Universidad de Alcalá

Sistemas de Visión Artificial

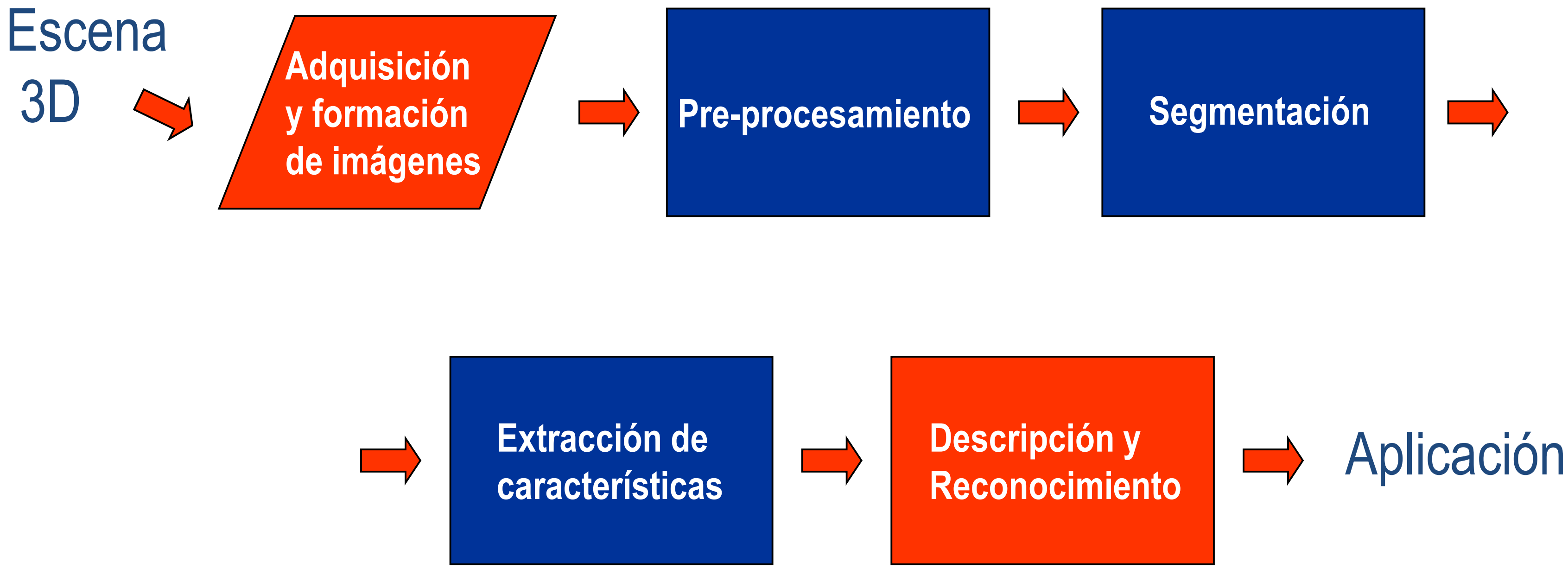
Tema 4: Representación y descripción de imágenes. Operaciones morfológicas.

Autores: Sira Palazuelos, Luis M. Bergasa , Manuel Mazo, M. Ángel García, Marisol Escudero, J. Manuel Miguel

Departamento de Electrónica. Universidad de Alcalá.

Departamento de Electrónica

Tareas generales de un sistema de visión por computador



- Ejemplo de aplicaciones:
  - Aplicación que contabiliza la cantidad total de dinero que hay en un conjunto dado de monedas y billetes.
  - Aplicación que clasifica piezas en una cinta transportadora.

Contenido

- Introducción
- Esquemas de representación
  - Código de cadena
  - Firmas o signaturas
- Descripción de contornos y regiones
  - Básicos: perímetro, área, compacidad, rectangularidad
  - Descriptores de Fourier
  - Descriptores topológicos
  - Momentos
  - Momentos invariantes
  - Momentos invariantes a partir del código cadena
  - Texturas
- Operaciones morfológicas
  - Erosión, dilatación, apertura, cierre, etc.

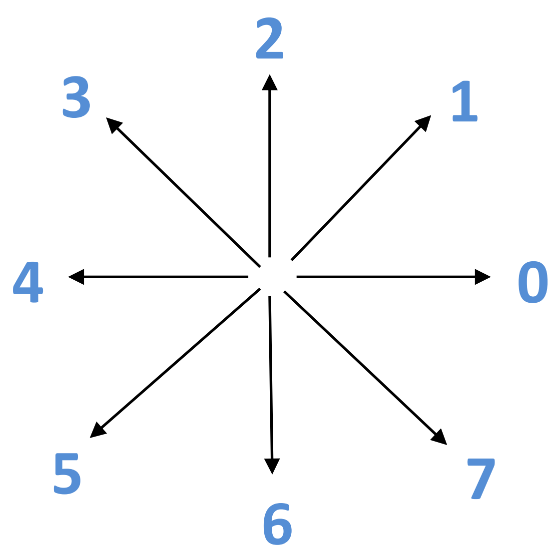
Introducción

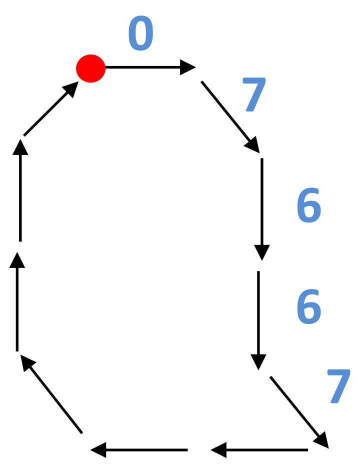
- Después de **segmentar** una imagen en regiones, **representaremos** éstas y las **describiremos** de la forma más adecuada para su posterior procesado.
- Básicamente, dos posibilidades para **representar** las regiones:
  - En términos de sus **características externas** (su contorno).
  - En términos de sus **características internas** (los píxeles internos).
- El siguiente paso es **describir** la región en la representación elegida. Algunos descriptores son:
  - Básicos: perímetro, área, compacidad, rectangularidad.
  - Momentos invariantes.
  - Momentos invariantes a partir del código cadena.
  - Descriptores topológicos.
  - Texturas.

Esquemas de representación

Representación de contornos
Código cadena

- Esquemas de representación externa
  - Código de cadena

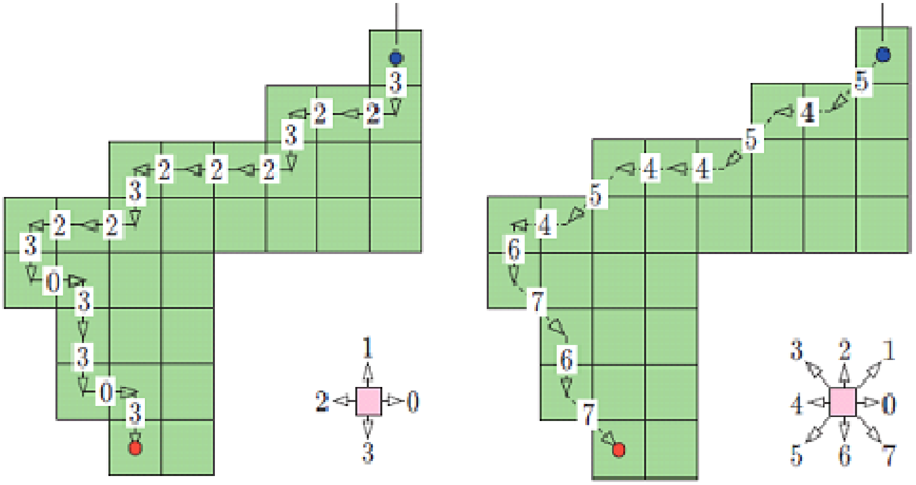




0 7 6 6 7 4 4 3 2 2 1

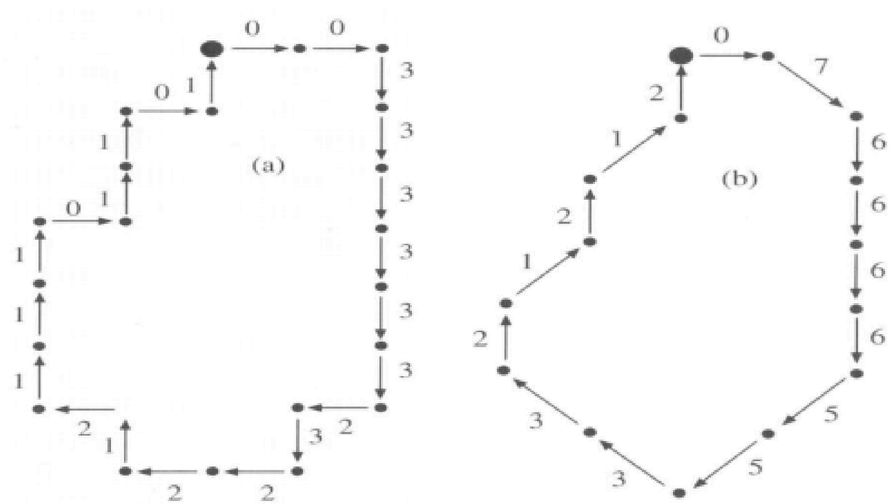
Representación de contornos

Código cadena



http://1.bp.blogspot.com/ -h2TQRsRvoBA/ UZqvJwdTicI/ AAAAAAAAAD8/ PJlUZAMBDS0/ s1600/cc1.png

http://4.bp.blogspot.com/-LnQmuh4Khr4/ UZqv8udEoTI/ AAAAAAAAAEE/_6UD3V9Ta2I/ s1600/cc2.png



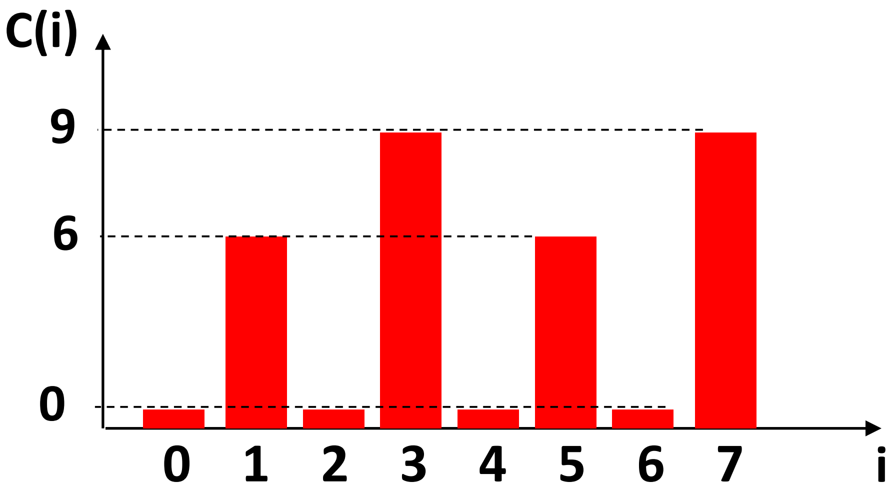
Representación de contornos
Código cadena

- Dado el código **cadena de un contorno**, se define el **histograma** del mismo de la siguiente forma:

$$A = \{i \mid C(i)\}; \quad N = cardinal\,(A)$$

  - Donde *C(i)* es la frecuencia de aparición del número de un determinado código de cadena, *i* designa el correspondiente código de cadena y *N* es el cardinal de *A* (el número de barras del histograma).

- Ejemplo: código cadena: (4,3), 111111777777777555555333333333

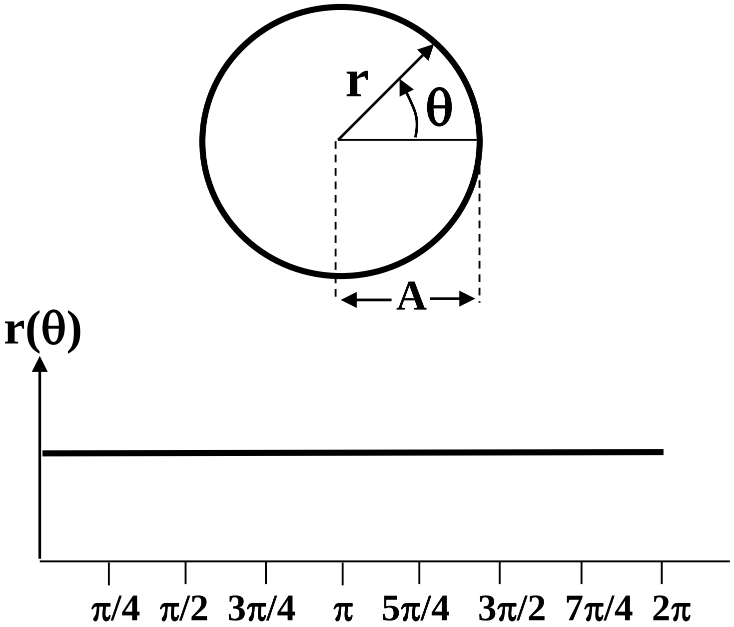


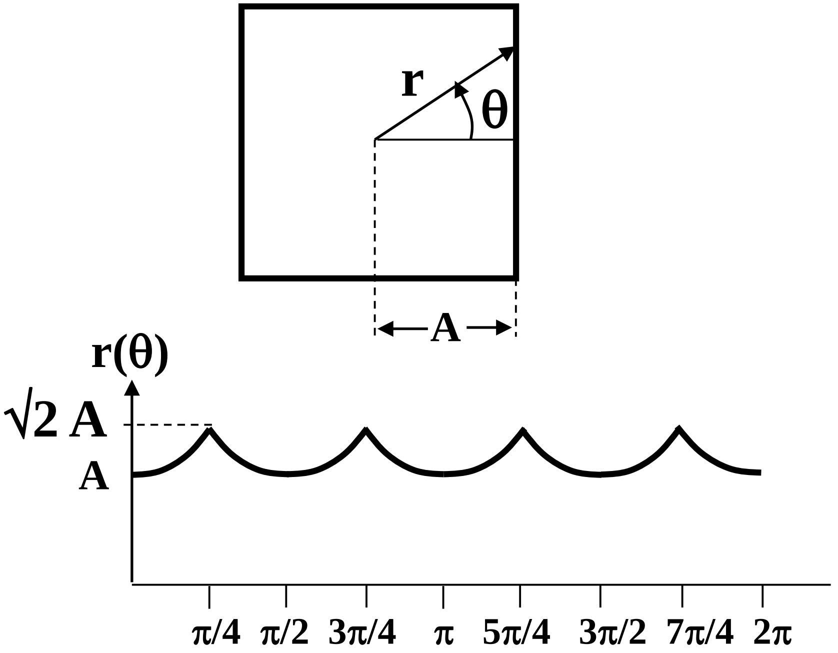
Representación de contornos
Código cadena: Algoritmo

1. Si el histograma tiene **4 o más barras (N≥4)**, la línea **no es recta**, ya que tiene al menos 4 orientaciones diferentes.
2. Si el histograma tiene una única barra **(N=1)**, la línea es **puramente recta** con alguna de las 8 posibles direcciones.
3. Si el histograma tiene 2 barras **(N=2)**, se pueden dar dos casos:
   - Si las dos barras son **adyacentes**, hay que considerar dos casos:
     - Si la máxima longitud del código de menor frecuencia es menor que un **umbral** prefijado T, la línea se declara **recta**.
     - Si la máxima longitud del código de menor frecuencia es mayor que un determinado umbral T, la línea se declara **no recta.**
   - Si las 2 barras **no son adyacentes**, esta línea se declara **no recta** (la línea contiene al menos dos orientaciones diferentes, y los ángulos de esas orientaciones difieren al menos 90º).
4. Si el histograma tiene **3 barras (N=3)**, se pueden dar dos casos:
   - Si las barras son **adyacentes** entre sí, la **barra central es la más larga**, y la altura de la barra vecina más próxima es menor que un umbral T fijado por el usuario, entonces es declarada como **recta**.
   - Si de las 3 barras 2 **no son adyacentes**, esta línea se declara **no recta**. La línea tiene al menos dos orientaciones diferentes cuyos ángulos difieren al menos 90º.

Representación de contornos
Signaturas

- **Signatura**:
  - La idea es **representar el contorno** como una **función polar unidimensional**.
  - El procedimiento más habitual consiste en calcular un **punto característico del interior del contorno**, por ejemplo el centro de masas (centroide) y a partir de él, representar la **distancia de cada punto del contorno a dicho centroide en función del ángulo.**

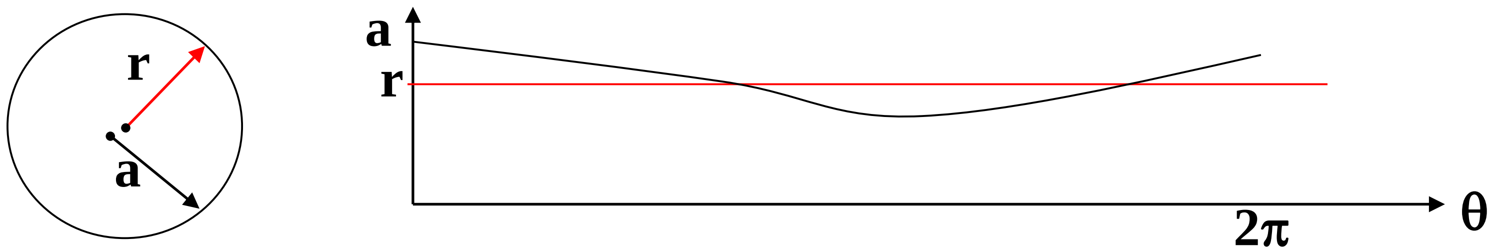




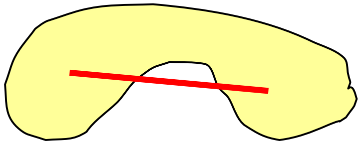
Representación de contornos
Signaturas

- Este tipo de representación es **invariante** a la **posición** del objeto en la imagen, pero **depende del tamaño** y del **punto de comienzo** (punto por donde se empieza a describir la frontera).
  - **Invarianza al tamaño**: se consigue **normalizando**, es decir, dividiendo la función por la distancia máxima al centroide, de forma que la distancia máxima será uno.
  - **Invarianza al ángulo de comienzo**: se consigue comenzando la representación por el ángulo para el cual la distancia es máxima.
- **Presenta dos inconvenientes:**

1. **Es muy sensible a la posición del centroide**

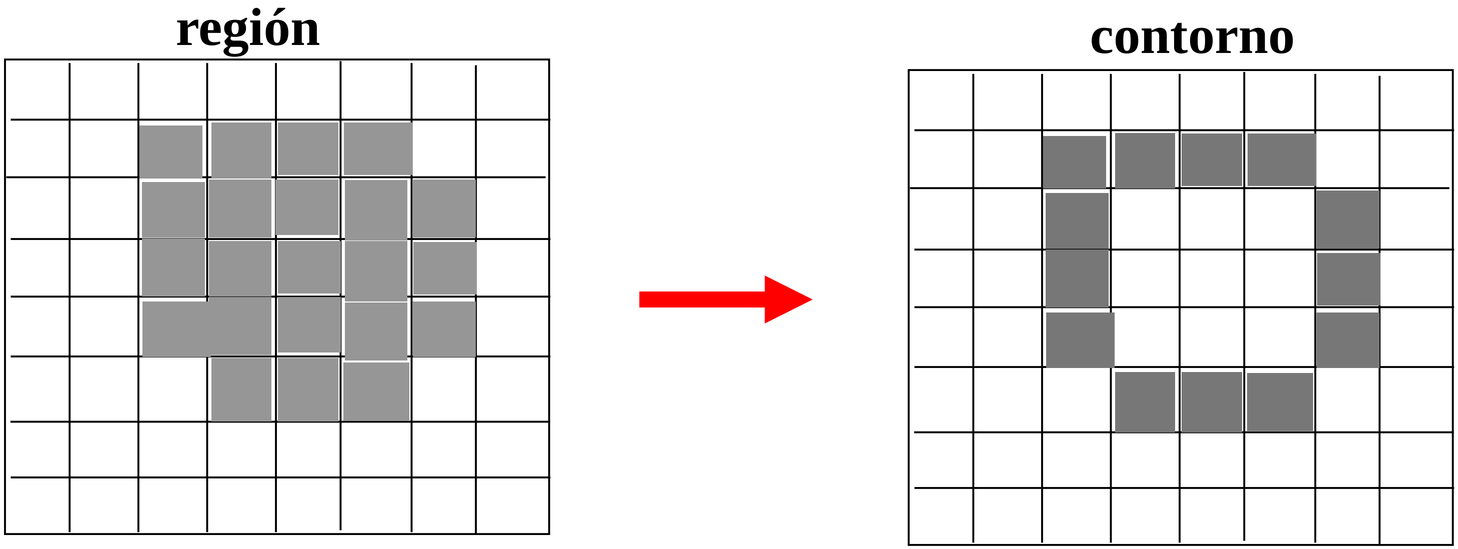


2. Las **concavidades** pueden dar lugar a una **representación multievaluada** para algunos ángulos (varias distancias para un mismo ángulo):



Descripción de contornos y regiones (descripción de objetos)

Descripción de objetos
Perímetro



- Para calcular el **perímetro** (P) solamente hay que recorrer completamente el contorno, partiendo de un punto:

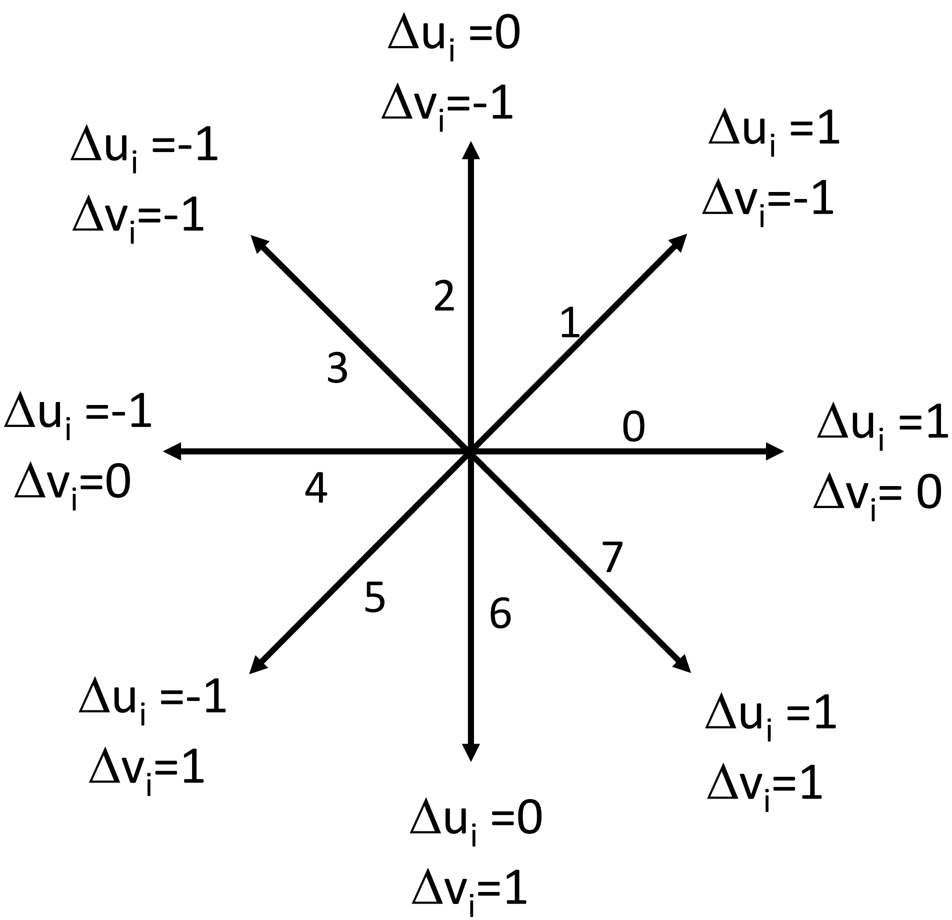
  P= Nº de desplazamientos horizontales + Nº de desplazamientos verticales + √2 x Nº desplazamientos diagonales.

- Utilizando el **código cadena**:

  P= Nº de códigos pares + √2 x Nº de códigos impares

Descripción de objetos
Área

$$A = \sum pixeles \in R;$$

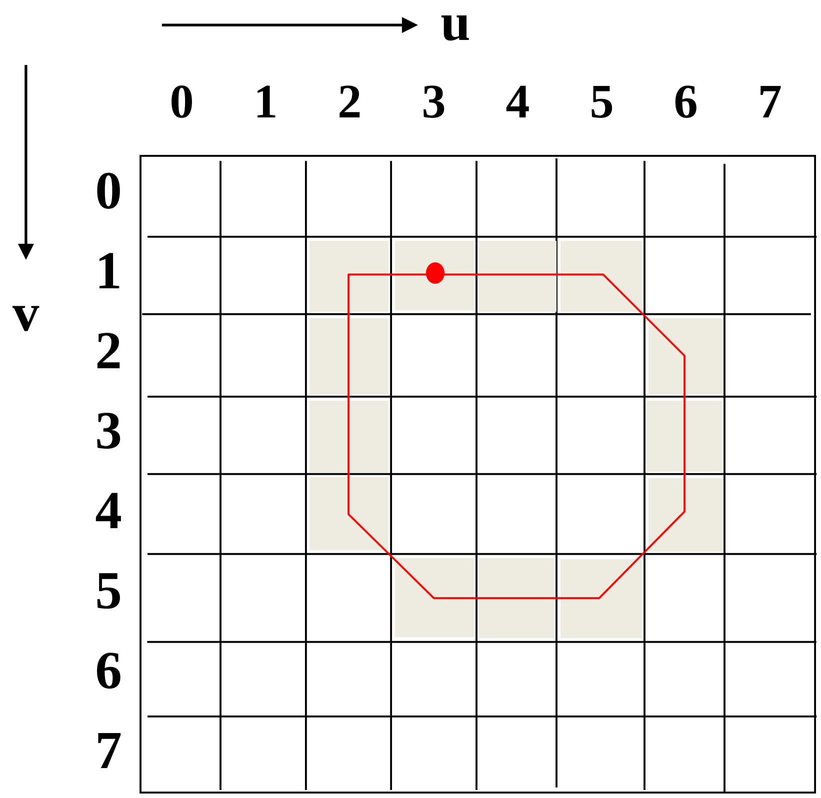


A partir del código cadena del contorno: para cada píxel de coordenadas (u,v) del contorno se define: $\Delta u_i = u_i - u_{i-1}$ e $\Delta v_i = v_i - v_{i-1}$, donde "i" es uno de los segmentos que forman el código cadena.

En este caso: Δu_i e Δv_i pueden tomar valores 1, 0, -1

$$A = \frac{1}{2}\left|\sum_{i=1}^{n}(u_i \Delta v_i - v_i \Delta u_i)\right|$$

Descripción de objetos
Área



$$A = \frac{1}{2}\left|\sum_{L=1}^{n}(u_i \Delta v_i - v_i \Delta u_i)\right| =$$

$$\frac{1}{2}\left| \begin{array}{l} [3\cdot 0 - 1\cdot 1] + [4\cdot 0 - 1\cdot 1] + [5\cdot 1 - 1\cdot 1] + \\ [6\cdot 1 - 2\cdot 0] + [6\cdot 1 - 3\cdot 0] + [6\cdot 1 - 4\cdot(-1)] + \\ [5\cdot 0 - 5\cdot(-1)] + [4\cdot 0 - 5\cdot(-1)] + \\ [3\cdot(-1) - 5\cdot(-1)] + [2\cdot(-1) - 4\cdot 0] + \\ [2\cdot(-1) - 3\cdot 0] + [2(-1) - 2\cdot 0] + \\ [2\cdot 0 - 1\cdot 1] \end{array} \right| = 14.5$$

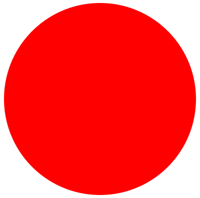
Obsérvese como el área obtenida utilizando el código cadena coincide con la encerrada por la línea roja (línea que une los centros de cada píxel del contorno)

Descripción de objetos

Compacidad, rectangularidad

- **Circularidad o compacidad (C):**

$$C = \frac{(\text{perímetro})^2}{\text{superficie}}$$



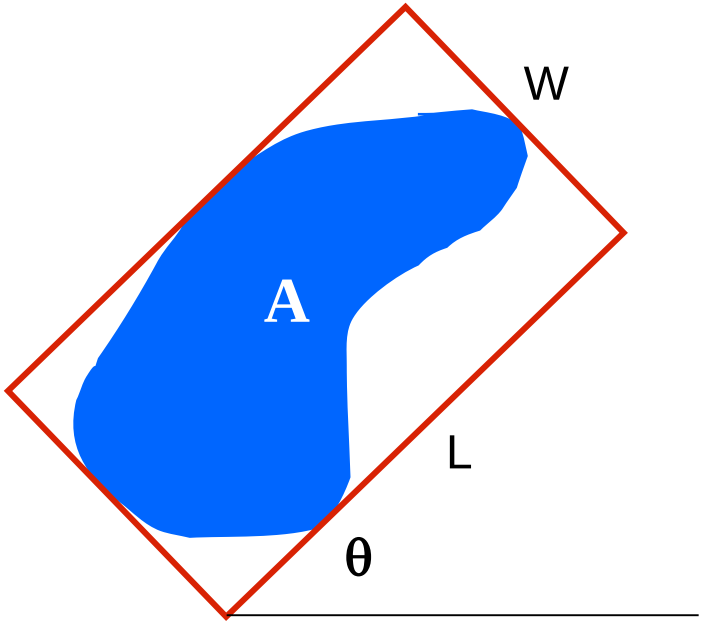
$$C = \frac{(2\pi r)^2}{\pi r^2} = 4\pi$$

$$C = \frac{(4L)^2}{L^2} = \frac{16L^2}{L^2} = 16$$

- **Rectangularidad (R):**

$$R = \frac{W \cdot L}{A}$$

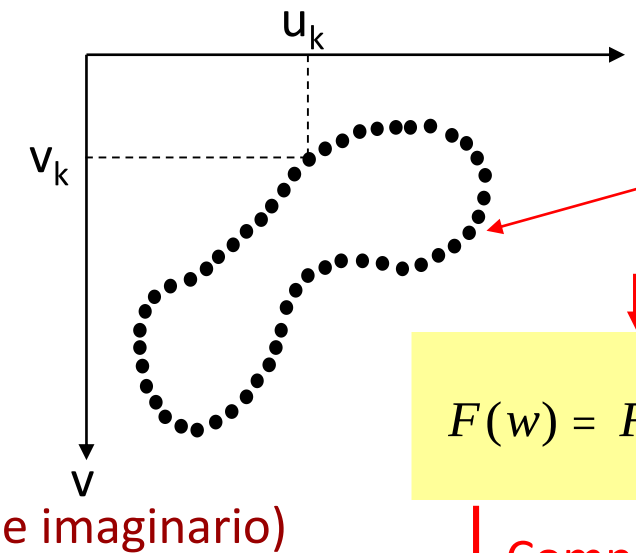
θ se elige para que WxL sea mínima (rectángulo mínimo).



Descripción de contornos

Descriptores de Fourier

- La **transformada de Fourier** permite extraer las **componentes en frecuencia de una curva discreta cerrada** (puesto que el contorno de un objeto es una curva cerrada, y por tanto periódica)



Contorno de N puntos. Código de cadena

Si: $s(k) = u_k + jv_k$

$$F(w) = F[s(k)] = \sum_{k=0}^{N-1} s(k)\exp[-\frac{j2\pi}{N}wk], \quad w = 0,1,2,\ldots N-1$$

Componentes de Fourier

$$\{F(w)\} = \{F(0), F(1), F(2), \ldots\ldots F(N-1)\}$$

Definen el contorno

Descripción de contornos

Descriptores de Fourier

- La **transformada de Fourier inversa** de F(w):

$$s(k)=\frac{1}{N}\sum_{w=0}^{N-1}F(w)\exp[\frac{j2\pi}{N}wk],\quad k=0,1,2,\ldots N-1$$

↑

Recuperación del contorno a partir de las componentes de Fourier

- En muchas ocasiones se puede recuperar el contorno con una buena aproximación con un número M de componentes F(w) inferior a N: **M componentes mayores.**
- Entre 10 y 15 descriptores son suficientes para definir cualquier forma → permite comprimir, utilizar muchos menos datos, sin perder información.

Descripción de contornos

Descriptores de Fourier

- Efecto de la **rotación, translación, escalado** en las componentes de Fourier

Rotación (θ):

$$\sum_{k=0}^{N-1}[s(k)e^{j\theta}]\exp[-\frac{j2\pi}{N}wk] = e^{j\theta}\sum_{k=0}^{N-1}s(k)\exp[-\frac{j2\pi}{N}wk]$$

Escalado (C):

$$\sum_{k=0}^{N-1}[Cs(k)]\exp[-\frac{j2\pi}{N}wk] = C\sum_{k=0}^{N-1}s(k)\exp[-\frac{j2\pi}{N}wk]$$

Translación (s_0):

$$\sum_{k=0}^{N-1}[s(k)+s_0]\exp[-\frac{j2\pi}{N}wk] = \sum_{k=0}^{N-1}s(k)\exp[-\frac{j2\pi}{N}wk] + s_0\sum_{k=0}^{N-1}\exp[-\frac{j2\pi}{N}wk]$$

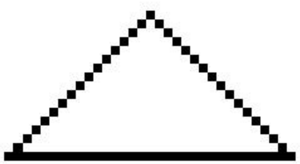
$$\sum_{k=0}^{N-1}\exp[-\frac{j2\pi}{N}wk]\begin{cases} = 0, & w \neq 0 \\ N, & w = 0 \end{cases}$$

- **Para hacer que los descriptores de Fourier (DF) sean invariantes a: rotación, translación, escalado:**

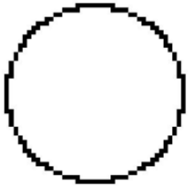
  1. Invariante a rotación: tomando solamente el **módulo:** ($|e^{j\theta}|=1$).
  2. Invariante a traslación: **eliminando F(0).**
  3. Invariante a escalado (homotecias): escalamos, **dividiendo por F(1)**

$$\{DF(i)\} = \left\{ \frac{|F(2)|}{|F(1)|}, \frac{|F(3)|}{|F(1)|}, \frac{|F(4)|}{|F(1)|}, \ldots\ldots\ldots\ldots\ldots, \frac{|F(M)|}{|F(1)|} \right\}$$

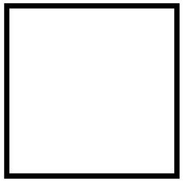
Descripción de contornos
Ejemplos



$$\{DF(i)\} = \{1.8083, \quad 0.2366, \quad 0.9284, \quad 0.1602\}$$



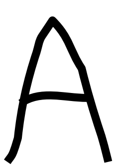
$$\{DF(i)\} = \{0.3763, \quad 0.2539, \quad 0.2040, \quad 0.1530\}$$



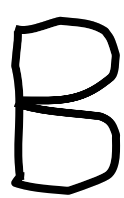
$$\{DF(i)\} = \{0.3022, \quad 0.3458, \quad 0.3458, \quad 0.1577\}$$

Descripción de objetos
Descriptores topológicos

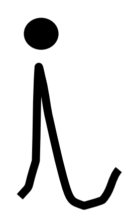
- Las **propiedades topológicas** se usan para describir los objetos de una imagen de forma global. Son propiedades que **no se ven afectadas por las deformaciones.**
- Los **descriptores topológicos** no tratan de dar un número exacto, sólo dar una **idea** sobre la forma del objeto.
- Algunos descriptores topológicos:
  - **Nº de huecos en una región (H).**
  - **Nº de componentes conectados** (C): elementos separados que forman un objeto.
  - **Nº de Euler**: Diferencia entre los dos anteriores (E=C-H).



C=1, H=1, E=0



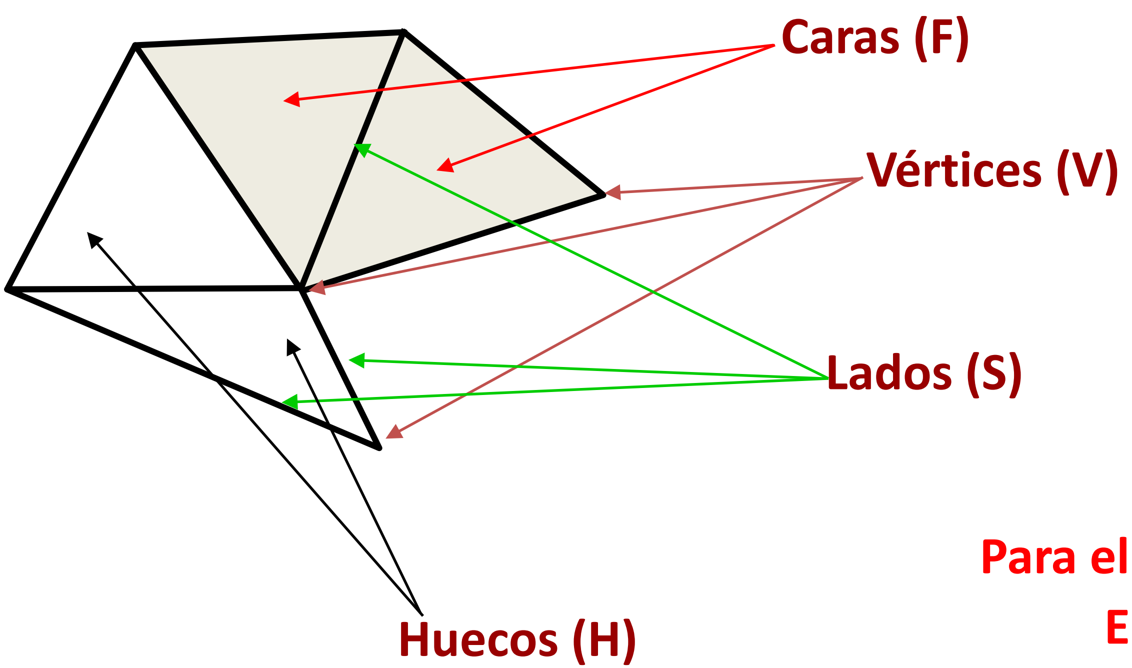
C=1, H=2, E=-1



C=2, H=0, E=2

Descripción de objetos
Descriptores topológicos

- **En las figuras formadas únicamente por líneas rectas, llamadas *redes poligonales,* suele ser de interés distinguir entre dos tipos de regiones interiores de dicha red: *caras (C)* y *huecos (H)*.**



E=V-S+F=C-H

Para el ejemplo de la figura:

E=6- 9+2= 1-2=-1

Descripción de objetos
Momentos

- Dada **una función continua** f(u,v) se define su momento de orden p+q:

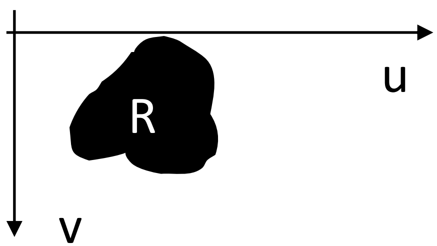
$$m_{p,q} = \iint_R u^p v^q f(u,v)\,du\,dv$$

- Si **f(u,v)=1** (imagen umbralizada)**:**

$$m_{p,q} = \iint_R u^p v^q \,du\,dv$$

- En el caso de una **imagen digital binaria** [f(u,v)=1: objeto, f(u,v)=0: fondo]:

$$m_{pq} = \sum_u \sum_v u^p v^q$$



$$f(u,v) = 1, \forall (u,v) \in R$$
$$f(u,v) = 0, \forall (u,v) \notin R$$

donde el sumatorio se toma sobre todas las coordenadas (u,v) de puntos de la región (objeto).

$$\mathrm{m}_{00} = \sum_{\mathrm{u}} \sum_{\mathrm{v}} 1 = \text{superficie en píxeles de un objeto}$$

Descripción de objetos
Momentos invariantes

- **Momentos invariantes a la posición (translación):** ***momentos centrales***

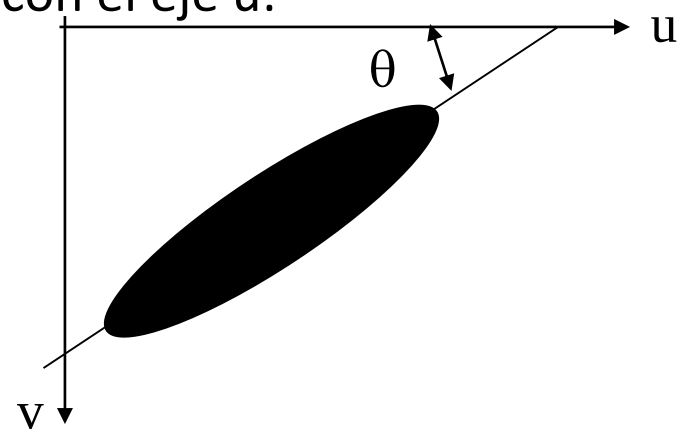
$$\mu_{pq} = \sum_u \sum_v (u - \bar{u})^p (v - \bar{v})^q$$

donde $\bar{u}$ y $\bar{v}$ son las coordenadas del **centroide** o centro de masas del objeto:

$$\bar{u} = \frac{m_{10}}{m_{00}} = \frac{\sum\sum u}{\sum\sum 1}; \quad \bar{v} = \frac{m_{01}}{m_{00}} = \frac{\sum\sum v}{\sum\sum 1}$$

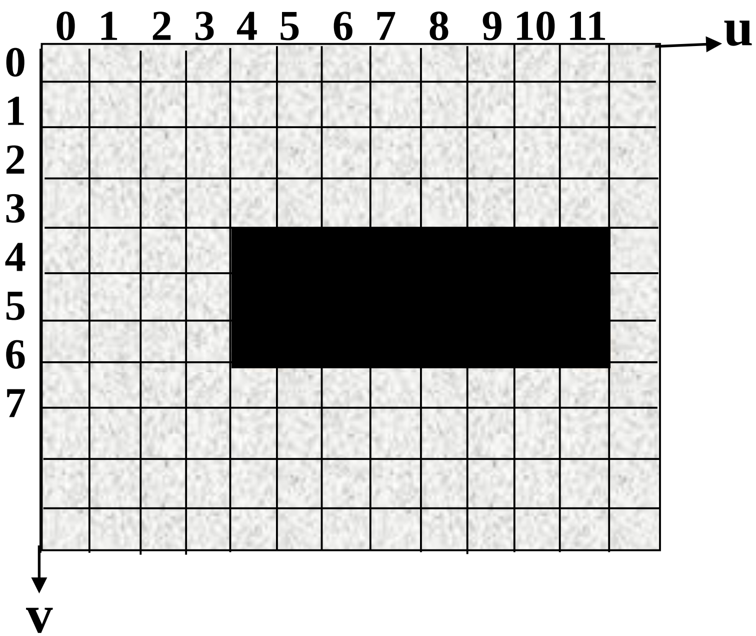
- **Ángulo que forma el eje de mínima inercia** con el eje u:

$$\theta = \frac{1}{2}\tan^{-1}\frac{2(m_{00}m_{11} - m_{10}m_{01})}{(m_{00}m_{20} - m_{10}^2) - (m_{00}m_{02} - m_{01}^2)}$$



Descripción de objetos

Momentos invariantes: Ejemplos



$$m_{00} = \sum_{u=4}^{11}\sum_{v=4}^{6} 1 = 24$$

$$m_{10} = \sum_{u=4}^{11}\sum_{v=4}^{6} u = 4x3 + 5x3 + 6x3 + 7x3 + \ldots\ldots + 11x3 = 180$$

$$m_{01} = \sum_{u=4}^{11}\sum_{v=4}^{6} v = 4x8 + 5x8 + 6x8 = 120$$

$$m_{20} = \sum_{u=4}^{11}\sum_{v=4}^{6} u^2 = 4^2x3 + 5^2x3 + \ldots.11^2x3 = 1476$$

$$m_{02} = \sum_{u=4}^{11}\sum_{v=4}^{6} v^2 = 4^2x8 + 5^2x8 + 6^2x8 = 1476$$

$$m_{11} = \sum_{u=4}^{11}\sum_{v=4}^{6} uv = 4x4 + 4x5 + 4x6 + 5x4 + + \ldots 11x5 + 11x6 = 900$$

$$\bar{u} = \frac{180}{24} = 7.5; \ \bar{v} = \frac{120}{24} = 5;$$

$$\theta = \frac{1}{2}\tan^{-1}\frac{2(24x900 - 180x120)}{(24x1476 - 180^2) - (24x616 - 120^2)} = 0^{o}$$

- Los **momentos centrales** se pueden obtener a partir de los **generales**, según la siguiente expresión:

$$\mu_{pq} = \sum_{r=0}^{p} \sum_{s=0}^{q} C_r^p C_s^q (-\bar{u})^r (-\bar{v})^s m_{p-r} m_{q-s}$$

$$con: C_r^p = \frac{p!}{r!(p-r)!}, C_s^q = \frac{q!}{s!(q-s)!}$$

$$\mu_{10} = \sum \sum (u - \bar{u}) = 0 \quad \mu_{01} = \sum \sum (v - \bar{v}) = 0$$

Descripción de objetos
Momentos invariantes

- Los **momentos invariantes al escalado** (momentos centrales invariantes) se obtienen a partir de:

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}}; \quad donde: \gamma = \frac{p+q}{2} + 1,\ para\ \ p+q = 2,3,\ldots.$$

- Demostración de invarianza frente al escalado utilizando los momentos generales (m_{pq}) :

Normalizando con respecto al escalado:

$$m^*_{pq} = \iint (u^*)^p (v^*)^q du^* dv^*; u^* = \lambda u,\ v^* = \lambda v \Rightarrow m^*_{pq} = \lambda^{2+p+q} \iint u^p v^q du dv \rightarrow m^*_{pq} = \lambda^{2+p+q} m_{pq}$$

Si se fuerza a que $m^*_{00} = 1$ entonces:

$$\lambda^{2+p+q} = \frac{1}{(m_{00})^{\frac{p+q}{2}+1}} \rightarrow m^*_{pq} = \frac{m_{pq}}{(m_{00})^{\frac{p+q}{2}+1}},\ p+q \geq 2$$

Descripción de objetos
Momentos invariantes

- **Finalmente, los siete momentos invariantes a traslaciones, rotaciones y cambios de escala, propuestos por Hu (1962):**

$$\boxed{\phi_{pq} = \sum_{r=0}^{p}\sum_{s=0}^{q}(-1)^{q-s} C_r^p C_s^q (\cos\theta)^{p-r+s}(sen\,\theta)^{q-s+r}\,\eta_{p-r+q-s,r+s}\,,\; \theta = \frac{1}{2}\tan^{-1}\frac{2\eta_{11}}{\eta_{20}-\eta_{02}}}$$

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

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

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

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

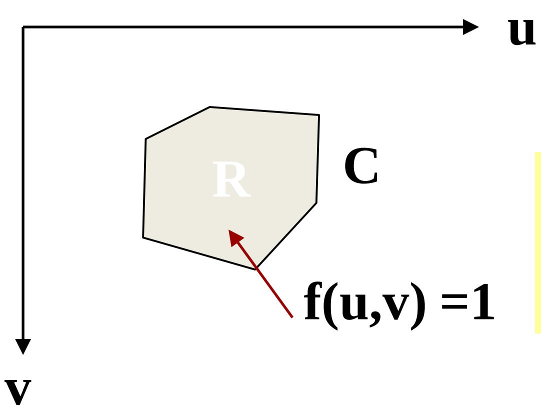
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

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

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

Descripción de objetos

Momentos invariantes a partir del código cadena



$$m_{pq} = \iint_R u^p v^q \, du dv$$

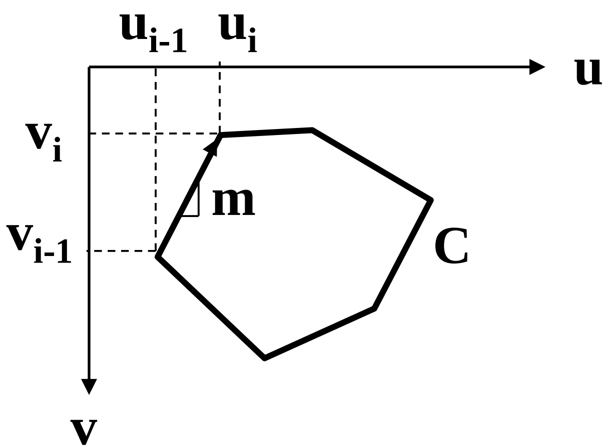
$$\iint_R \left(\frac{\partial S}{\partial u} - \frac{\partial T}{\partial v} \right) du dv = \int_C S dv + T du; \; si: \begin{cases} S = u^{p+1} v^q \rightarrow \dfrac{\partial S}{\partial u} = (p+1) u^p v^q \\ T = -u^p v^{q+1} \rightarrow \dfrac{\partial T}{\partial v} = -(q+1) u^p v^q \end{cases}$$

$$\iint_R (p+q+2) u^p v^q \, du dv = \int_C u^{p+1} v^q dv - u^p v^{q+1} du$$

$$m_{pq} = \frac{1}{p+q+1} \iint_R (p+q+2) u^p v^q \, du dv = \frac{1}{p+q+1} \int_C u^{p+1} v^q dv - u^p v^{q+1} du$$

Descripción de objetos

Momentos invariantes a partir del código cadena



$$v - v_i = m_i(u - u_i)$$

$$m_i = \frac{v_i - v_{i-1}}{u_i - u_{i-1}} = \frac{\Delta v_i}{\Delta u_i} \approx \frac{dv}{du}$$

$$m_{pq} = \frac{1}{p+q+1}\int_C u^{p+1}[v_i + m_i(u-u_i)]^q m_i\, du - u^p[v_i + m_i(u-u_i)]^{q+1} du;$$

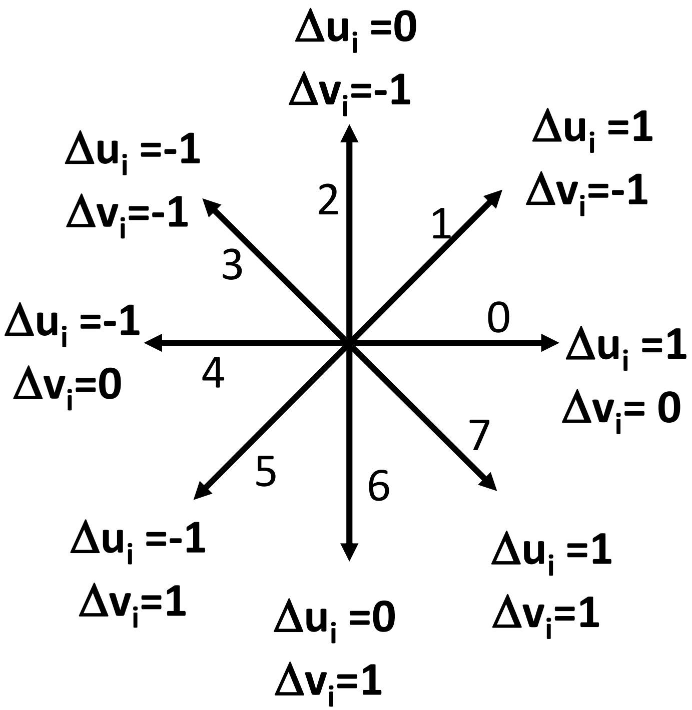
$i = 1,2,\ldots.N$ **donde N es el número de segmentos que constituyen el objeto**

$$m_{pq} = \frac{1}{p+q+1}\sum_{i=1}^{N}\left[\int_{u_{i-1}}^{u_i} u^{p+1}[v_i + m_i(u-u_i)]^q m_i - u^p[v_i + m_i(u-u_i)]^{q+1}\right]du$$

Descripción de objetos

Momentos invariantes a partir del código cadena

Llamando: $A_i = (u_i \Delta v_i - v_i \Delta u_i)$



$$m_{00} = \frac{1}{2}\sum_{i=1}^{N} A_i$$

$$m_{01} = \frac{1}{3}\sum_{i=1}^{N} A_i\left(v_i - \tfrac{1}{2}\Delta v_i\right)$$

$$m_{10} = \frac{1}{3}\sum_{i=1}^{N} A_i\left(u_i - \tfrac{1}{2}\Delta u_i\right)$$

$$m_{11} = \frac{1}{4}\sum_{i=1}^{N} A_i\left(u_i v_i - \tfrac{1}{2}u_i\Delta v_i - \tfrac{1}{2}v_i\Delta u_i + \tfrac{1}{3}\Delta u_i\Delta v_i\right)$$

$$m_{20} = \frac{1}{4}\sum_{i=1}^{N} A_i\left(u_i^2 - u_i\Delta u_i + \tfrac{1}{3}\Delta u_i^2\right)$$

$$m_{02} = \frac{1}{4}\sum_{i=1}^{N} A_i\left(v_i^2 - v_i\Delta v_i + \tfrac{1}{3}\Delta v_i^2\right)$$

$$m_{30} = \frac{1}{5}\sum_{i=1}^{N} A_i\left(u_i^3 + u_i\Delta u_i^2 - \tfrac{3}{2}u_i^2\Delta u_i - \tfrac{1}{4}\Delta u_i^3\right)$$

$$m_{03} = \frac{1}{5}\sum_{i=1}^{N} A_i\left(v_i^3 + v_i\Delta v_i^2 - \tfrac{3}{2}v_i^2\Delta v_i - \tfrac{1}{4}\Delta v_i^3\right)$$

m_{00} = perímetro

Texturas

Texturas
Generalidades

- No existe una definición formal de **textura**, aunque la idea intuitiva es fácil de asimilar.
- Existen **parámetros** mediante los cuales se pueden evaluar texturas.
- Conceptualmente, **la característica principal de una textura es la repetición de un patrón básico.**
- La **estructura** del patrón básico puede **no ser determinista**, sino **estadística**.
- La **repetición** del patrón básico puede **no ser ni regular ni determinista, sino estadísticamente regular**.

Texturas
Ejemplos

Texturas

¿Dónde está la textura de un objeto?

¿Transición de objetos a texturas?

Texturas

¿Dónde está la textura de un objeto?

Texturas
Clasificación

- **Regla de repetición:**
  1. **Repetitivas:** repetición periódica de un patrón sobre una superficie mucho mayor que el patrón.
  2. **Aleatorias:** El patrón que se repite no lo hace de forma periódica, aunque exista cierta regla de formación.

- **Patrón:**
  3. **Determinístico** (regular o estructurado): ordenamientos regulares que normalmente son artificiales.
  4. **Estocásticos** (irregular o aleatorio): suelen ser naturales: tierra, arena, etc.

1.
3.


2.


4.


Texturas

Modelos en el tratamiento

- **Métodos estructurales**: La descripción de la textura se realiza a través de **gramáticas**. Una de las más interesantes es la denominada gramática de formas.

- **Análisis frecuencial**: Utiliza el análisis en el dominio de la frecuencia, porque una textura es una señal más o menos periódica. El contenido frecuencial de una textura da idea sobre su distribución en el espacio.

- **Métodos estadísticos**:
  - ✓ **Estadísticos de primer orden**: Miden valores dependientes de los niveles de gris de la imagen. **Dependen sólo de los niveles de gris de un píxel** y **no** de su relación con los vecinos. Los atributos de un histograma son estadísticos de primer orden.
  - ✓ **Estadísticos de segundo orden**: se basan en la **observación de un par de valores de gris** que ocurren en los extremos de un "dipolo" de longitud preestablecida, situado en cualquier posición y orientación de la imagen. Son propiedades de pares de píxeles.

- Solamente nos referiremos al análisis frecuencial y a los métodos estadísticos.

- Dado que un **atributo** de las texturas es la **frecuencia de repetición del patrón**, una **textura fina** (con muchos cambios) tendrá componentes de **alta frecuencia**, mientras que una **textura más rugosa** (patrón de repetición más grande) tendrá su energía concentrada en las **bajas frecuencias**.

- **Autocorrelación** *(C):* Es uno de los parámetros más utilizados dentro del análisis en frecuencia:

$$C(i,j) = \frac{\sum_{u \in W} \sum_{v \in W} I(u,v) I(i+u, j+v)}{\sum_{u \in W} \sum_{v \in W} I^2(u,v)}$$

donde W es una región (ventana) de la imagen.

Texturas

Estadísticos de primer orden

- Si P(f) es el **histograma**, suponiendo L niveles de gris diferentes, el **momento de orden "n" respecto a la media** se define como:

$$m^n = \sum_{f=0}^{L-1} (f - \bar{f})^n P(f)$$

$$m^0 = \sum_{f=0}^{L-1} (f - \bar{f})^0 P(f) = 1; \quad m^1 = \sum_{f=0}^{L-1} (f - \bar{f})^1 P(f) = 0; \quad m^2 = \sigma^2 = \sum_{f=0}^{L-1} (f - \bar{f})^2 P(f) \rightarrow \text{varianza};$$

  - Siendo el **nivel medio de gris**:

$$\bar{f} = \sum_{f=0}^{L-1} f \cdot P(f)$$

- El **momento de segundo orden** es de particular interés para la descripción de texturas. En concreto la **suavidad relativa (R)**:

$$R = 1 - \frac{1}{1 + m^2}$$

Texturas
Estadísticos de segundo orden

- **Matriz de co-ocurrencia**: Es una de las fuentes de propiedades de la textura más importantes.
- Dada una imagen, se define:
  - ✓ Un **operador de posición de píxel** dentro de la imagen:

$$P \equiv (d \cdot \cos\theta,\ d \cdot \mathrm{sen}\theta)$$

    - ✓ Ejemplos: $P \equiv (\sqrt{2} \cdot \cos 45, \sqrt{2} \cdot \mathrm{sen} 45) = (1,1)$ = *“un píxel por encima y uno a la derecha”*
    - ✓ $P \equiv (1 \cdot \cos 180, 1 \cdot \mathrm{sen} 180)$ = *“un píxel a la izquierda”.*
  - ✓ Una **matriz A** de dimensiones k x k cuyo elemento a_{ij} es el número de veces que los píxeles cuya intensidad es z_i aparecen en la posición especificada por P en relación a puntos de intensidad z_j, con $1 \leq i, j \leq k$ (número de niveles de gris).

Texturas

Estadísticos de segundo orden

- Ejemplo: considerando P= *"un píxel por encima y uno a la derecha"*, y una imagen con tres niveles de gris: z_0=0, z_1=1, z_2=2, los elementos de la matriz A son:

  a_{11}: Número de veces que un píxel de intensidad z_0=0 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_0=0 .

  a_{12}: Número de veces que un píxel de intensidad z_0=0 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_1=1 .

  a_{13}: Número de veces que un píxel de intensidad z_0=0 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_2=2 .

  a_{21}: Número de veces que un píxel de intensidad z_1=1 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_0=0.

  a_{22}: Número de veces que un píxel de intensidad z_1=1 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_1=1.

  a_{23}: Número de veces que un píxel de intensidad z_1=1 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_2=2.

  a_{31}: Número de veces que un píxel de intensidad z_2=2 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_0=0.

  a_{32}: Número de veces que un píxel de intensidad z_2=2 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_1=1.

  a_{33}: Número de veces que un píxel de intensidad z_2=2 aparece un píxel a la derecha y un píxel por encima de píxeles de intensidad z_2=2

- Para P= *"un píxel por encima y uno a la derecha"* y una imagen I como la indicada, la matriz A es:

$$I = \begin{bmatrix} 0 & \boxed{1} & 0 & 2 & 0 \\ 0 & \boxed{1} & \boxed{1} & 2 & \boxed{1} \\ 0 & \boxed{1} & 0 & 0 & 2 \\ 1 & 0 & 2 & \boxed{1} & 2 \\ 0 & 0 & 1 & 2 & 1 \end{bmatrix} \longrightarrow A = \begin{bmatrix} 2 & 1 & 2 \\ \boxed{3} & \boxed{3} & \boxed{0} \\ 2 & 2 & 1 \end{bmatrix}$$

- El tamaño de la matriz A está determinado únicamente por el número de niveles de intensidad distintas de la imagen de entrada. En ocasiones se cuantifica la imagen de entrada con un menor nivel de grises para que A sea *"manejable"*.

Texturas

Estadísticos de segundo orden

- Tomando **n como el número total de pares de puntos de la imagen que satisfacen P** (en el ejemplo anterior n=2+1+2+3+3+0+2+2+1=16).
- Se define una **matriz C dividiendo cada elemento de A por n.**
- Cada **elemento de la matriz C, c_{ij} es una estimación de la probabilidad compuesta de que un par de puntos que satisfagan P tengan valores (z_i, z_j).**
- La matriz **C se llama matriz de co-ocurrencia del nivel de gris.**
- Debido a **que C depende de P (operador de posición de píxel), es posible detectar la presencia de unos patrones de textura dados, eligiendo adecuadamente el operador P**.
- Un conjunto de descriptores propuestos por Haralick (1979) o Ballard y Brown (1982) utilizando los coeficientes de la matriz C se indican a continuación.

Texturas
Estadísticos de segundo orden

1. **Probabilidad máxima:**

$$\max_{i,j}(c_{ij})$$

2. **Energía:**

$$\sum_i \sum_j \left| c_{ij} \right|^2$$

3. **Momento de distancia de elementos de orden k:**

$$\sum_i \sum_j (i-j)^k c_{ij}$$

4. **Momento inverso de distinción de elementos de orden k:**

$$\frac{\sum_i \sum_j c_{ij}}{(i-j)^k} \qquad i \neq j$$

Texturas

Estadísticos de segundo orden

5. **Entropía:**

$$-\sum_i \sum_j c_{ij} \log c_{ij}$$

6. **Uniformidad:**

$$\sum_i \sum_j c_{ij}^2$$

7. **Correlación:**

$$\frac{\sum_i \sum_j (i-\mu_x)(j-\mu_y)c_{ij}}{\sigma_x \sigma_y}$$

donde:

$$\mu_x = \sum_i i \sum_j c_{ij};\ \mu_y = \sum_i j \sum_j c_{ij};\ \sigma_x^2 = \sum_i (i-\mu_x)^2 \sum_j c_{ij};\ \sigma_y^2 = \sum_j (i-\mu_y)^2 \sum_i c_{ij}$$

Texturas
Estadísticos de segundo orden

8. Inercia:

$$\sum_i \sum_j (i - j)^2 c_{ij}$$

9. Homogeneidad local:

$$\sum_i \sum_j \frac{1}{1 + (i - j)^2} c_{ij}$$

Operaciones morfológicas (filtrado morfológico)

Operaciones morfológicas

- Las **operaciones morfológicas** consisten en desplazar B (elemento estructurante) sobre la imagen A y realizar las operaciones específicas entre el solape de B sobre A.
- El **elemento estructurante** define el vecindario mediante una matriz de “1s” y “0s”. El píxel central representa el píxel de interés mientras que los demás elementos de la matriz que estén a “1" definen la vecindad que consideraremos.
- Algunos elementos estructurantes:

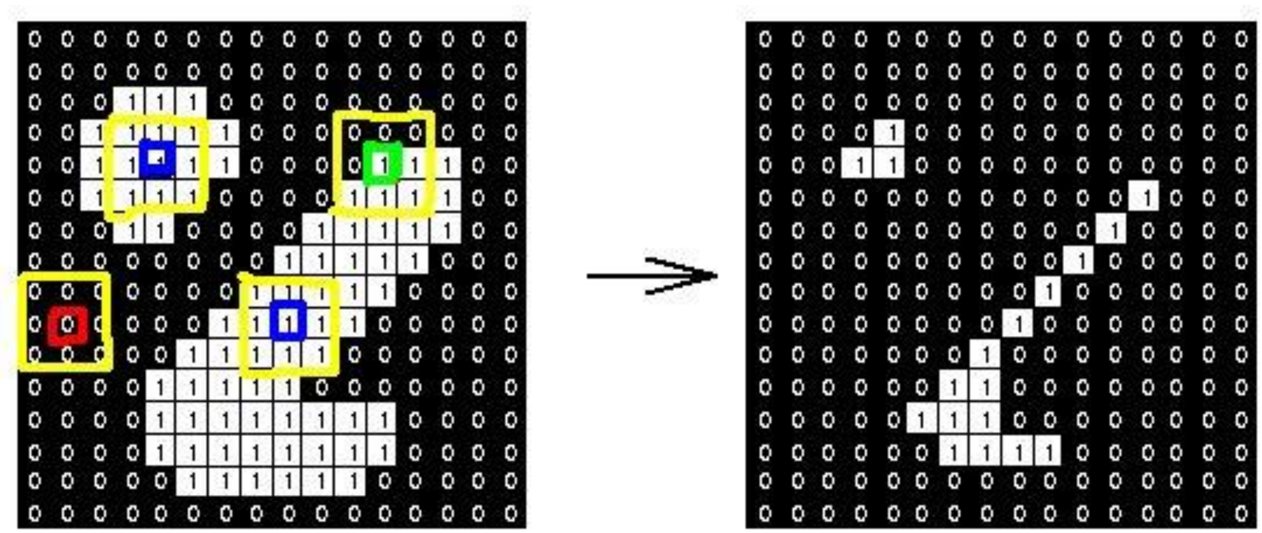
1	1	1
1	**1**	1
1	1	1

0	1	0
1	**1**	1
0	1	0

1	0	1
0	**1**	0
1	0	1

Operaciones morfológicas

- **Erosión**: Un pixel de la imagen de salida será “1” si, cuando se sitúa el elemento estructurante sobre esa posición en la imagen de entrada, **todos** los píxeles a 1 del elemento estructurante son 1 en la imagen original. En caso contrario, el pixel correspondiente en la imagen de salida será “0".
- Si se aplica varias veces los objetos desaparecen.

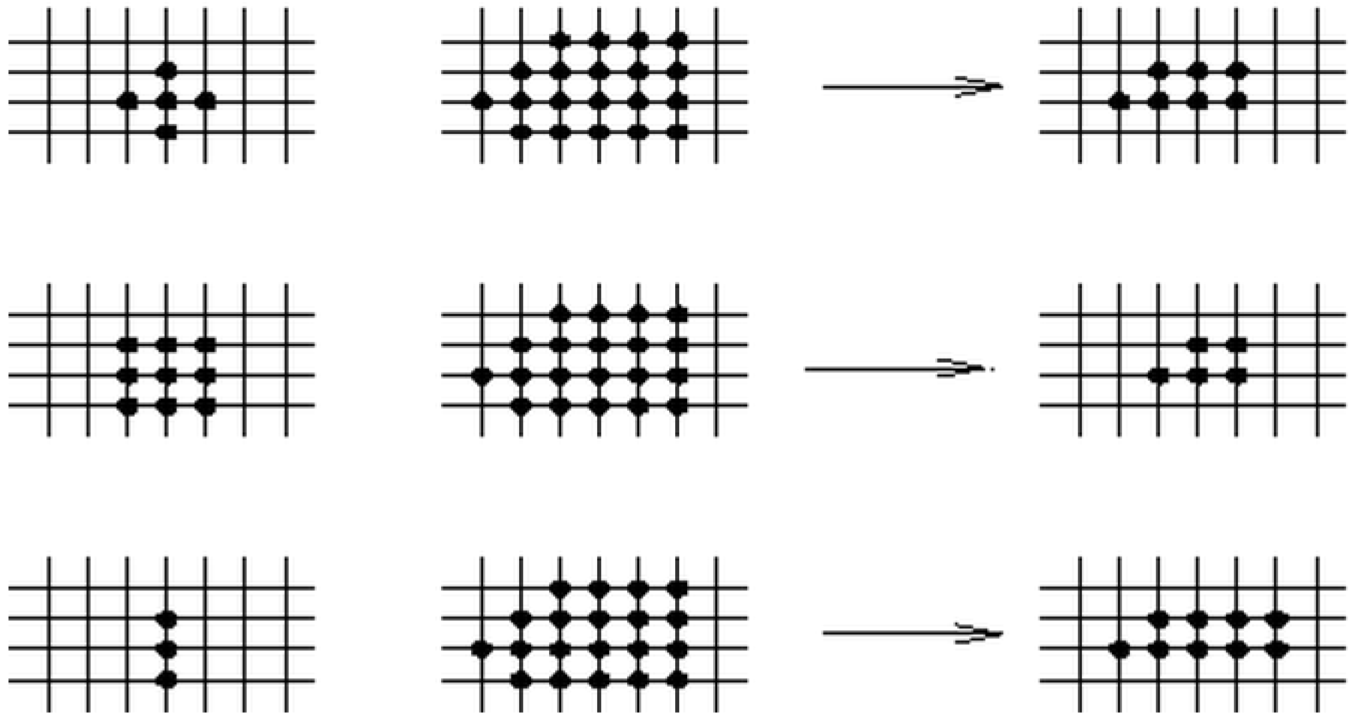


http://chilmania.blogspot.com.es/2010/06/matlab-code-for-dilation-and-erosion.html

Operaciones morfológicas

- Si se erosiona un objeto con distintos elementos estructurantes el resultado será diferente.

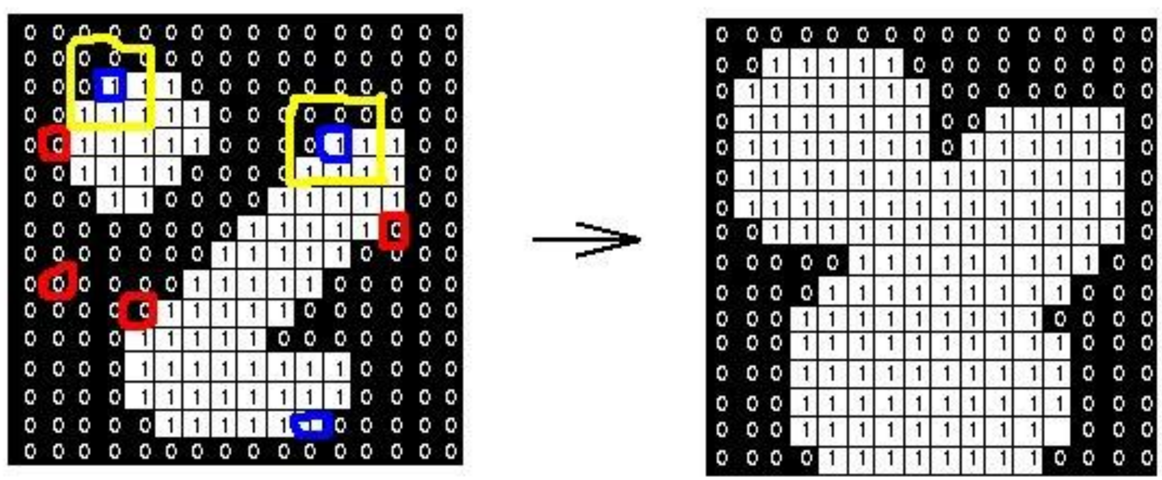
Elem. estructural | Objeto | Objeto erosionado



http://utam.gg.utah.edu/tomo03/03_mid/HTML/node103.html

Operaciones morfológicas

- **Dilatación**: Un pixel de la imagen de salida será “1” si, cuando se sitúa el elemento estructurante sobre esa posición en la imagen de entrada, **cualquiera** de los píxeles a 1 del elemento estructurante son 1 en la imagen original. En caso contrario será “0".
- Si se aplica varias veces, el fondo desaparece.
- Si se dilata un objeto con distintos elementos estructurantes el resultado será diferente.



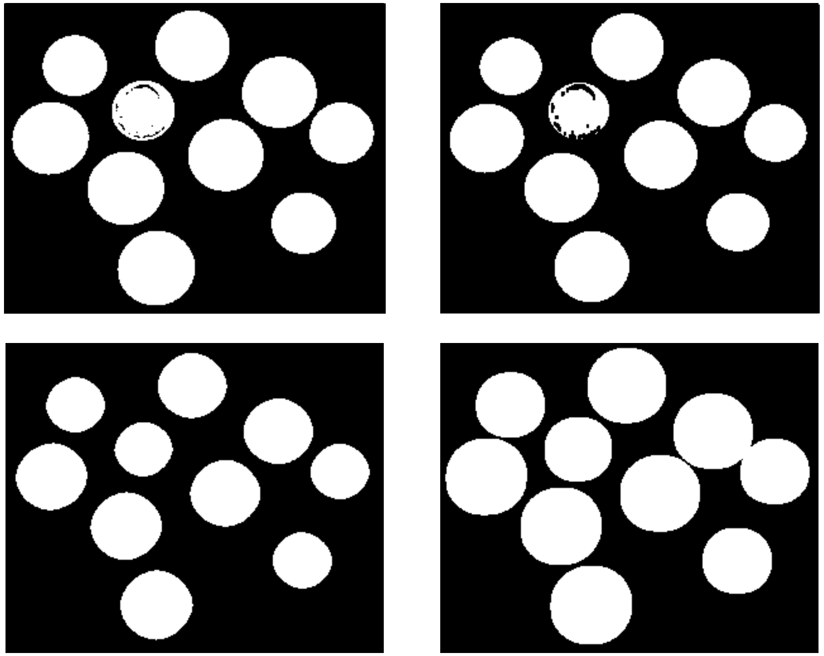
http://chilmania.blogspot.com.es/2010/06/matlab-code-for-dilation-and-erosion.html

Operaciones morfológicas

```
BW1 = im2bw(imread('coins.png'),0.4); %im2bw convierte la
    imagen leída a binaria.
figure, imshow(BW1);

SE=ones(3,3);
BW2 = imerode(BW1,SE);
figure, imshow(BW2);

BW3 = imdilate(BW1,SE);
figure, imshow(BW3);
```



- Observe en la parte inferior el efecto de la dilatación usando dos elementos estructurantes diferentes.

Apertura y cierre

- **Apertura**: Erosión seguida de dilatación. Elimina pequeños píxeles aislados que haya en la imagen.

$$A \circ B = (A \ominus B) \oplus B$$

- **Cierre**: Dilatación seguida de erosión. Rellena/cierra los pequeños agujeros que existan en la imagen.

$$A \bullet B = (A \oplus B) \ominus B$$

- Apertura y cierre poseen la **propiedad de idempotencia**: tras la primera aplicación, la figura no sufrirá más cambios en la posteriores aplicaciones.

Operaciones morfológicas

- Ejemplo de apertura y cierre:

```
BW1=im2bw(imread('coins.png'),0.35);
figure, imshow(BW1);

SE=ones(5,5);
BW2=imerode(BW1,SE);  % Erosión
BW3=imdilate(BW2,SE);  % Apertura
figure, imshow(BW3);

BW4=imdilate (BW1,SE);  % Dilatación
BW5=imerode (BW4,SE);  % Cierre
figure, imshow(BW5);
```

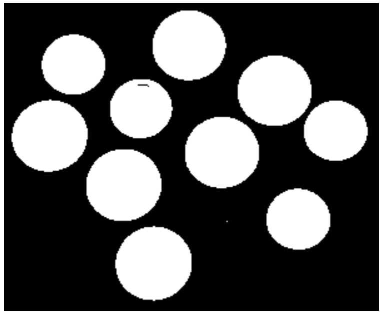
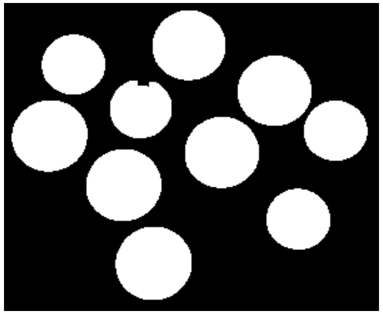
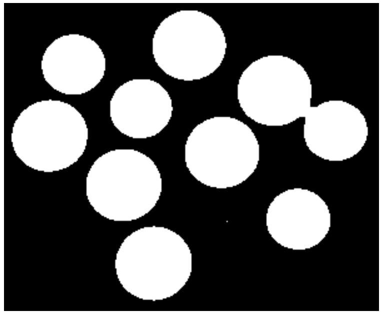


Imagen original


Tras proceso de apertura


Tras proceso de cierre

Operaciones morfológicas

- La toolbox de Matlab de procesamiento de imágenes contiene una función ***bwmorph***, que permite implementar una variedad de **funciones morfológicas** predefinidas: *open, close, erode, dilate*, etc…
- Por ejemplo: Una de las aplicaciones que se puede hacer es reducir los objetos de un circuito a una línea sin cambiar la estructura esencial de la imagen y conservando el numero de Euler. Este proceso se conoce como **esqueleto** (operación: ‘skel’):
  - BW3 = bwmorph(BW1,'skel',Inf); % inf=infinito.
  - figure, imshow(BW3); title('Esqueleto');

Operaciones basadas en objetos

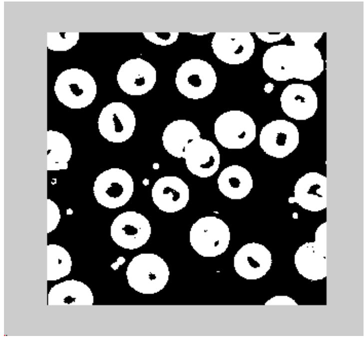
- Un objeto es un conjunto de píxeles a “1” conectados.
- **Entorno de vecindad/vecindario de conectividad**: la existencia de un objeto viene dada por la convención tomada para considerar píxeles conectados: pueden ser 4 vecinos conectados u 8 vecinos conectados.
- Posibles funciones para objetos: se ejecutan sobre todos los objetos de la imagen a la vez o sobre algunos de ellos. Pueden ser operaciones sobre ellos, cálculo de características, etc.:
  - Cálculo del perímetro
  - Llenado binario por crecimiento
  - Etiquetado de componentes conectados
  - Selección de objetos

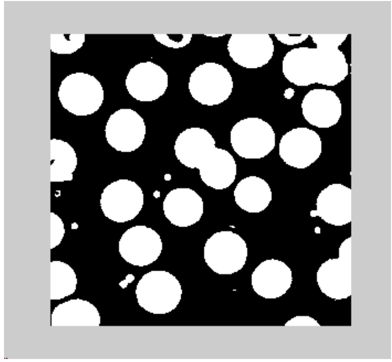
Cálculo del perímetro

- La función ***bwperim*** determina los píxeles que forman el **perímetro** de los objetos de una imagen binaria. Se puede utilizar tanto un vecindario de conectividad 4 como de 8. Por ejemplo:
  - BW1=imread('circbw.tif');
  - BW2=bwperim(BW1,4);
  - BW3=bwperim(BW1,8);
  - figure, imshow(BW1);
  - figure, imshow(BW2);
  - figure, imshow(BW3);

Llenado binario

- La función ***bwfill*** realiza una operación de **llenado** en imágenes binarias: rellena los huecos que haya dentro de los objetos. Se puede especificar un píxel de fondo como punto de inicio y la función cambia los píxeles de fondo conectados a píxeles de objetos, parando cuando alcanza los bordes del objeto.
  - BW4 = im2bw(imread('blood1.tif')); %im2bw convierte la imagen leída a binaria
  - BW4=~BW4; % Invierte la imagen. Ejecutar solo si es necesario.
  - BW5 = bwfill(BW4,'holes');
  - figure, imshow(BW4), figure, imshow(BW5)





Etiquetado y selección de objetos

- La función ***bwselect*** selecciona uno o varios objetos individuales en una imagen binaria.
- La función ***bwlabel***, etiqueta los objetos con diferentes valores, uno por cada objeto.
  - BW1 = imread(‘text.tif'); figure, imshow(BW1);
  - c = [50 90 144]; %Posición fila
  - r = [85 197 247]; %Posición columna
  - BW2 = bwselect(BW1,c,r,4); figure, imshow(BW2);
  - BW3 = bwlabel(BW2,4);
  - numero_objetos = max(max(BW3)) %número de objetos igual a etiqueta mayor
  - map=[0 0 0; jet(numero_objetos)];
  - figure,imshow((BW3+1),map,'InitialMagnification','fit');

Medidas de características

- ***bweuler*** devuelve el número de objetos en una imagen menos el número de agujeros.
- ***bwarea*** devuelve una estimación del área total de los objetos binarios de una imagen.
- Ejemplo: calcular el % de incremento de área de una imagen cuando se hace una dilatación.
  - BW1=imread('circbw.tif');
  - SE=ones(5);
  - BW2=imdilate(BW1,SE);
  - figure, imshow(BW1); figure, imshow(BW2);
  - increase=(bwarea(BW2)-bwarea(BW1))/bwarea(BW1)

Medidas de características

- ***imfeature / regionprops***
- Las medidas que pueden obtenerse son:

'Area'	'ConvexHull'	'EulerNumber'
'Centroid'	'ConvexImage'	'Extrema'
'BoundingBox'	'ConvexArea'	'EquivDiameter'
'MajorAxisLength'	'Image'	'Solidity'
'MinorAxisLength'	'FilledImage'	'Extent'
'Orientation'	'FilledArea'	'PixelList'
'Eccentricity'		

Ejemplo: Medidas de características

- BW1 = im2bw(imread('coins.png'),0.35); %im2bw convierte la imagen leída a binaria.
- figure, imshow(BW1);
- L = bwlabel(BW1, 4); %Imagen etiquetada.
- numero_objetos = max(max(L))   %número de objetos igual a etiqueta mayor
- map=[0 0 0; jet(numero_objetos)];
- figure, imshow(L+1,map,'InitialMagnification','fit'); title('fondo en negro, objetos en colores');
- objects = regionprops(L, 'area','EulerNumber');
- objects(:).Area
- objects(:).EulerNumber

Ejemplo: Medidas de características

- BW = im2bw(imread('coins.png'),0.35); % im2bw convierte la imagen leída a binaria.
- s = regionprops(BW, 'all');
- centroids = cat(1, s.Centroid); % concatena vectores en una matriz
- imshow(BW)
- hold on
- plot(centroids(:,1), centroids(:,2), 'b*')
- hold off