TEMA 4 – DESCRIPCIÓN MATEMÁTICA DE OBJETOS

DESCRIPCIÓN E INTRODUCCIÓN AL RECONOCIMIENTO DE OBJETOS:

- **RECONOCIMIENTO DE OBJETOS:**

⇒ **Paso previo => DESCRIPCIÓN DE OBJETOS: representación y caracterización de las regiones de la imagen que previamente han sido aisladas mediante la segmentación.**

→ **Los objetos son descritos mediante un vector de características.**

⇒ **Objetivo: a partir de las características encontradas y de los posibles objetos que el conocimiento a priori del problema se espera que puedan aparecer, el sistema debe determinar el tipo de los objetos que están presentes en la imagen.**

Requisitos de atributos:

- **Poder de discriminación: las características a escoger deben tomar valores significativamente distintos para objetos que pertenecen a clases diferentes.**
- **Fiabilidad: las características han de tomar valores similares para objetos de la misma clase.**
- **Independencia: las diversas características no deben estar correladas unas con otras ya que si no reflejarían la misma propiedad del objeto.**
- **Número de características: debe ser el más pequeño posible ya que la complejidad del sistema aumenta rápidamente con la dimensión del vector de características, esto es, con el número de características utilizadas en la descripción del objeto.**

INTRODUCCIÓN AL RECONOCIMIENTO DE OBJETOS: Ejemplo

➢ **Reconocimiento de un objeto de una imagen como perteneciente a alguna de las siguientes clases u objetos: triángulo equilátero, círculo y cuadrado.**

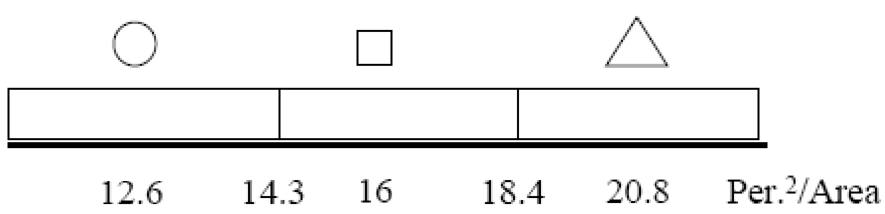
→ **Característica empleada para el reconocimiento: compacticidad (relación perímetro^2 y área).**

→ **Idealmente cada una de las clases están definidas por unos valores concretos de esta característica adimensional:**

	Área	Perímetro	Perímetro^2 / Área
O	πr^2	$2\pi r$	**12.56**
□	l^2	$4l$	**16**
Δ	$\sqrt{3}\,l^2/4$	$3l$	**20.8**

→ **Construir una regla de decisión que clasifique en función del valor medido de dicha característica:**

- **Si per.^{2} / área < 14.3 entonces objeto = círculo.**
- **Si 14.3 < per.^{2} / área < 18.4 entonces objeto = cuadrado**
- **Si 18.4 < per.^{2} / área entonces objeto = triángulo**



DESCRIPCIÓN DE OBJETOS: Planteamiento

➢ **Descripción de objetos: extracción de características de un objeto para su reconocimiento.**

⇒ **La imagen debe ser previamente segmentada (los objetos se aíslan del resto de la imagen).**

⇒ **Descripción matemática del/os objeto/s presentes en la imagen: obtención de un conjunto de valores numéricos que cuantifiquen determinadas propiedades del objeto (Vector de Características).**

➢ **Requisitos de la descripción:**

⇒ **Representatividad: objetos que se deseen clasificar como distintos deben tener distintas características (cada objeto debe tener una descripción única).**

⇒ **Invarianza frente a transformaciones geométricas (escalado, traslación, rotación): los descriptores deben ser independientes del tamaño, la localización y orientación del objeto.**

⇒ **Sensible: deben reflejar diferencias entre objetos similares.**

➢ **Tipos de descriptores:**

⇒ **Externos (descriptores de contorno): hacen referencia a características externas de la región (su contorno) – Ejemplos: códigos de cadena, aproximación poligonal, representación polar, descriptores de Fourier, esqueletización, etc…**

⇒ **Internos (descriptores de región): hacen referencia a los píxeles que forman la región en sí (su interior). Ejemplos: momentos, descriptores topológicos, textura, etc…**

❖ **Descriptores de similitud: dan una medida del parecido de regiones de la imagen con un patrón determinado. Ejemplo: correlación bidimensional normalizada.**

➢ **Algunos descriptores básicos:**

⇒ **Para reconocimiento de objetos:**

→ **Tamaño o Área (A): número de píxeles comprendidos en el contorno del objeto.**

❖ **Si f(x,y) es una imagen binaria donde los píxeles del objeto tienen todos valor "1" y los del fondo valor "0":**

$$A = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)$$

→ **Longitud o Perímetro (P): número de píxeles exteriores al objeto.**

❖ **Estas 2 medidas dependen del tamaño del objeto. Para disponer de una medida invariante al tamaño:**

→ **Compacticidad: P^2 / A (*circularidad* – su valor mínimo corresponde al círculo con un valor 4π).**

→ **Excentricidad: cociente de las longitudes de los ejes mayor y menor de un objeto (definidos en términos de su frontera).**

⇒ **Para localizar al objeto en la imagen:**

→ **Centroide o centro de gravedad: se determina promediando las coordenadas *x* e *y* de los píxeles que conforman el contorno.**

$$\overline{x} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x f(x,y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)} \quad ; \quad \overline{y} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} y f(x,y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)}$$

→ **Dirección o eje principal: autovector asociado al mayor autovalor de la matriz de dispersión de los píxeles (distancias al cuadrado de los puntos sobre el centroide).**

Representación y descripción del contorno: códigos de cadena

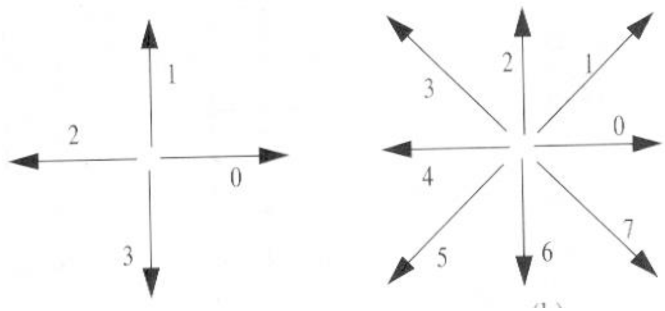
➢ **Los códigos de cadena se usan para representar la frontera de un objeto en base a un conjunto de segmentos, de una longitud y dirección específica, conectados entre sí.**

➢ **Procedimiento básico de obtención:**

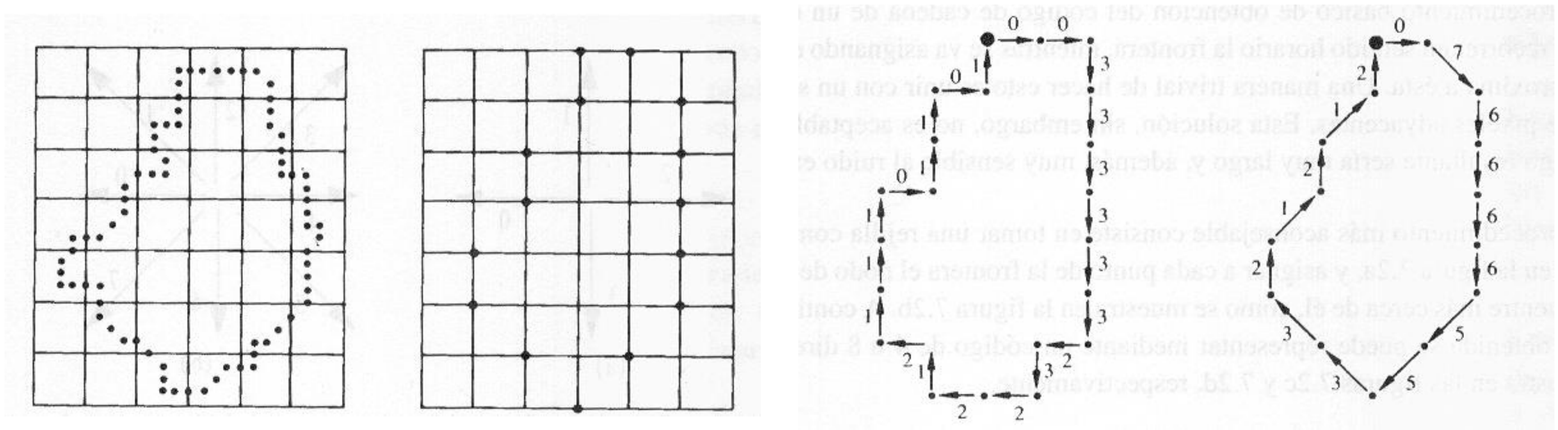
⇒ **Recorrer en sentido horario la frontera (serie de 1's conectados sobre 0's – imagen binaria).**

⇒ **Codificar la dirección de avance para llegar al vecino:**

→ **Los códigos de cadena más usados son de 4 y 8 direcciones codificadas como indica la figura:** ⟶

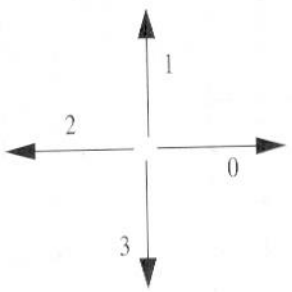


⇒ **Tomar una rejilla, asignar a cada punto frontera el nodo de la rejilla que se encuentre más cerca de él, y codificar la dirección de avance:**



Frontera digital muestreada con una rejilla superpuesta. Puntos de la nueva frontera. Código cadena de 4 y 8 direcciones.

Códigos de cadena: invarianza a rotación.



➢**Para conseguir una representación invariante a la orientación del objeto en la imagen se utiliza la diferencia de las direcciones en lugar del propio código:**

→ **Se calcula contando, en sentido antihorario, el número de direcciones que separan dos vectores adyacentes. Ejemplo: la primera diferencia del código de 4 direcciones 10103322 es 3133030.**

→ **Este procedimiento no consigue la invarianza plena ante cualquier orientación, sino sólo ante orientaciones que coinciden con las consideradas por el propio código de cadena, es decir, 90º o 45º, para los códigos de 4 y 8 direcciones respectivamente.**

➢ **Para conseguir una representación del contorno realmente invariante ante rotación, es necesario que el código de cadena se oriente según alguna dirección intrínseca del objeto, por ejemplo, la dirección principal asociada a la máxima dispersión de los píxeles del contorno (eje de mayor envergadura):**

⇒ **Obtener el rectángulo mínimo que circunscribe al objeto.**

⇒ **Se toman las direcciones del código de cadena según los ejes de dicho rectángulo.**

⇒ **Determinar el tamaño de la rejilla, dependiendo de la precisión que se desee.**

→ **Para conseguir invarianza ante el tamaño, el rectángulo se debe dividir siempre en el mismo número de celdas, independientemente del tamaño del objeto.**

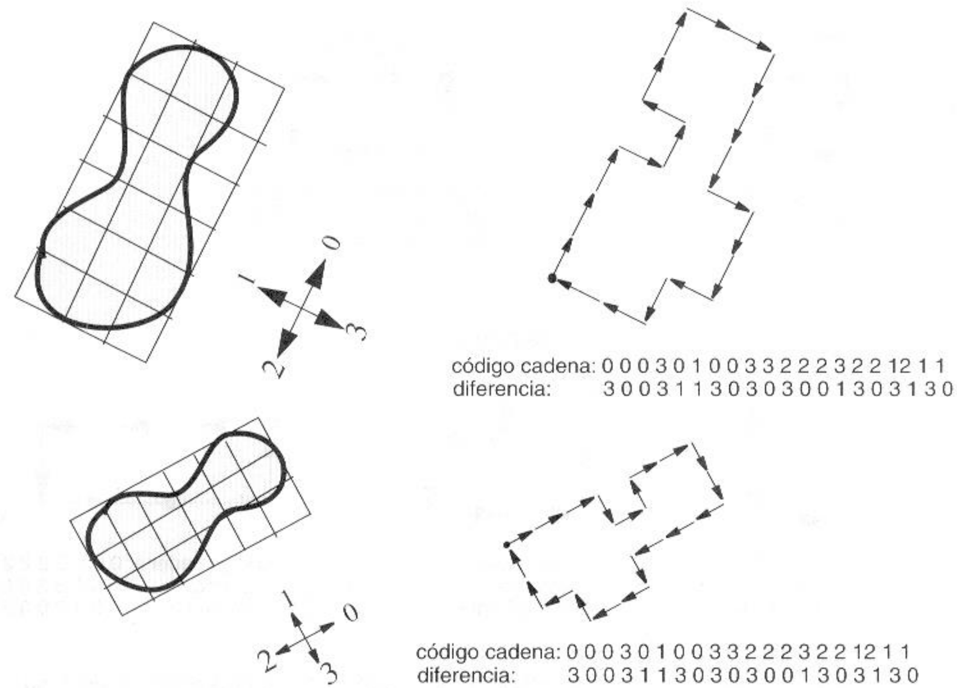
⇒ **Obtener el código de cadena y su primera diferencia.**

<u>Códigos de cadena: invarianza a rotación.</u>

Ejemplo:

⇒ Obtención del código de cadena para un mismo objeto con orientación y tamaño diferentes.

→ Los códigos obtenidos son similares: representación invariante ante orientación y tamaño.



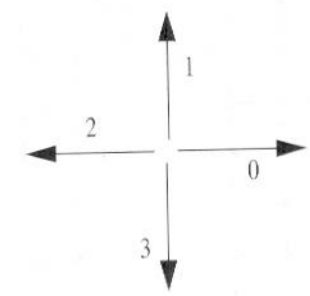
<u>Códigos de cadena: problema</u>

⇒ Depende del punto de comienzo

⇒ Solución: se rota el punto de comienzo de la cadena hasta conseguir el menor entero.

→ Obtener el código de cadena comenzando por un punto arbitrario.

→ Tratar dicho código como una secuencia circular de números de dirección y redefinir el punto de comienzo tal que el número entero resultante sea el menor posible.



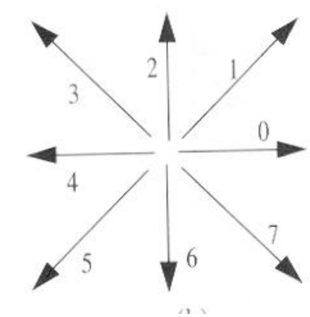
Códigos de cadena: ventajas

⇒ **Es invariante a traslación, y se puede conseguir invarianza a escalado y a rotación.**

⇒ **Buena compresión de la información:**

→ **2 bits por dirección en códigos de 4-direcciones.**

→ **3 bits por dirección en códigos de 8-direcciones.**



⇒ **A partir de este código de cadena es inmediata la descripción del objeto mediante características relacionadas con su forma: perímetro, curvatura, área, longitud máxima, etc…**

- **Perímetro:**

→ **códigos de 4-direcciones: P = longitud del código.**

→ **códigos de 8-direcciones: P = número de códigos pares + √2 número códigos impares.**

- **Ancho:**

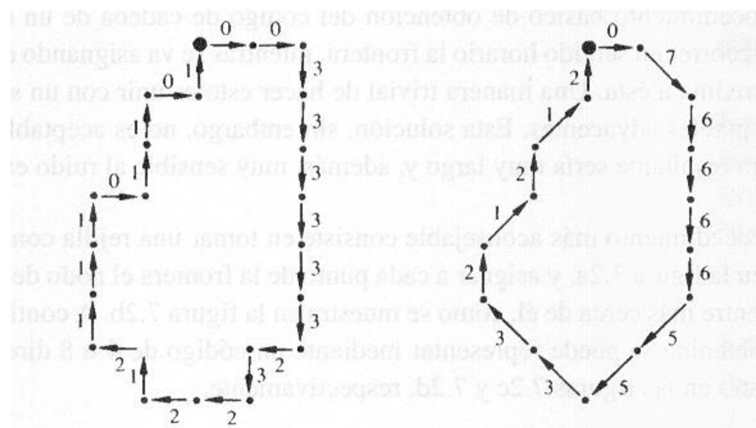
→ **códigos de 4-direcciones: A= nº de {0} en el código.**

→ **códigos de 8-direcciones: A= nº de {0, 1, 7} en el código.**

- **Alto:**

→ **códigos de 4-direcciones: A= nº de {1} en el código.**

→ **códigos de 8-direcciones: A= nº de {1,2,3} en el código.**



Representación y descripción del contorno: Aproximación poligonal

➢ **Aproximar el contorno mediante el uso de un polígono:**

$\Rightarrow$ **Cuanto mayor sea el número de segmentos (lados) del polígono utilizado más exacta será la aproximación.**

$\Rightarrow$ **El objetivo consiste en emplear el menor número posible de segmentos de tal manera que no se pierdan las características esenciales de la forma del contorno.**

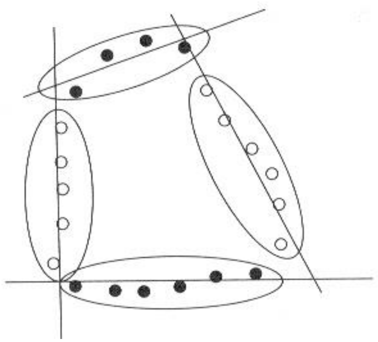
$\Rightarrow$ **Algoritmos iterativos complejos.**

$\Rightarrow$ **Métodos más empleados: *método de fusión y de división recursiva.***

Aproximación poligonal: técnica de fusión

➢ **Se van ajustando los puntos del contorno mediante una recta hasta que el error cometido supere un umbral preestablecido.**

➢ **Se almacenan los parámetros de la recta obtenida y se repite el proceso con los puntos siguientes del contorno hasta que se hayan tratado todos.**



➢ **Las intersecciones de las rectas adyacentes son los puntos del polígono que se aproxima al contorno.**

➢ **Problema al aproximar esquinas: los vértices del polígono no se suelen corresponder con las inflexiones propias del contorno.**

Aproximación poligonal: método de división recursiva

- **Consiste en dividir sucesivamente un segmento en dos hasta que se satisfaga un determinado criterio.**
- **Procedimiento:**

  ⇒ Elegir puntos de comienzo: en contornos cerrados, son los puntos de la frontera más lejanos entre sí.

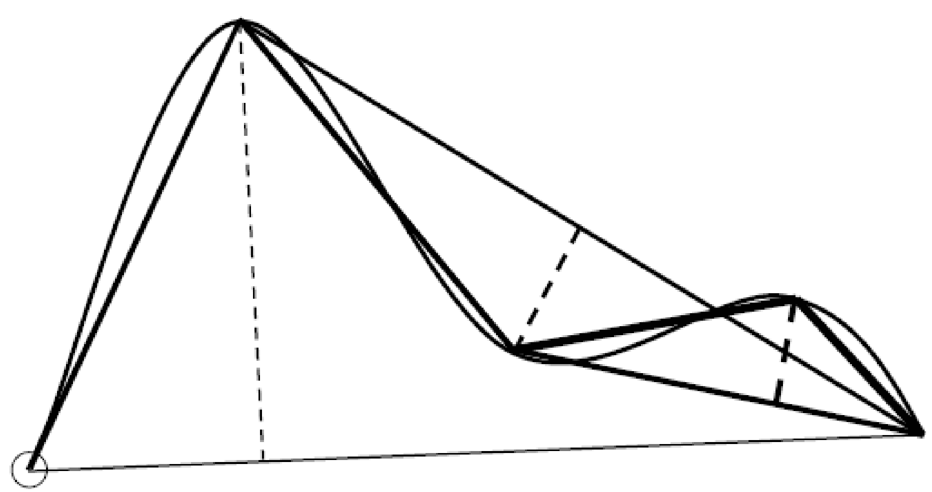
  ⇒ Unir ambos puntos con un segmento.

  ⇒ Obtener el punto más alejado del contorno al segmento.

  ⇒ Comparar con un umbral de error seleccionado en base a algún criterio: por ejemplo, que la distancia del punto al segmento no sobrepase un umbral determinado

  → En caso de que lo sobrepase, el punto más lejano es considerado un nuevo vértice del polígono por donde se subdivide el segmento en dos.

  → Si no lo supera, el segmento es una buena aproximación a la curva.



Sucesivos pasos en el proceso de ajuste poligonal de un contorno mediante el método de división recursiva

Representación y descripción del contorno: representación polar

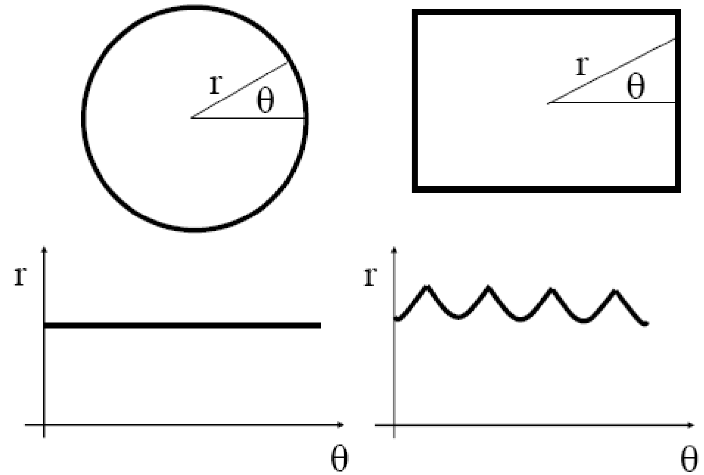
- **Representación polar (*signatura*):**

  ⇒ **Representación funcional de una frontera.**

  ⇒ **Procedimiento más habitual:**

  → **Obtener la curva que represente la distancia al centroide del objeto de todos los puntos de su borde en función del ángulo.**

  → **Ejemplo: representación polar del contorno de un círculo y un cuadrado:**



Representación polar: características

- **Representación invariante a la posición del objeto en la imagen.**
- **Representación dependiente del tamaño.**

  ⇒ **La invarianza al tamaño se consigue dividiendo la función por la distancia máxima al centroide, de forma que la distancia máxima resultante sea uno.**

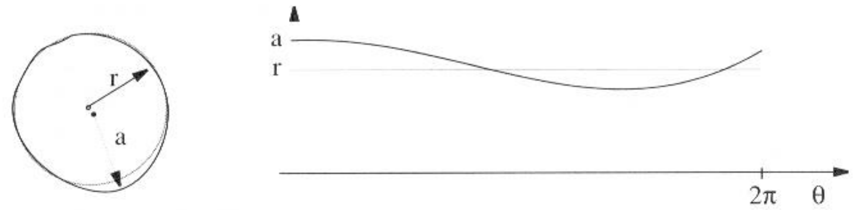
- **Representación dependiente del ángulo de comienzo (depende del punto de comienzo, punto por donde se empieza a describir la frontera).**

  ⇒ **La invarianza ante el ángulo de comienzo se logra comenzando la representación por el ángulo para el cual la distancia sea máxima.**

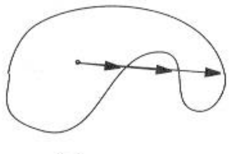
Representación polar: características

➢ **Inconvenientes:**

⇒ **Método muy sensible a la posición del centroide: cualquier error o perturbación en el contorno se refleja en cálculo del centroide, repercutiendo en la representación polar del mismo.**



⇒ **La presencia de concavidades en el contorno pueden dar lugar a una representación polar multivaluada (varias distancias para un mismo ángulo) para algunos ángulos.**



Representación y descripción del contorno: descriptores de Fourier

➢ **Procedimiento:**

⇒ **Para un borde formado por N puntos en el plano X-Y, se toma un punto al azar *(x_0,y_0)* como primer elemento de una sucesión de puntos, (x_0,y_0) , (x_1,y_1) , (x_2,y_2) , … , (x_{N-1},y_{N-1}) tomados en sentido antihorario: borde → secuencia s(k) = [x(k)=x_k , y(k)=y_k] con k=0,1,…,N-1.**

⇒ **Estas parejas de puntos se transforman a números complejos *s(k) = x(k)+jy(k)*.**

⇒ **Se obtiene la transformada discreta de Fourier (TDF) de *s(k):***

$$\text{Descriptores de Fourier} \equiv a(u) = F\left[s(k)\right] = \frac{1}{N}\sum_{k=0}^{N-1} s(k)\exp\left[-\frac{2\pi juk}{N}\right] \quad u = 0,1,2,\ldots,N-1$$

⇒ **La transformada inversa de Fourier de los *a(u)* restaura *s(k):***

$$s(k) = F^{-1}\left[a(u)\right] = \sum_{u=0}^{N-1} a(u)\exp\left[\frac{2\pi juk}{N}\right] \quad k = 0,1,2,\ldots,N-1$$

Descriptores de Fourier: características

➢ **Permiten reconstruir el borde, incluso anulando algunos *a(u)*: utilizando los M primeros descriptores { a(u)=0 para u>M-1 }:**

$$\hat{s}(k) = F^{-1}\left[a(u)\right] = \sum_{u=0}^{M-1} a(u)\exp\left[\frac{2\pi uk}{N}\right] \quad k = 0,1,2,\ldots,N-1$$

⇒ **Los descriptores de Fourier están ordenados por frecuencias, lo que permite fácilmente reducir el número de éstos a consta de perder la componentes asociadas a las más altas, y que generalmente no aportan información básica del contorno.**

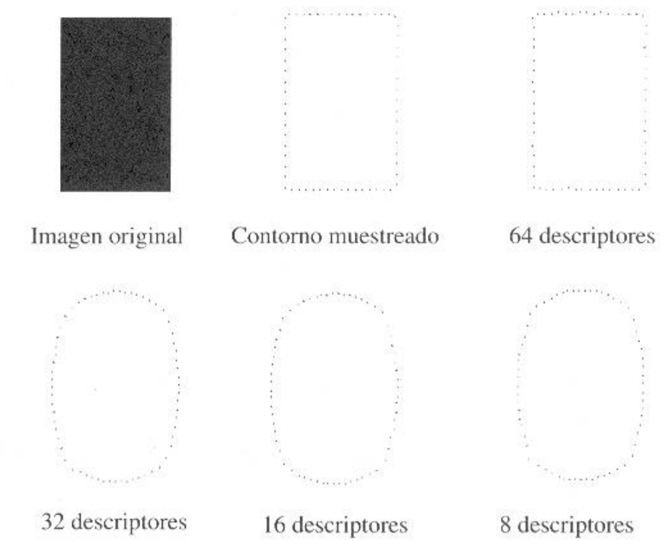
⇒ **Cuanto más pequeño es M, mayor es la pérdida de los detalles en la reconstrucción .**

⇒ **Ejemplo: sea una forma geométrica con N = 64 puntos:**

→ **La reconstrucción total se logra con M = 64.**

→ **Se requiere un elevado número de términos para definir los detalles (esquinas, bordes rectos).**

→ **Unos pocos descriptores son capaces de capturar el borde del objeto de una forma aproximada y se pueden utilizar para obtener cierta información sobre su forma.**



⇒ **En principio, entre 10 y 15 descriptores son suficientes para definir cualquier forma.**

Descriptores de Fourier: ejemplo

Descriptores de Fourier:

a) imagen original

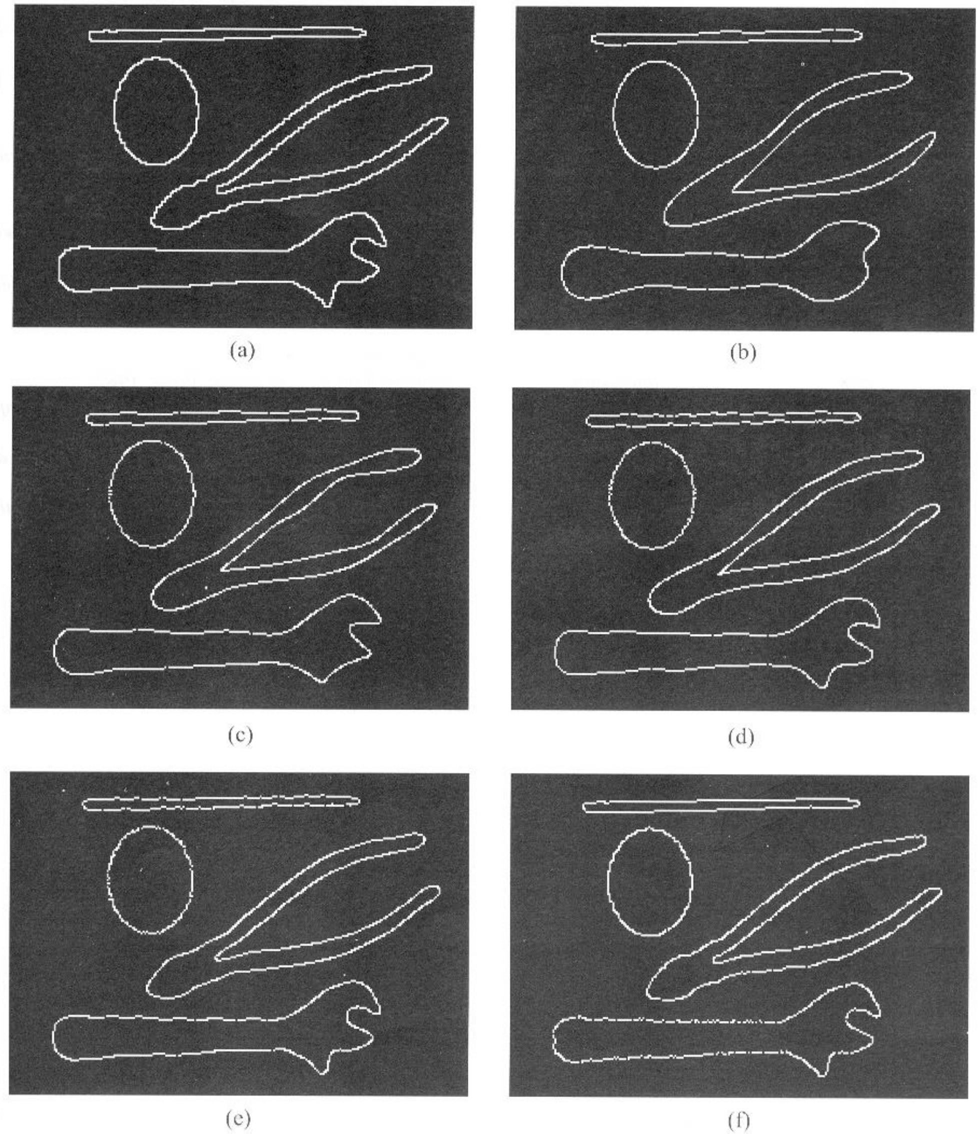
b) resultado de usar 10 descriptores de Fourier

c) 20 descriptores

d) 30

e) 50

f) 100



(a) (b) (c) (d) (e) (f)

Descriptores de Fourier: propiedades

- **Invarianza ante traslaciones, rotaciones, cambios de escala y al punto de comienzo de la secuencia *s(k)*.**

⇒ Los descriptores de Fourier no son directamente invariantes a esos cambios geométricos, pero dichos cambios se pueden relacionar con simples transformaciones:

Para una secuencia de borde *s(k)*, cuyos descriptores de Fourier son *a(u)*:

→ Rotación: la rotación afecta a todos los coeficientes por igual, por un término constante multiplicativo $e^{j\theta}$:

$$s_r(k) = s(k)e^{j\theta} \Rightarrow a_r(u) = a(u)e^{j\theta}$$

→ Traslación: no tiene efecto en los descripores, excepto para el coeficiente de orden 0, que se añade un factor.

→ Escalado: los coeficientes aparecen multiplicados por el factor de escala:

$$s_e(k) = \alpha s(k) \Rightarrow a_e(u) = \alpha a(u)$$

→ Cambio en el punto de comienzo:

$$s_p(k) = s(k-k_0) \Rightarrow a_p(u) = a(u)\exp\left(-j\frac{2\pi k_o u}{N}\right)$$

donde $s(k-k_0)$ es una redefinición de la secuencia s(k) donde se cambia el punto de comienzo a $k=k_0$ desde k=0.

Representación y descripción del contorno: Esqueletización

➢ **Idea básica: conservar toda la información sobre la forma y estructura del contorno del objeto, eliminando redundancias que no aportan nada a la hora de reconocerlo.**

➢ **Esqueleto de una región:**

⇒ **Se define mediante la transformación del eje medio o *MAT* (*Medial Axis Transformation*):**

→ **El objeto se divide en dos conjuntos: R, puntos del interior, y C, puntos del contorno.**

→ **Para cada punto de R se busca el punto de C más cercano (concepto cercano establecido, por ejemplo, a partir de la distancia euclídea):**

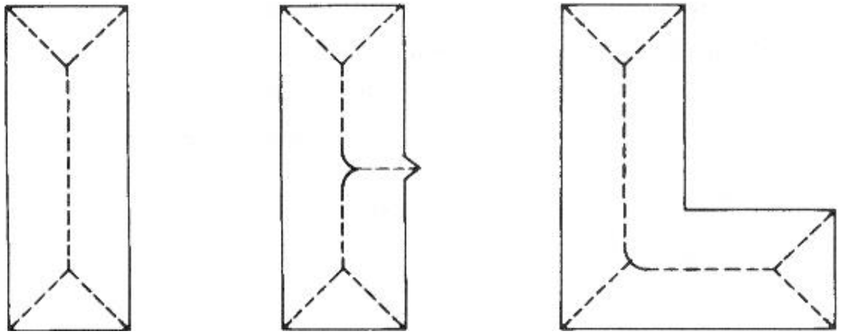
❖ **Si hay más de un punto de C que se encuentre a esa distancia mínima, el punto de R pertenecerá al MAT que se equipará al esqueleto.**

❖ **Si hay un único punto de C a distancia mínima, el punto de R no pertenece al esqueleto.**

⇒ **Inconveniente: implementación prohibitiva debido al gran número de operaciones que hay que realizar para cada punto.**

⇒ **Solución: se recurre a algoritmos que determinen el esqueleto de una región de forma más rápida:**

→ ***Mediante técnicas de adelgazamiento.***



Esqueletos de 3 regiones simples obtenidos mediante la transformación del eje medio

Esqueletización mediante técnicas de adelgazamiento

➢ **Algoritmos de adelgazamiento: borran iterativamente píxeles del exterior sin llegar a eliminar aquellos terminales, ni que se produzca desconexión de la región.**

➢ **Algoritmo de Zhang y Suen:**

⇒ **Supone que los píxeles del objeto están marcados con valor “1” mientras que el resto lo están con “0”.**

⇒ **Los píxeles del contorno son aquellos etiquetados como “1’s” que tienen al menos 1 de los 8 vecinos a “0”.**

⇒ **Procedimiento basado en dos etapas:**

→ **Etapa 1: un píxel del contorno p_0 se marca para borrado si cumple:**

1. 2 ≤ Número de vecinos de p_0 no nulos $N(p_0) \leq 6$ (para que los puntos finales se preserven).

2. Sólamente hay una transición de “0” a “1”, $S(p_0)$, en la secuencia ordenada $p_1, p_2, \ldots, p_8$ (se preservan los puntos que se encuentran en los extremos).

3. Alguno de p_1, p_3, y p_5 es cero.

4. Alguno de p_3, p_5, y p_7 es cero.

$$\begin{bmatrix} p_8 & p_1 & p_2 \\ p_7 & p_0 & p_3 \\ p_6 & p_5 & p_4 \end{bmatrix}$$

Notación utilizada

→ **Etapa 2: se marcan para eliminar también los píxeles del contorno que verifican:**

1. y 2. Las mismas dos primeras condiciones de la etapa 1.

3.Alguno de p_1, p_3, y p_7 es cero.

4. Alguno de p_1, p_5, y p_7 es cero.

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & p_0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$N(p_0) = 4 \; ; \; S(p_0) = 3$$

Esqueletización mediante técnicas de adelgazamiento: pseudocódigo algoritmo de Zhang y Suen

⇒ **Para todos los píxeles de contorno:**

→ **Marcarlo como candidato a ser borrado si verifica las cuatro condiciones de la etapa 1.**

⇒ **Una vez recorrido todos los puntos:**

→ **Borrar (poner a “0”) todos los píxeles marcados en la etapa 1.**

⇒ **Para todos los píxeles de contorno:**

→ **Marcarlo como candidato a ser borrado si verifica las cuatro condiciones de la etapa 2.**

⇒ **Una vez recorrido todos los puntos:**

→ **Borrar (poner a “0”) todos los píxeles marcados en la etapa 2.**

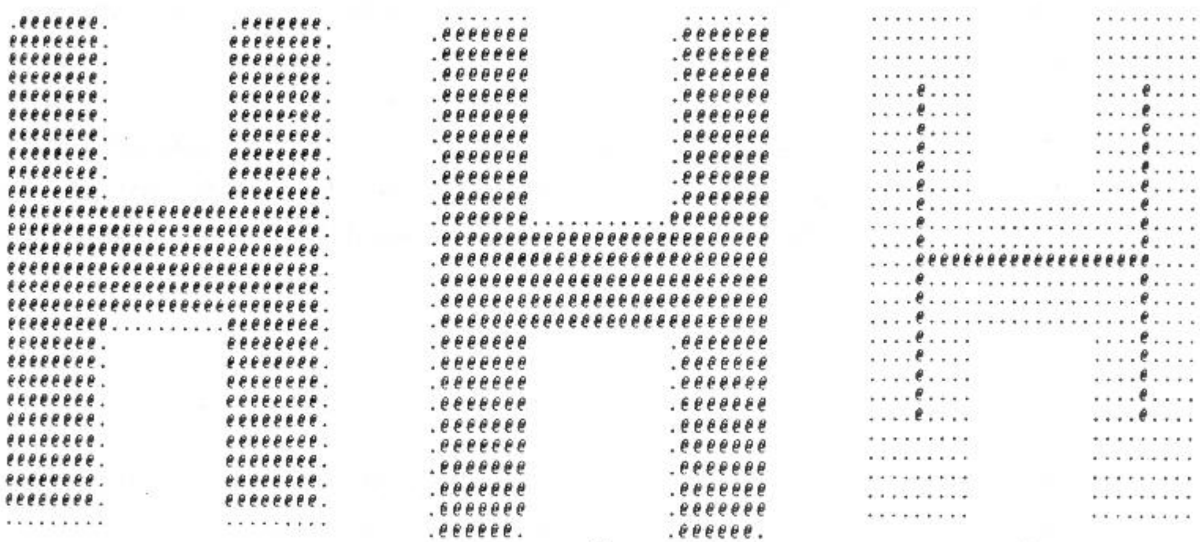
⇒ **Volver al primer paso mientras se sigan marcando píxeles.**

Ejemplo:

a) Resultado del borrado tras la etapa 1 del algoritmo de adelgazamiento.

b) Resultado del borrado tras la etapa 2.

c) Resultado final.



(a) (b) (c)

4.1.- Introducción y planteamiento

4.2.- Representación de objetos: descriptores de contorno

4.2.1.- Códigos de cadena

4.2.2.- Aproximación poligonal

4.2.3.- Representación polar

4.2.4.- Descriptores de Fourier

4.2.5.- Esqueletización

4.3.- Representación de objetos: descriptores de región

4.3.1.- Momentos

4.3.2.- Descriptores topológicos

4.3.3.- Textura

4.4.- Descripción de similitud mediante correlación

Descriptores de región:

- **Descriptores que utilizan información aportada por todos los píxeles del objeto, no sólo los del contorno.**
- **Tipos: momentos, descriptores topológicos, textura.**

Descriptores de región: Momentos:

- **Objetivo:**

⇒ Obtención de momentos invariantes a la rotación, la escala y la traslación.

⇒ Con estos descriptores se pretende poder reconocer los objetos aunque no se encuentren siempre en la misma posición, estén girados o su tamaño sea distinto.

- **Procedimiento de obtención:**

1.- Momento ordinario de orden (p+q): $m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad p,q = 0,1,2,\ldots$

donde el sumatorio se toma sobre todas las coordenadas espaciales (x,y) de puntos de la región.

⇒ Para conseguir la invarianza a la traslación o posición del objeto, se definen los *momentos centrales* restando a las coordenadas de cada punto las del centro de gravedad:

2.- Momento central de orden (p+q): $\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \;,\; \bar{x} = \frac{m_{10}}{m_{00}} \;;\; \bar{y} = \frac{m_{01}}{m_{00}}$

Los momentos centrales se pueden poner en función de los no centrales. Por ejemplo, hasta los de orden 2:

$$\mu_{00} = m_{00} \;;\; \mu_{10} = \mu_{01} = 0 \;;\; \mu_{11} = m_{11} - \frac{m_{10}m_{01}}{m_{00}} \;;\; \mu_{20} = m_{20} - \frac{m_{10}^{\;2}}{m_{00}} ;\; \mu_{02} = m_{02} - \frac{m_{01}^{\;2}}{m_{00}}$$

***Procedimiento de obtención momentos invariantes:*.**

⇒ **Para conseguir además, invarianza ante escalados se definen los *momentos centrales normalizados*:**

2.- Momento central de orden (p+q): $\eta_{pq} = \dfrac{\mu_{pq}}{\left(\mu_{00}\right)^{\gamma}} \;\; ; \;\; \gamma = \dfrac{p+q}{2} + 1 \;\; ; \;\; \left(p+q\right) = 2, 3, \ldots$

⇒ **De los momentos centrales de 2 y 3 orden, se deriva un conjunto de 7 momentos invariantes a traslaciones, rotaciones y cambios de escala:**

3.- Momentos invariantes de Hu:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = \left(\eta_{20} - \eta_{02}\right)^2 + 4\eta_{11}^2$$

$$\phi_3 = \left(\eta_{30} - 3\eta_{12}\right)^2 + \left(3\eta_{21} - \eta_{03}\right)^2$$

$$\phi_4 = \left(\eta_{30} + \eta_{12}\right)^2 + \left(\eta_{21} + \eta_{03}\right)^2$$

$$\phi_5 = \left(\eta_{30} - 3\eta_{12}\right)\left(\eta_{30} + \eta_{12}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - 3\left(\eta_{21} + \eta_{03}\right)^2\right] + \left(3\eta_{21} - \eta_{03}\right)\left(\eta_{21} + \eta_{03}\right)\left[3\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right]$$

$$\phi_6 = \left(\eta_{20} - \eta_{02}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] + 4\eta_{11}\left(\eta_{30} + \eta_{12}\right)\left(\eta_{21} + \eta_{03}\right)$$

$$\phi_7 = \left(3\eta_{21} - \eta_{30}\right)\left(\eta_{30} + \eta_{12}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - 3\left(\eta_{21} + \eta_{03}\right)^2\right] + \left(3\eta_{21} - \eta_{03}\right)\left(\eta_{21} + \eta_{03}\right)\left[3\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right]$$

⇒ **Para caracterizar un objeto no siempre será necesario obtener estos 7 descriptores, dependiendo del problema de reconocimiento puede bastar un número menor de ellos.**

Descriptores de región: descriptores topológicos

➢ **Ofrecen una descripción global de las regiones de una imagen, es decir, propiedades que no se ven afectadas por las deformaciones.**

➢ **No tratan de dar un número exacto sino indicar alguna idea sobre la forma del objeto.**

➢ **Ejemplo de descriptores:**

$\Rightarrow$ **Número de *huecos* del objeto.**

$\Rightarrow$ **Número de *componentes conectados*:**

$\rightarrow$ **Aquellos elementos separados que forman parte del objeto.**

$\rightarrow$ **Estos elementos se pueden tratar como subregiones tales que cualquier par de puntos de la misma se pueden interconectar mediante una curva totalmente incluida en ella.**

$\Rightarrow$ **Número de *Euler*: diferencia entre el número de regiones conectadas y el número de huecos.**

A	**B**	**i**
Número de componentes: 1	**Número de componentes: 1**	**Número de componentes: 2**
Número de huecos: 1	**Número de huecos: 2**	**Número de huecos: 0**
Número de Euler: 0	**Número de Euler: -1**	**Número de Euler: 2**

Descriptores de región: textura

- **“Medida de la disposición espacial de los niveles de grises de los píxeles de una imagen” (IEEE Standard, 1990).**

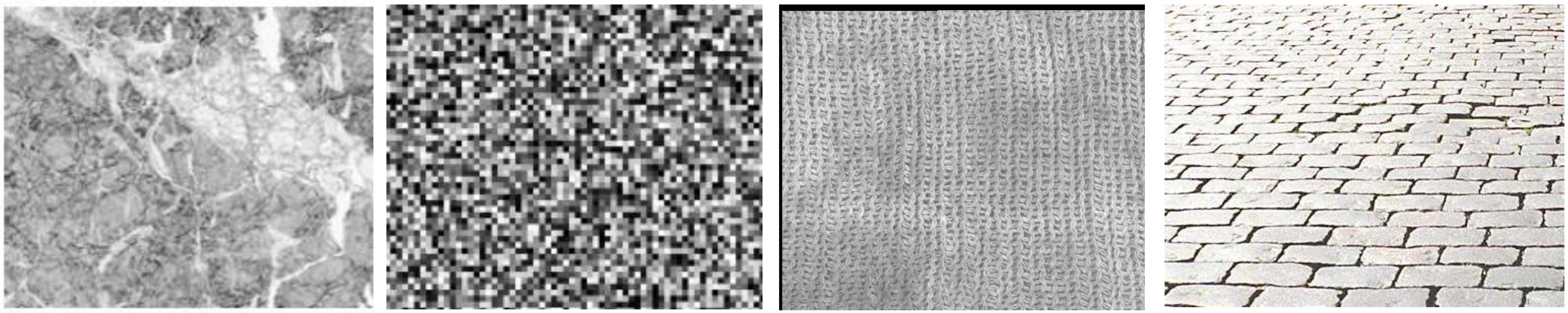
- **Característica principal: repetición en el espacio de un patrón básico de niveles de gris.**

- **Un descriptor de textura debe cuantificar propiedades como *suavidad, rugosidad y regularidad*.**

- **Importancia:**

  ⇒ Forma parte de las características que definen los objetos: es un descriptor muy utilizado en aplicaciones de reconocimiento de objetos (ej. en imágenes aéreas y de satélite, en imágenes de microscopio – tejidos).

  ⇒ Al observar su deformación se puede obtener información tridimensional del objeto (fundamentalmente se utiliza para determinar la forma y orientación de una superficie).



Distintos ejemplos de texturas

Textura: métodos fundamentales de descripción

➢ ***Estadístico:*** **basados en el análisis de valores estadísticos de la distribución de los niveles de gris en una región.**

➢ **Espectral: basado en la transformada de Fourier (la textura es una repetición espacial de un patrón, por lo que aparecerán valores altos o picos en su T.F. a la frecuencia de repetición).**

➢ **Estructural: basado en la localización y descripción de primitivas (pequeños objetos simples que se repiten a lo largo de la región a describir).**

Textura: enfoque estadístico

➢ **Caracterizar la distribución espacial de los niveles de gris de una región mediante los momentos de su histograma:**

$\Rightarrow$ **Si *i* es una variable aleatoria que representa la intensidad de un píxel genérico de la región a describir (*i =1,…L*, para *L* niveles de gris considerados) y *p(i)* es su histograma correspondiente:**

$\rightarrow$ **Momento central n-ésimo de *i* respecto de la media *m*:**

$$\mu_n = \sum_{i=1}^{L} (i-m)^n p(i)$$

donde $m = \sum_{i=1}^{L} i p(i)$ es el valor medio de intensidad i de región tratada (estimación del nivel de gris de la textura)

$$\left\{ p(i) = \frac{\text{Número de píxeles en el nivel de intensidad i}}{\text{Número de píxeles de la imagen}} \;\; ; \;\; p(i) \le 1 \;\; ; \;\; \sum_{i=1}^{L} p(i) = 1 \;\; (\, L \equiv \text{nº total de niveles de gris}\,) \right\}$$

Textura: enfoque estadístico

⇒ **Momentos de orden cero y de primer orden:** $\mu_0 = 1 \; ; \; \mu_1 = 0$

⇒ **Momento de segundo orden o varianza:** $\mu_2 = \sigma^2 = \sum_{i=1}^{L} (i - m)^2 \, p(i)$

→ **Indica la media de la dispersión de los niveles de gris respecto al valor medio**

→ **Da una medida del contraste del objeto y puede ser usado para establecer descriptores de suavidad relativa:**

$$\text{Ejemplo}: R = 1 - \frac{1}{1+\sigma^2} \Rightarrow \begin{cases} 0 \;\; \text{para áreas de intensidad constante} \\ 1 \;\; \text{para valores grandes de } \sigma^2 \end{cases}$$

⇒ **Momento de tercer orden:**

→ **Da una medida del la asimetría del histograma:**

❖ **Histograma simétrico (ej. Distribución gaussiana) → μ_3 = 0 ; según el histograma esté más inclinado a un lado o a otro sus valores serán negativos o positivos.**

⇒ **Momento de cuarto orden:**

→ **Da una medida de la uniformidad del histograma (cuánto de plano es el histograma)**

- **Ventaja: son de fácil obtención (coste computacional bajo).**
- **Inconveniente:**

⇒ **Al estar basados en el histograma, no captan toda la información espacial (no se tiene en cuenta información sobre la posición relativa de cada píxel respecto de los demás)**

⇒ **Solución: estadísticos de segundo orden – matrices de coocurrencia de niveles de gris.**

4.1.- Introducción y planteamiento

4.2.- Representación de objetos: descriptores de contorno

4.2.1.- Códigos de cadena

4.2.2.- Aproximación poligonal

4.2.3.- Representación polar

4.2.4.- Descriptores de Fourier

4.2.5.- Esqueletización

4.3.- Representación de objetos: descriptores de región

4.3.1.- Momentos

4.3.2.- Descriptores topológicos

4.3.3.- Textura

4.4.- Descripción de similitud mediante correlación

Descripción de similitud mediante correlación

➢ **Objetivo: obtener una medida de similitud entre los objetos de la imagen y un modelo o patrón conocido que se emplea de referencia.**

➢ **Correlación:**

Correlación de dos funciones continuas unidimensionales: $f(x)\bullet g(x)=\int_{-\infty}^{\infty} f(\alpha)g(x+\alpha)d\alpha$

(Valor máximo de la función en el punto de máximo solapamiento de ambas señales)

Correlación de 2 imágenes digitales $f(x,y)$, tamaño MxN, y $w(x,y)$, tamaño JxK:

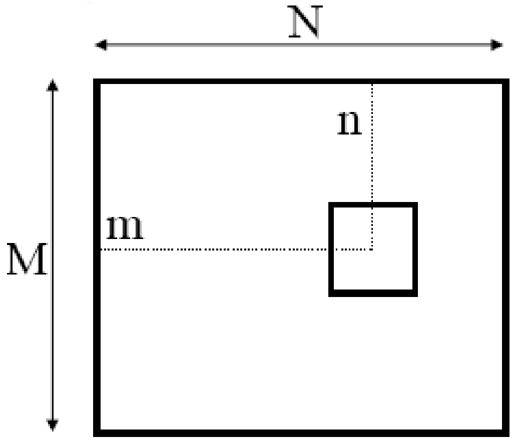
$$R(m,n)=\sum_{\alpha=-J/2}^{J/2}\sum_{\beta=-K/2}^{k/2} f(m+\alpha,n+\beta)w(\alpha,\beta)$$

donde $m=0,1,\ldots,M-1$ y $n=0,1,\ldots,N-1$, y el doble sumatorio se extiende en la región donde w y f se solapan.

⇒ **Normalmente: *f(x,y)* = imagen completa ; *w(x,y)* = imagen patrón.**

⇒ **Al ir variando *m* y *n*, *w(x,y)* se va desplazando sobre la imagen *f*, obteniéndose los valores de *R*.**

⇒ **En cada píxel *(m,n)*, la matriz *R* indica el grado de solapamiento o parecido entre *w(x,y)* y el área de imagen de *f(x,y)* en el punto *(m,n)*.**



Descripción de similitud mediante correlación

- **Problema:** **la correlación así definida es dependiente de los cambios de escala en las funciones *f* y *w*.**
- **Solución:** **correlación normalizada**

$$r(m,n)=\frac{\sum_x \sum_y \left[f(x,y)-\hat{f}(x,y)\right]\left[w(m+x,n+y)-\hat{w}\right]}{\sqrt{\sum_x \sum_y \left[f(x,y)-\hat{f}(x,y)\right]^2 \sum_x \sum_y \left[w(m+x,n+y)-\hat{w}\right]^2}}$$

donde:

$\Rightarrow \hat{w}$: media de las intensidades de la imagen $w(x,y)$ que actúa como patrón (es constante).

$\Rightarrow \hat{f}(x,y)$: media de las intendides de los píxeles de la imagen $f(x,y)$ en la región coincidente con $w(x,y)$.

$\Rightarrow$ Los sumatorios han de tomarse sobre los píxeles en los que f(x,y) y w(x,y) coinciden.

$\Rightarrow$ **Los valores de *r(m,n)* son reales comprendidos en el intervalo [-1,1], independientemente de los cambios de escala en la amplitud de *f(x,y)* y *w(x,y)*.**

$\Rightarrow$ **Para representar la matriz *r(m,n)* los valores de -1 a 1 para cada píxel se transforman linealmente al intervalo de niveles de grises (números enteros) entre 0 y 255, de tal forma que los puntos más claros de la imagen resultado representan las zonas de la imagen donde ha habido mayor semejanza.**

BLOQUE II ASIGNATURA: PARTE PRÁCTICA
TEMAS 4-5 DESCRIPCIÓN Y RECONOCIMIENTO DE OBJETOS

DESCRIPTORES MATEMÁTICOS

Descriptores o características a analizar:

- **Compacticidad, Excentricidad, Extensión, Solidez, Número de Euler.** Estos descriptores se calcularán utilizando la función de matlab `regionprops` (ver definiciones en ayuda matlab).

- **Descriptores de Fourier:** para su cálculo, se debe utilizar la siguiente función:

```
FD = Funcion_Calcula_DF(Matriz_Binaria,NumDF)
```

donde:

- `Matriz_Binaria`: matriz binaria, tipo lógica, con dos posibles valores, *0*'s, píxeles que no son del objeto que se quiere caracterizar y *1*'s, píxeles del objeto a caracterizar.
- `FD`: Vector columna con los valores de los `NumDF` primeros descriptores de Fourier del objeto.

- **Momentos invariantes de Hu:** para su cálculo se debe implementar una función que implemente los 4 primeros momentos de Hu (ver Documentación Anexo I). Los resultados de esta función deben comprobarse con los que obtiene la siguiente función facilitada:

```
Hu = Funcion_Calcula_Hu (Matriz_Binaria)
```

donde:

- `Matriz_Binaria`: matriz binaria con dos posibles valores, *0*'s, píxeles que no son del objeto que se quiere caracterizar y *1*'s, píxeles del objeto a caracterizar.
- `Hu`: Vector columna con los valores de los 7 momento invariantes de Hu del objeto.

- **Extensión Invariante a Rotación:** para su cálculo se debe implementar la siguiente función:

```
Extent = Funcion_Calcula_Extent(Matriz_Binaria)
```

`donde:`

- `Matriz_Binaria`: matriz binaria con dos posibles valores, *0*'s, píxeles que no son del objeto que se quiere caracterizar y *1*'s, píxeles del objeto a caracterizar.
- `Extent:` valor de extensión invariante a la rotación calculado como el máximo de los valores de extensión (razón de píxeles del objeto y píxeles de su bounding box) calculados para rotaciones del objeto de 0 a 355 grados (paso de 5 grados). Para ello, utilizar la función de Matlab `imrotate`.

Observación para la implementación de la función: esta función debe generar en primer lugar, a partir de la matriz binaria de entrada, una nueva matriz binaria con el objeto centrado. Sobre esta matriz se realizarán el resto de cálculos. Para ello utilizar la función facilitada `Funcion_Centra_Objeto.`

Imágenes a utilizar para la implementación de los descriptores y generación del conjunto de datos X-Y:

- **Imágenes de entrenamiento:** para obtener características de muestras cuya clase es conocida, seleccionar aquellas que sean las más adecuadas y diseñar, a partir de ellas, el clasificador.

- **Imágenes de test:** para comprobar el funcionamiento del clasificador diseñado.

Generación del conjunto de datos X-Y:

1.- Matriz X:

- Tantas filas como muestras u objetos haya en las imágenes disponibles.

  Cada columna representa un descriptor. Se deben generar en el siguiente orden:

```
% Compacticidad: 1
% Excentricidad: 2
% Solidez_CHull(Solidity): 3
% Extension_BBox(Extent): 4
% Extension_BBox(Invariante Rotacion): 5
% Hu1-Hu7: 6-12
% DF1-DF10: 13-22
% NumEuler: 23
```

2.- Vector Y:

- Vector columna, con los valores de codificación de cada clase del problema. El valor de Y de una determinada fila indica la clase a la que pertenece la instancia de esa fila de X.

ANEXO I - MOMENTOS INVARIANTES DE HU

Sea *f* una imagen binaria con dos posibles valores, *0*'s, píxeles que no son del objeto que se quiere caracterizar y *1*'s, píxeles del objeto a caracterizar:

- Momentos ordinarios de orden (p+q): $m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad p,q = 0,1,2,\ldots$
- Momentos centrales de orden (p+q): $\omega_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \;,\; \bar{x} = \frac{m_{10}}{m_{00}} \;;\; \bar{y} = \frac{m_{01}}{m_{00}}$
- Momentos centrales normalizados: $\eta_{pq} = \frac{\omega_{pq}}{(\omega_{00})^\gamma} \;;\; \gamma = \frac{p+q}{2}+1 \;;\; (p+q) = 2,3,\ldots$
- Momentos invariantes de Hu:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

- Para que todos los momentos se encuentren dentro del mismo orden de magnitud, determinar los momentos normalizados: $\phi^*_i = abs\left\{\ln\left[abs(\phi_i)\right]\right\}$. Si algún momento resulta ser 0, antes de calcular el momento normalizado reemplazar su valor por 1*exp(-100).