  es una aplicación entre semigrupos o monoides y conserva las operaciones internas  $c, c'$  definidas en los conjuntos inicial y final respectivamente

$$\forall x, y \in A: f(c(x, y)) = c'(f(x), f(y))$$

Una correspondencia de  $A$  en  $A$  recibe el nombre de *relación*:

$$xRy \Leftrightarrow (x, y) \in R \subset A \times A$$

Una *relación* es de *equivalencia* si es:

- reflexiva:  $xRx \vee x \in A$
- simétrica:  $xRy \Rightarrow xRz, x, y, z \in A$
- transitiva:  $xRy \wedge yRz \Rightarrow xRz, x, y, z \in A$
- Una *relación de orden* tiene las propiedades:
- reflexiva
- antisimétrica:  $xRy \wedge yRx \Rightarrow x=y, x, y \in A$ .

— *transitiva*

Dada una relación  $R$  de equivalencia en  $A$ , se definen sus *clases de equivalencia* como:

$$C(x) = \{y \in A : xRy\}$$

Una *partición* de  $A$ , es una colección de subconjuntos  $\{C_i\}$  tales que:

- $C_i \neq \emptyset$
- $\cup_i C_i = A$
- $C_i \cap C_j = \emptyset \Leftrightarrow i \neq j$

Dada  $R$  de orden en  $A$ , se denota:

- $a \leq b \Leftrightarrow (a,b) \in R \subset A \times A$
- $a < b \Leftrightarrow a \leq b \wedge a \neq b$
- $[a,b] = \{x \in A : a \leq x \leq b\}$

Se dice que  $C \subset A$  es *convexo*  $\Leftrightarrow \forall x, y \in C : [x,y] \subset C$

Se define  $C$ , *clausura* o *cierre convexo* de  $C$ , como el menor conjunto convexo que contiene a  $C$ . El concepto de clausura es un concepto bastante general, y siempre se entiende como una extensión del conjunto de partida, hasta que este adquiere una cierta propiedad.

### A.2 Cardinal. Conjunto infinito.

La noción intuitiva de cardinal de un conjunto (el número de elementos que contiene) no está exenta de problemas, sobre todo cuando se considera la existencia de conjuntos infinitos. Por otro lado, en la moderna teoría matemática este concepto resulta clave para poder introducir de forma rigurosa los números naturales. Por ello se debe definir de forma precisa lo que se entiende por cardinal o potencia de un conjunto.

Diremos que dos conjuntos  $A$  y  $B$  son equipotentes ( $A \cong B$ ) si existe una biyección entre ellos  $f : A \leftrightarrow B$ . Esta definición genera una partición entre los conjuntos, ya que la relación que se establece es de equivalencia.

Si  $A \cong B$  diremos que  $A$  y  $B$  tiene el mismo cardinal, y escribiremos:  $\text{card}(A) = \text{card}(B)$ .

Interesa definir una relación de orden entre los cardinales. Ello puede hacerse de la siguiente forma:

$$\text{card}(A) \leq \text{card}(B) \Leftrightarrow A \cong C \subset B$$

Las propiedades reflexiva y transitiva de esta relación son inmediatas.

Los números naturales se introducen como símbolo para denotar cardinales, y se define de forma constructiva, a partir de la siguiente sucesión de conjuntos:

$$C_0 = \emptyset$$

$$C_1 = \{\emptyset\}$$

$$C_2 = \{\emptyset, \{\emptyset\}\}$$

$$C_3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$$

M

Cada conjunto se define como reunión de todos los objetos (en este caso conjunto) previamente definidos. De esta forma, los números naturales no son más que etiquetas para denotar los cardinales de los conjuntos de la sucesión anterior. Por ejemplo,  $3 = \text{card}(C_3)$ . Si un conjunto  $A$  es equipotente con alguno de los  $C_n$  anteriores, entonces su cardinal es el número natural  $n \in \mathbb{N}$  y se dice que  $A$  es *finito*.

Un conjunto *infinito* no puede ponerse en biyección con ninguno de los  $C_n$  anteriormente definidos, y posee, además, la sorprendente propiedad (que suele tomarse como definición) de que puede en biyección con una parte propia de sí mismo. Por ejemplo:

$$f(n) = 2n, f: \mathbb{N} \leftrightarrow \{n \in \mathbb{N}, n \text{ es par}\}$$

$$f(x) = \frac{x}{1-x^2}, f: (-1,1) \leftrightarrow \mathbb{R}$$

Otra propiedad importante de los conjuntos infinitos es que no todos tiene el mismo cardinal, sino que existe una sucesión infinita de potencias posibles diferentes. Esto es consecuencia del teorema de Cantor, que afirma que el cardinal del conjunto de las partes  $P(A)$  es estrictamente mayor que el cardinal de los conjuntos de mayor cardinalidad que cualquiera dada:

$$\text{card}(A) < \text{card}(P(A)) < \text{card}(P(P(A))) < \dots$$

El menor cardinal infinito es el del conjunto  $\mathbb{N}$  de los números naturales. A todos los conjuntos equipotentes a  $\mathbb{N}$  se les denomina infinitos numerables (ya que sus elementos se pueden contar) Por tanto,  $A$  se dice *numerable* si  $A \cong \mathbb{N}$ . El siguiente cardinal conocido es el de  $\mathbb{R}$ , conjunto de los números reales. Si  $A \cong \mathbb{R}$  se dice que  $A$  tiene la *potencia del continuo* (por ser los números reales representables como puntos sobre una recta, la recta real).

La cuestión si existe algún conjunto cuyo cardinal  $c$  sea tal que  $\text{card}(\mathbb{N}) < c < \text{card}(\mathbb{R})$  es un ejemplo clásico de teorema indefinible dentro de la aritmética.

R · E · F · E · R · E · N · C · I · A · S ·

B

B I B L I O G R A F I A

# Bibliografía

## [REFERENCIAS BIBLIOGRAFICAS]

- [1] González Rafael C., Woods Richard E. TRATAMIENTO DIGITAL DE IMAGENES, Addison-Wesley Iberoamérica, E.U., 1996.
- [2] Lira Jorge. LA PERCEPCION REMOTA: NUESTROS OJOS DESDE EL ESPACIO, Fondo de Cultura Económica, México, 1987.
- [3] Chuvieco Emilio. FUNDAMENTOS DE TELEDETECCION ESPACIAL, Ediciones Rialp, Madrid, 1990.
- [4] Button & Button. PHOTOSHOP 4.0, Ed. Prentice Hall, 1997.
- [5] Greenber & Greenber. MANUAL DE PHOTOSHOP 4, Ed. Osbara-McGraw Hill, Madrid, 1997.
- [6] Bustos Martín I. MULTIMEDIA, Ed. Anaya Multimedia, Madrid, 1995.
- [7] Duda & Hart. PATTERN CLASIFICACION AND SCENE ANALYSIS, Jhon Wiley & Sons, E.U., 1973.
- [8] Jukka Iivarien, Makus Peura, Jaakko Sarela, Ari Visa. COMPARISION OF COMBINED SHAPE DESCRIPTORS FOR IRREGULAR OBJECTS, in the proceedings of the Eighth British Machini Vision Conference, volumen 2 pages 430-439, University of Essex, UK, September 8-11, 1997.
- [9] Pavlidis T. STRUCTURAL PATTERN RECOGNITION, Sprin-Verlag Berlin Heidelberg, 2<sup>nd</sup>. Edición, Germany, 1980.
- [10] Alfonseca M., Sancho J., Martínez Orga M. TEORIA DE LENGUAJES, GRAMATICA Y AUTOMATAS, Universidad y Cultura, Madrid, 1990.
- [11] Castleman R. Kenneth. DIGITAL IMAGEN PROCESSING, Ed. Prentice -Hall, E.U., 1979.
- [12] González Rafael C., Wintz Paul. DIGITAL IMAGEN PROCESSING, Addison-Wesley, E.U., 1977.
- [13] Lira Jorge. APUNTES DE PROCESAMIENTO DIGITAL DE IMAGENES,
- [14] Ole Vilhelm Larsen, Artículo: STATISTICAL CHARACTERISTICS OF STOCHASTICS CONTEXT-FREE LENGUAGES, Intitute of Electronic Systems Aalborg University Denmark, second revised edition, july 1995.
- [15] Muñoz Conejo Ricardo. APUNTES: PROCESADORES DE LENGUAJE I, referencia: <http://apolo.lcc.uma.es/docencia/ETSIInf/pl/p11.html>.
- [16] Nakamura Shoichiro. ANALISIS NUMERICO Y VISUALIZACION GRAFICA CON MATLAB, Prentice-Hall Hispanoamericana, México, 1997.

**[REFERENCIAS DE FIGURAS]**

[A] Procesada y/o editada por el autor de la tesis.

[B] González Rafael C., Woods Richard E. TRATAMIENTO DIGITAL DE IMAGENES, Addison-Wesley Iberoamérica, E.U., 1996.

[C] Chuvieco Emilio. FUNDAMENTOS DE TELEDETECCION ESPACIAL, Ediciones Rialp, Madrid, 1990.

[D] Bustos Martín I. MULTIMEDIA, Ed. Anaya Multimedia, Madrid, 1995.

[E] Button & Button. PHOTOSHOP 4.0, Ed. Prentice Hall, 1997.

[F] Jukka Iivarién, Markus Peura, Jaakko Sarela, Ari Visa. COMPARISON OF COMBINED SHAPE DESCRIPTORS FOR IRREGULAR OBJECTS, in the proceedings of the Eighth British Machine Vision Conference, volumen 2 pages 430-439, University of Essex, UK, September 8-11, 1997.

[G] Alfonseca M., Sancho J., Martínez Orga M. TEORIA DE LENGUAJES, GRAMATICA Y AUTOMATAS, Universidad y Cultura, Madrid, 1990.

[H] Levine M.D., VISION IN MAN AND MACHINE, McGraw-Hill, 1985.