

# Procesamiento de imagen usando Teoría Fuzzy

**Edurne Barrenechea**

edurne.barrenecha@unavarra.es

**Giara Research group**

<http://giara.unavarra.es/>

**Universidad Pública de Navarra**

<http://www.unavarra.es/>



**EVI A 2016**

Asociación Española para la Inteligencia Artificial (**AEPIA**)

# [ Índice ]

---

- ✓ Segmentación de imágenes
  - ✓ Algoritmos de umbralización FSS
  - ✓ Algoritmos de umbralización IVFSS
  - ✓ Algoritmos de detección de bordes
- ✓ Reducción de imágenes

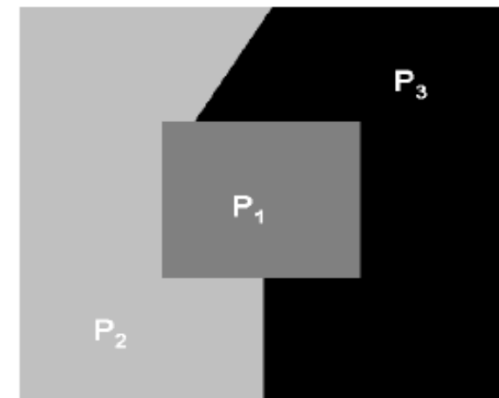
# [ Segmentación de imágenes ]

- “Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the success or failure of computerized analysis procedures”

(González & Woods, *Digital Image Processing*)

- La división de una imagen en regiones se denota **SEGMENTACIÓN**.
- La **segmentación** de imágenes digitales es el proceso de dividir una imagen en *partes*, *regiones* o *clases* tal que cada una de ellas posee propiedades o atributos concretos:

- $U_{i=1}^n P_i = \text{toda la imagen}$
- $P_j \cap P_i = \emptyset \quad i \neq j$



# [ Segmentación: niveles de gris ]

---

- Detección e identificación de objetos:
  - Separación de los objetos del fondo según el nivel de gris.
  - Reconocimiento de objetos por su forma.



# [ Segmentación: color ]

---

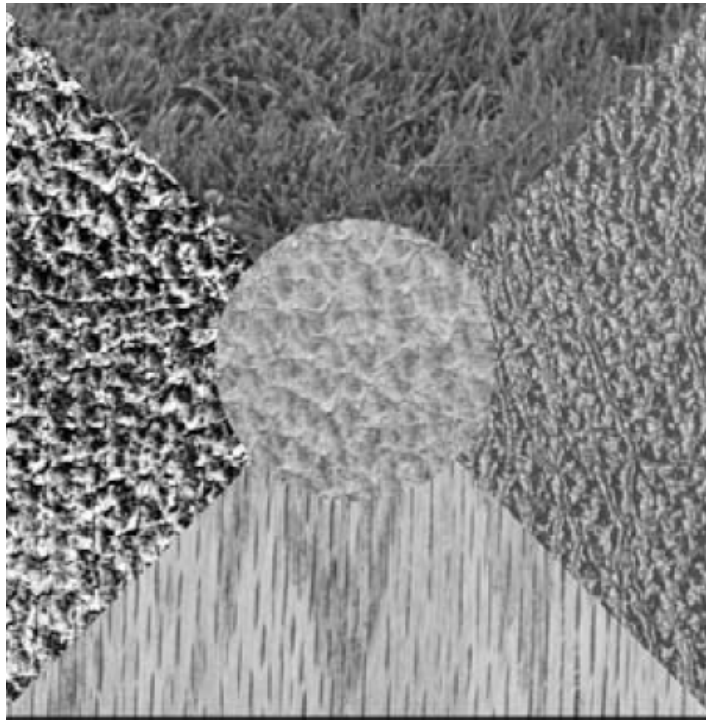
- Detección e identificación de objetos:
  - Identificación de regiones por su color.
  - Algunos objetos estarán formados por varias regiones.



# [Segmentación: textura]

---

- Detección e identificación de objetos:
  - Los niveles de gris de las regiones no son uniformes. Se requieren características más complejas.



# [ Segmentación de imágenes ]

---

- Existen principalmente 3 técnicas de segmentación:
  1. Segmentación basada en umbralización (**Thresholding**)
  2. Técnicas basadas en agrupamiento de píxeles en regiones (**Region-based** como Growing, Splitting and Merging)
  3. Técnicas basadas en detección de bordes (**Edge Detection**)
  
- La técnicas de **Thresholding y Region-based** tienen un enfoque de clasificación de píxeles, donde:
  - Píxeles de una misma región deben ser similares
  - Píxeles de regiones distintas deben ser no-similares
  - Las regiones resultantes deben tener un cierto significado para el procesamiento posterior
  - La similitud entre píxeles puede venir dada por: niveles de gris, colores, texturas, proximidad, etc.

# [ Segmentación de imágenes ]

- La técnicas basadas en **Edge Detection**:
  - Identificación de los píxeles que configuran la frontera de los objetos
  - Aislamiento de los objetos del resto de la imagen, de esta forma, queda definida la forma del objeto y por tanto los píxeles que lo integran





# [ Umbralización ]

- La **umbralización** permite convertir una imagen de niveles de gris en una binaria.
- Técnica útil si el fondo y los objetos son uniformes.

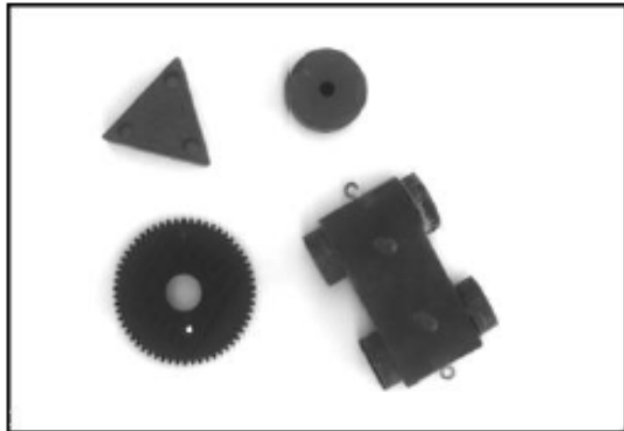
Objetos claros sobre fondo oscuro  $\longrightarrow g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq T \\ 0 & \text{en otro caso} \end{cases}$

Objetos oscuros sobre fondo claro  $\longrightarrow g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \leq T \\ 0 & \text{en otro caso} \end{cases}$

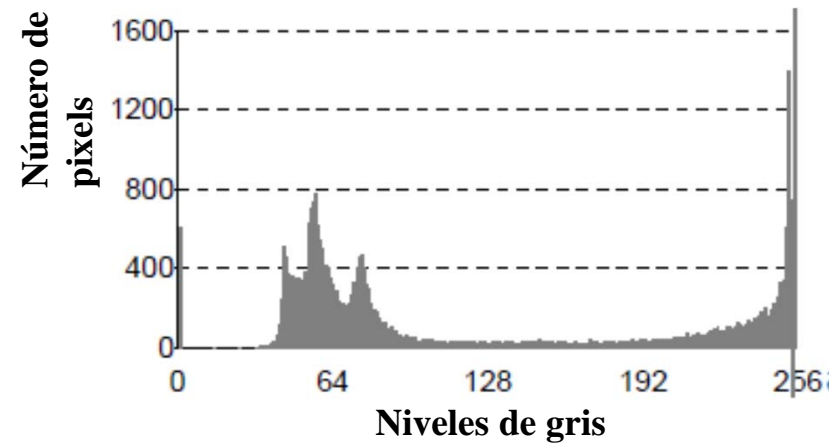
Objetos en un intervalo de grises  $\longrightarrow g(x, y) = \begin{cases} 1 & \text{si } T_{\min} \leq f(x, y) \leq T_{\max} \\ 0 & \text{en otro caso} \end{cases}$

# [ Umbralización ]

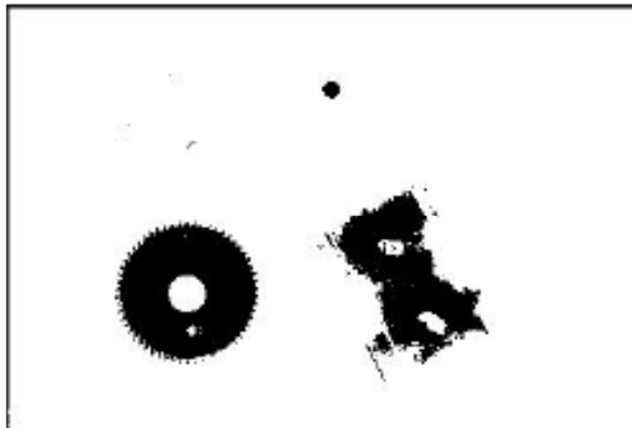
Imagen Original



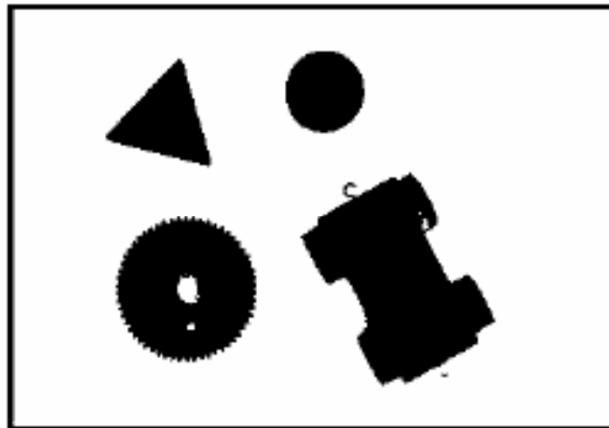
Histograma



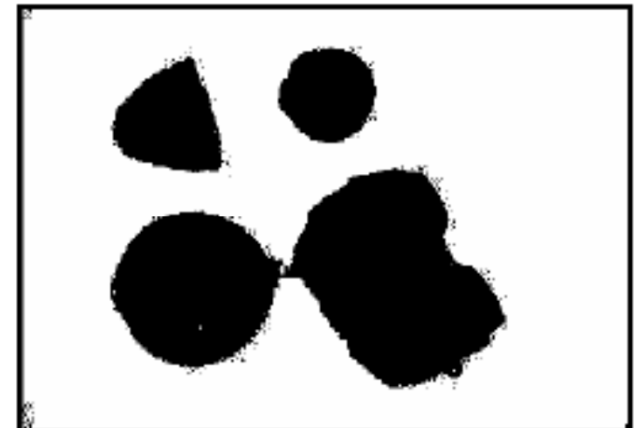
t = 61



t = 132



t = 250



# [ Umbralizacion ]

■ La elección de un *umbral* es complejo debido fundamentalmente a:

1. Un histograma no determina de manera única a una imagen;
2. La presencia de ruido en las imágenes;
3. La ambigüedad en el predicado (propiedad) que deben satisfacer los pixels de cada objeto de la imagen, debido al solapamiento en el histograma asociado a la imagen

***Por todo ello, una buena herramienta es la Teoría Fuzzy***

**Definition.** A fuzzy set  $A$  on a finite universe  $U$  is a mapping  $U \rightarrow [0, 1]$ .

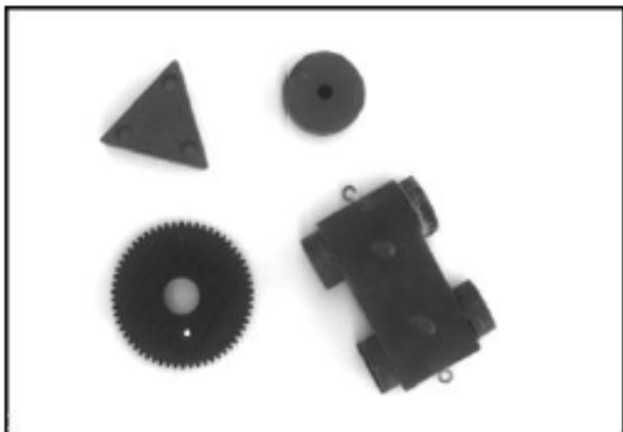
$$A = \{(u_i, \mu_A(u_i)) \mid u_i \in U\} \in FSSs(U)$$

$$A = \{(u_i, A(u_i)) \mid u_i \in U\} \in FSSs(U)$$

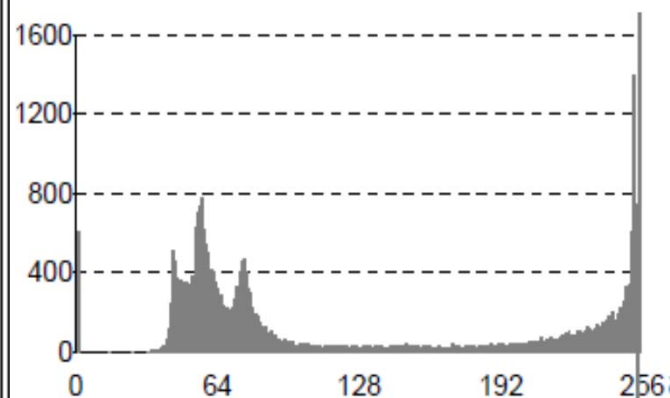
# Algoritmo Umbralización

- **Medidas de Similitud.** El algoritmo está basado en la construcción de  $L$  conjuntos fuzzy asociados a una imagen  $Q$  y elegir como umbral el valor de  $t$  tal que  $Q_t$  mejor representa a la binarización de la imagen original. Cada  $Q_t$  tiene dos clases, clase del Objeto,  $C_o(t)$  y, clase del fondo  $C_b(t)$ :
  - los píxeles con intensidad menor o igual que  $t$  pertenecen a  $C_o(t)$  y
  - los píxeles con intensidad mayor que  $t$  a  $C_b(t)$  (o viceversa).

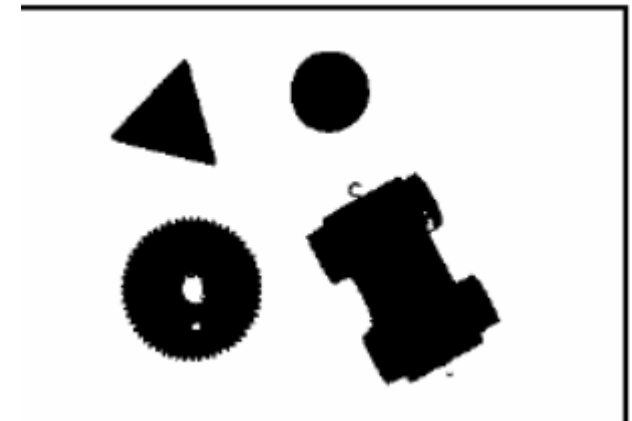
Imagen Original



Histograma



Binaria con  $t = 132$



# [ Algoritmo de umbralización ]

## ALGORITHM

**Input:** una imagen  $Q$  *en escala de grises*

**Output:** un umbral  $t$  para binarizar la imagen

(A) Construir  $L$  conjuntos fuzzy,  $Q_t$ , con  $t \in \{0, 1, \dots, L - 1\}$  asociados a la imagen  $Q$ , mediante *REFs*.

(B) Calcular la similitud de cada  $Q_t$  con el conjunto  $\tilde{I}$ .

(C) Tomar como **mejor umbral** el valor de  $t$  asociado con el conjunto  $Q_t$  de mayor similitud con el conjunto  $\tilde{I}$ .

# [ Algoritmo de umbralización: REFs ]

**Definición.** Una función  $REF: [0,1]^2 \rightarrow [0,1]$  se llama función de equivalencia restringida si cumple las siguientes condiciones:

- (1)  $REF(x, y) = REF(y, x) \forall x, y \in [0,1]$ ;
- (2)  $REF(x, y) = 1$  si solo si  $x = y$ ;
- (3)  $REF(x, y) = 0$  si solo si  $x = 0$  e  $y = 1$  o  $x = 1$  e  $y = 0$ ;
- (4)  $REF(x, y) = REF(n(x), n(y)) \forall x, y \in [0,1]$ ; siendo  $n$  una negación fuerte.
- (5) Para todo  $x, y, z \in [0,1]$   
Si  $x \leq y \leq z$  entonces  $REF(x, y) \geq REF(x, z)$  y  $REF(y, z) \geq REF(x, z)$ ;

**Definición.** Un automorfismo en el intervalo unidad es una función  $\varphi: [0,1] \rightarrow [0,1]$  que es continua, estrictamente creciente tal que  $\varphi(0) = 0$  y  $\varphi(1) = 1$ .

**Método de construcción.** Sean  $\varphi_1, \varphi_2$  dos automorfismos en  $[0,1]$ , entonces

$$REF(x, y) = \varphi_1^{-1}(1 - |\varphi_2(x) - \varphi_2(y)|) \text{ con } n(x) = \varphi_2^{-1}(1 - \varphi_2(x))$$

es una función de equivalencia restringida.

**Proposición.**  $\varphi^\circ REF$  es también una función de equivalencia restringida.

# [ Algoritmo de umbralización: REFs ]

## Ejemplos:

### Método de construcción.

- Sean  $\varphi_1(x) = \varphi_2(x) = x$  entonces  $REF(x, y) = 1 - |x - y|$ .
- Sean  $\varphi_1(x) = 1 - (1 - x)^{\frac{1}{d}}$  con  $d \geq 1$ ,  $\varphi_2(x) = x$  entonces  $REF(x, y) = 1 - |x - y|^d$ .

**Proposición.** Sea  $REF(x, y) = 1 - |x - y|$ .

- Sea  $\varphi(x) = x^4$  entonces  $\varphi \circ REF = (1 - |x - y|)^4$  es una REF.
- Sea  $\varphi(x) = x^{\frac{1}{4}}$  entonces  $\varphi \circ REF = (1 - |x - y|)^{\frac{1}{4}}$  es una REF.

# [ Algoritmo de umbralización: similitud ]

**Definición.** Sea  $M: [0,1]^n \rightarrow [0,1]$  tal que

$$M(x_1, \dots, x_n) = 0 \text{ si y solo si } x_1 = \dots = x_n = 0$$

$$M(x_1, \dots, x_n) = 1 \text{ si y solo si } x_1 = \dots = x_n = 1$$

$M$  es creciente.

Sea  $REF$  una función de equivalencia restringida. Bajo estas condiciones

$$SM: F(X) \times F(X) \rightarrow [0,1] \text{ dado por } SM(A, B) = \prod_{i=1}^n REF(\mu_A(x_i), \mu_B(x_i))$$

Satisface las siguientes condiciones:

- (1)  $SM(A, B) = SM(B, A) \forall A, B \in F(X)$ ;
- (2)  $SM(A, A_c) = 0$  si solo si  $A$  no es fuzzy;
- (3)  $SM(A, B) = 1$  si solo si solo  $A = B$ ;
- (4) Si  $A \leq B \leq C$  entonces,  $SM(A, B) \geq SM(A, C)$  y  $SM(C, B) \geq SM(C, A)$ .



# [ Algoritmo de umbralización ]

(A) Construir  $L$  conjuntos fuzzy  $Q_t$  con  $t \in \{0, 1, \dots, L - 1\}$  asociados a una imagen  $Q$ .

Sea  $Q$  una imagen en  $L$  niveles de gris y  $t$  un nivel de gris tal que  $0 \leq t \leq L - 1$ .

Construimos un conjunto fuzzy

$$Q_t = \{(q, \mu_{Q_t}(q)) | q \in \{0, 1, \dots, l - 1\}\} \text{ tal que}$$

$$\mu_{Q_t}(q) = \begin{cases} \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_b(t)}{L-1} \right) \right), & q \leq t \\ \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_o(t)}{L-1} \right) \right), & q > t \end{cases}$$

$$\text{con } m_b(t) = \frac{\sum_{q=0}^t q \cdot h(q)}{\sum_{q=0}^t h(q)} \quad m_o(t) = \frac{\sum_{q=t+1}^{L-1} q \cdot h(q)}{\sum_{q=t+1}^{L-1} h(q)}$$

# [ Algoritmo de umbralización ]

(A) Construir  $L$  conjuntos fuzzy  $Q_t$  con  $t \in \{0, 1, \dots, L-1\}$  asociados a una imagen  $Q$ . Ejemplos.

$$\mu_{Q_t}(q) = \begin{cases} \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_b(t)}{L-1} \right) \right), & q \leq t \\ \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_o(t)}{L-1} \right) \right) & q > t \end{cases}$$

$$\varphi(x) = x$$

$$REF(x, y) = 1 - |x - y|$$

$$\mu_Q(q) = \begin{cases} 1 - \left| \frac{q - m_b(t)}{L-1} \right|, & q \leq t \\ 1 - \left| \frac{q - m_o(t)}{L-1} \right|, & q > t \end{cases}$$

$$\varphi(x) = x$$

$$REF(x, y) = 1 - |x - y|^2$$

$$\mu_Q(q) = \begin{cases} 1 - \left| \frac{q - m_b(t)}{L-1} \right|^2, & q \leq t \\ 1 - \left| \frac{q - m_o(t)}{L-1} \right|^2, & q > t \end{cases}$$

$$\varphi(x) = x$$

$$REF(x, y) = 1 - |x - y|^{0.5}$$

$$\mu_Q(q) = \begin{cases} 1 - \left| \frac{q - m_b(t)}{L-1} \right|^{0.5}, & q \leq t \\ 1 - \left| \frac{q - m_o(t)}{L-1} \right|^{0.5}, & q > t \end{cases}$$

$$\varphi(x) = x$$

$$REF(x, y) = (1 - |x - y|)^2$$

$$\mu_Q(q) = \begin{cases} \left( 1 - \left| \frac{q - m_b(t)}{L-1} \right| \right)^2, & q \leq t \\ \left( 1 - \left| \frac{q - m_o(t)}{L-1} \right| \right)^2, & q > t \end{cases}$$

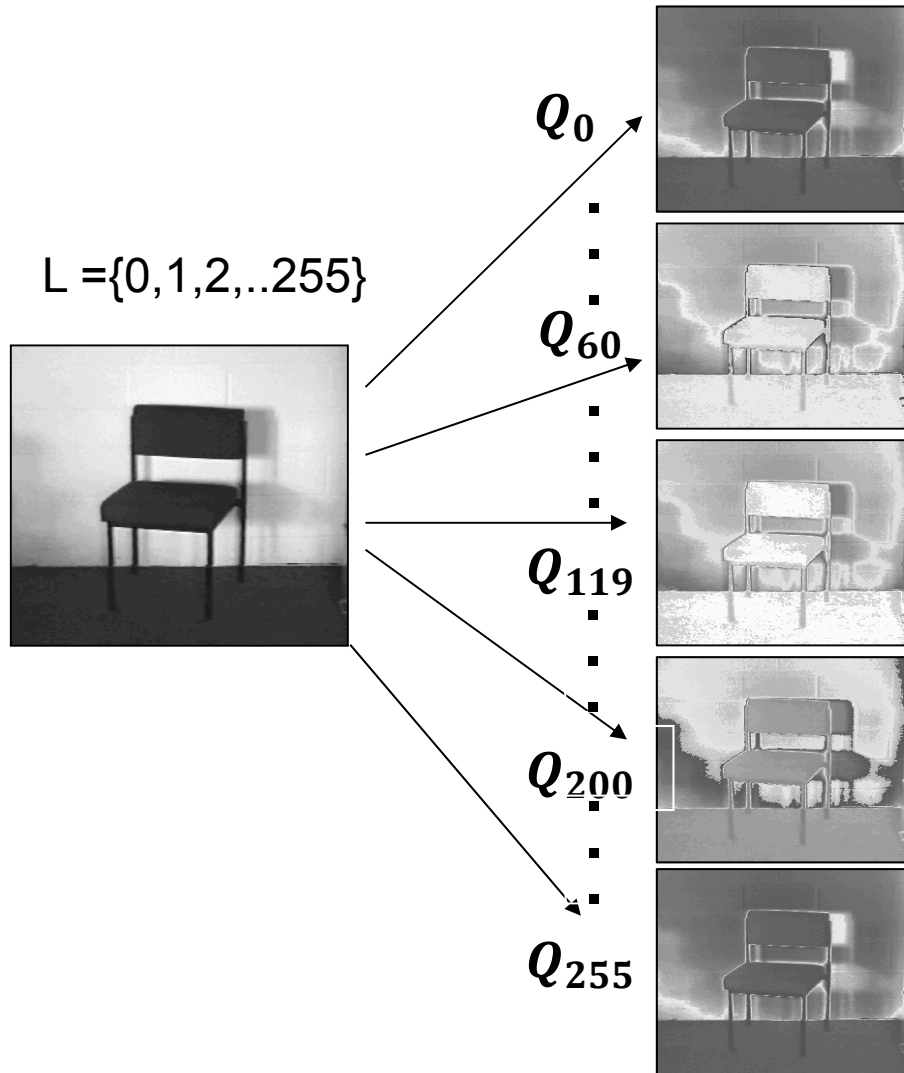
$$\varphi(x) = x$$

$$REF(x, y) = (1 - |x - y|)^{0.5}$$

$$\mu_Q(q) = \begin{cases} \left( 1 - \left| \frac{q - m_b(t)}{L-1} \right| \right)^{0.5}, & q \leq t \\ \left( 1 - \left| \frac{q - m_o(t)}{L-1} \right| \right)^{0.5}, & q > t \end{cases}$$

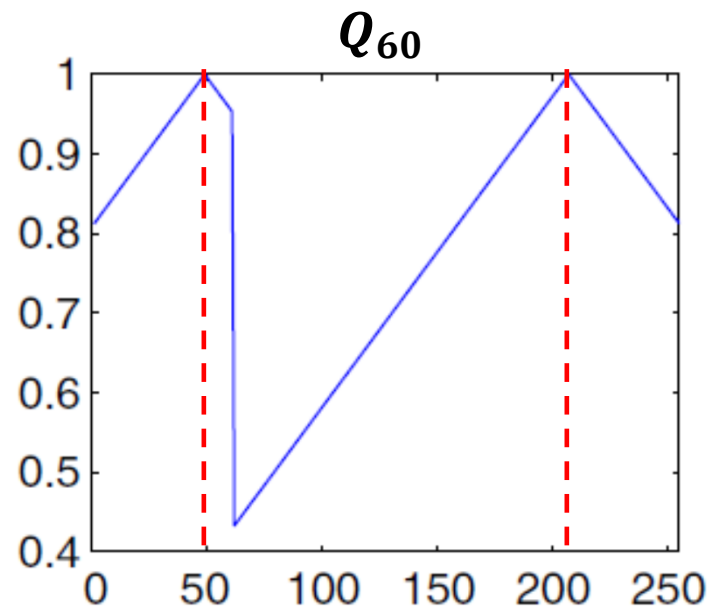
# [ Algoritmo de umbralización ]

(A) Construir  $L$  conjuntos fuzzy  $Q_t$  con  $t \in \{0, 1, \dots, L - 1\}$  asociados a una imagen  $Q$ .



$$\varphi(x) = x; REF(x, y) = 1 - |x - y|$$

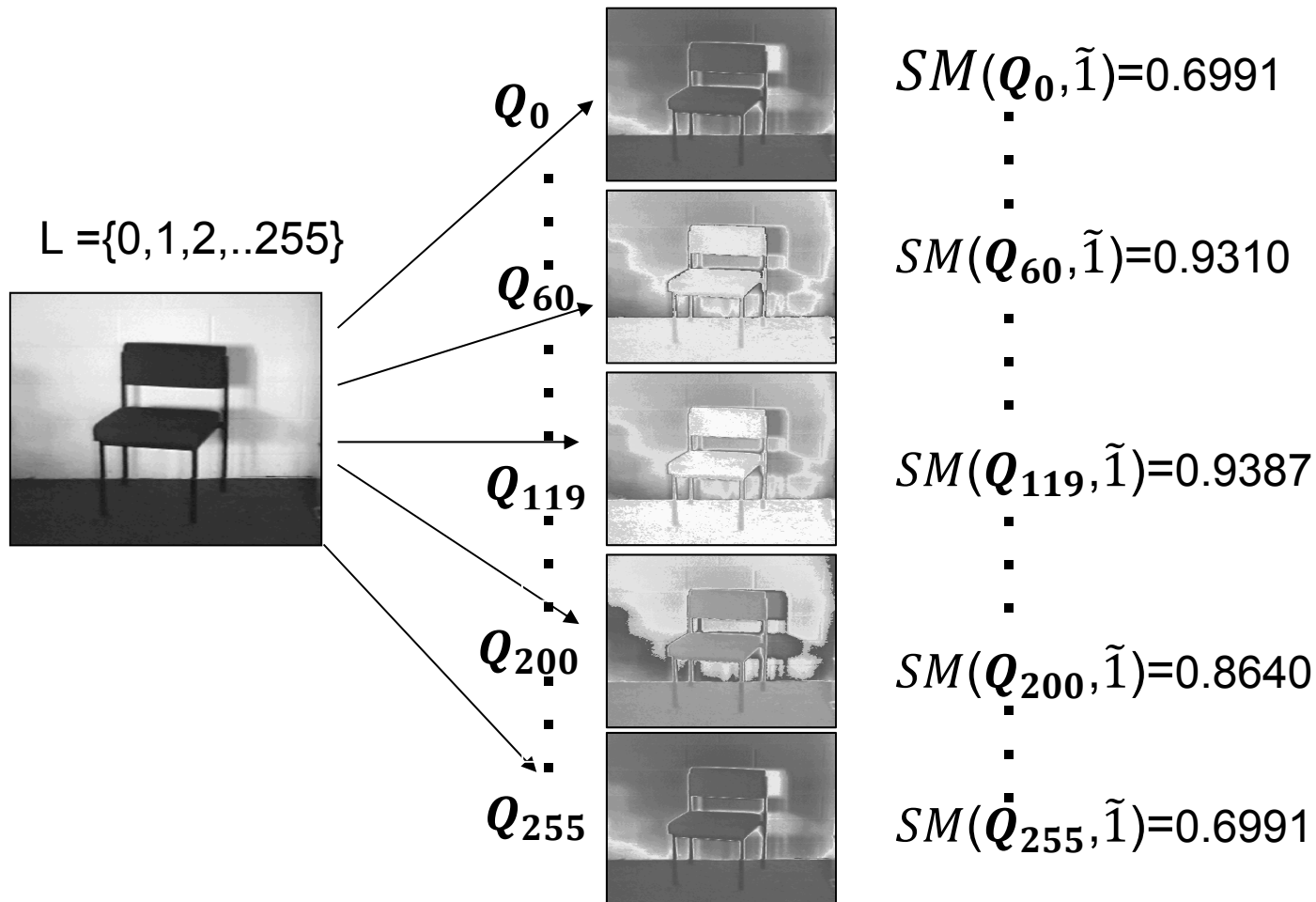
$$\mu_{Q_t}(q) = \begin{cases} 1 - \left| \frac{q - m_b(t)}{L - 1} \right|, & q \leq t \\ 1 - \left| \frac{q - m_o(t)}{L - 1} \right|, & q > t \end{cases}$$



# [ Algoritmo de umbralización ]

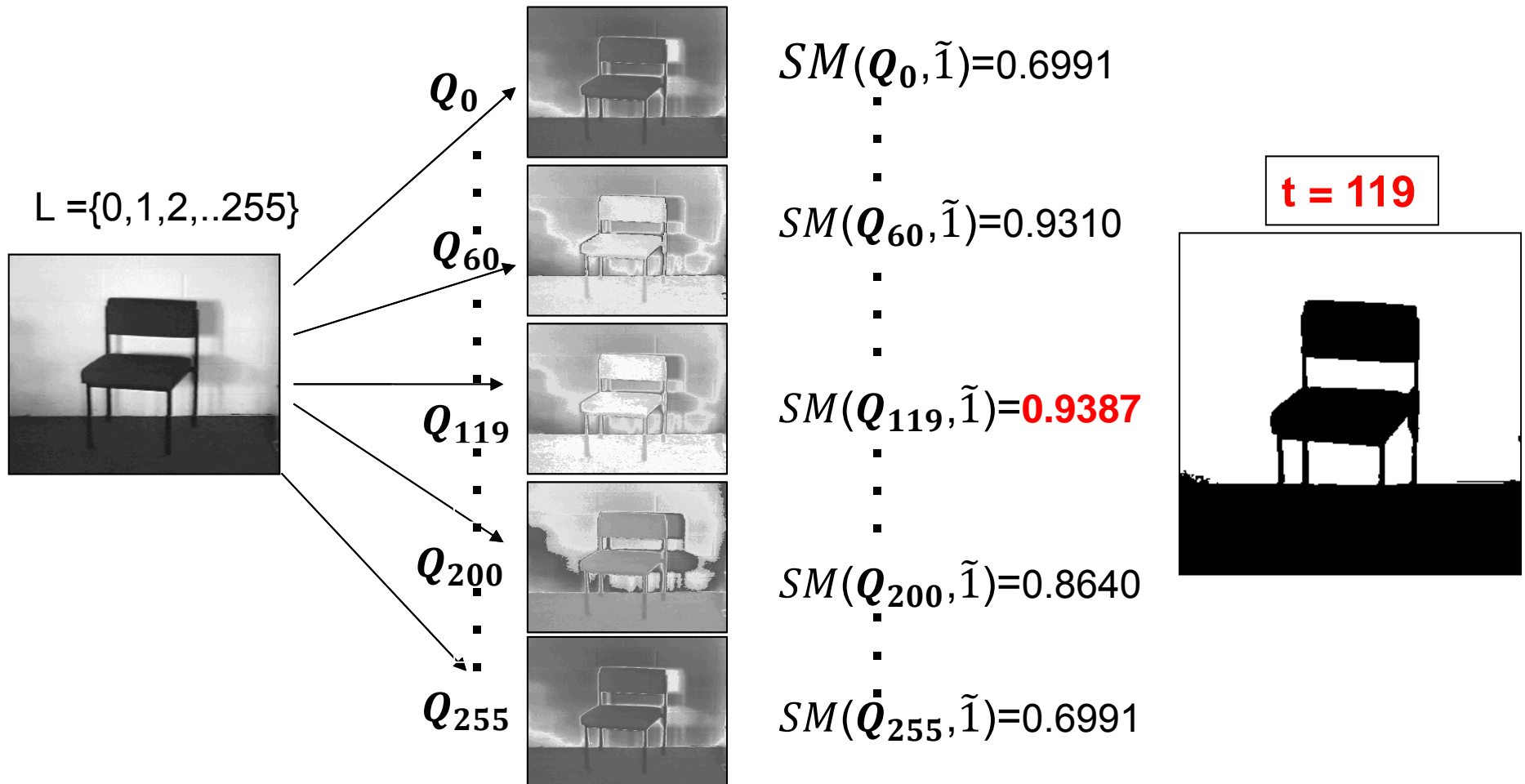
(B) Calcular la similitud de cada  $Q_t$  con el conjunto  $\tilde{1}$ .

$$SM(Q_t, \tilde{1}) = M_{q=0}^{L-1} REF_2(1, \mu_{Q_t}(q)); \quad M \text{ arithmetic mean and } REF_2(x, y) = 1 - |x - y|^2$$

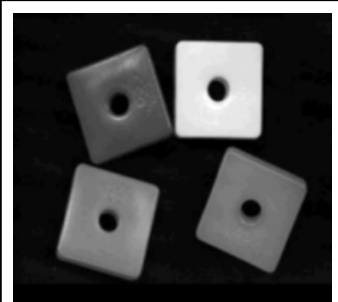
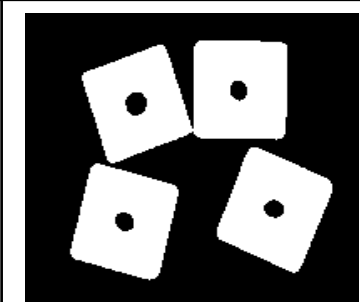
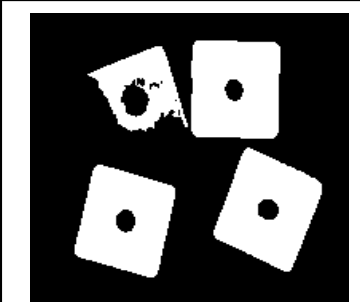
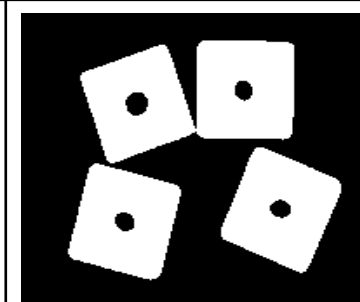
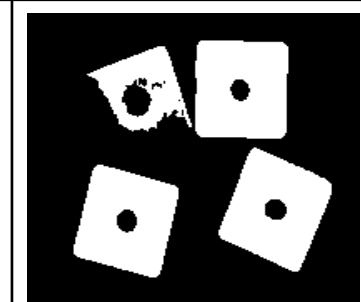


# [ Algoritmo de umbralización ]

(C) Tomar como *mejor umbral* el valor de  $t$  asociado con el conjunto  $Q_t$  de mayor similitud con el conjunto  $\tilde{I}$ .



# [ Algoritmo de umbralización ]

				
<p><b>Original</b></p>	$REF(x, y) = (1 -  x - y )^2$ $REF_2(x, y) = 1 -  x - y ^2$ <p><math>M</math> media aritmética</p> <p>Mejor umbral</p> <p><b>t = 49</b></p>	$REF(x, y) = 1 -  x - y ^2$ $REF_2(x, y) = 1 -  x - y ^2$ <p><math>M</math> media aritmética</p> <p>Mejor umbral</p> <p><b>t = 79</b></p>	$REF(x, y) = 1 -  x - y $ $REF_2(x, y) = 1 -  x - y ^2$ <p><math>M</math> media aritmética</p> <p>Mejor umbral</p> <p><b>t = 48</b></p>	<p><b>Otsu</b></p> <p><b>t = 79</b></p>

# [ Algoritmo de umbralización ]

- Una versión del mismo algoritmo denotado como del “Área”  $Q_t$ :

(1) Select  $\varphi_1$  and  $\varphi_2$ .

(2) FOR  $t := 0$  TO  $L - 1$  DO

(2-1) Calculate the area:  $A(Q_t) =$

$$\sum_{q=0}^t h(q) \varphi_1^{-1} \left( 1 - \left| \varphi_2 \left( \frac{q}{L-1} \right) - \varphi_2 \left( \frac{m_b(t)}{L-1} \right) \right| \right) \\ + \sum_{q=t+1}^{L-1} h(q) \varphi_1^{-1} \left( 1 - \left| \varphi_2 \left( \frac{q}{L-1} \right) - \varphi_2 \left( \frac{m_o(t)}{L-1} \right) \right| \right).$$

ENDFOR

(3) Take as the best threshold the value of  $t$  associated with the set  $Q_t$  of greatest area.

- El método clásico de Otsu es un caso particular de dicho algoritmo.

# [ Algoritmo de umbralización ]

- Practicar con las siguientes imágenes y código

SegmentacionFS(*imagen, caso, si, d*)

donde **caso** toma los siguientes valores:

- 'A' indica  $f_i(x)=x$
- 'B' indica  $f_i(x)=x.^2$
- 'C' indica  $f_i(x)=x.^{0,5}$
- 'D' indica  $f_i(x)=1-(1-x).^2$
- 'E' indica  $f_i(x)=1-(1-x).^0,5$

donde **si** toma los valores 'A' y 'B' para representar las variantes de REF

con **d** exponente de la REF, tomamos valor =1

siendo imagen por ejemplo 'D:\Fuzzy\CursoVeranoAEPIA2016\Codigo\_Images\00.bmp'

caso='A' and si='A, d=1'

$$\varphi(x) = x$$

$$REF(x, y) = 1 - |x - y|$$

$$\mu_Q(q) = \begin{cases} 1 - \left| \frac{q - m_b(t)}{L-1} \right|, & q \leq t \\ 1 - \left| \frac{q - m_o(t)}{L-1} \right|, & q > t \end{cases}$$

SegmentacionFS('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo\_Images\00.bmp','A','A',1)

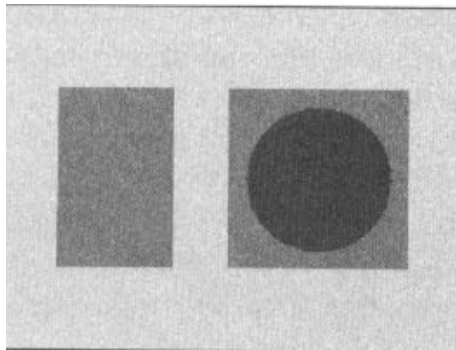
SegmentacionFS('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo\_Images\00.bmp','A','A',2)

SegmentacionFS('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo\_Images\00.bmp','A','A',0.5)



# [ Algoritmo de multithresholding ]

Imagen Original



Histograma

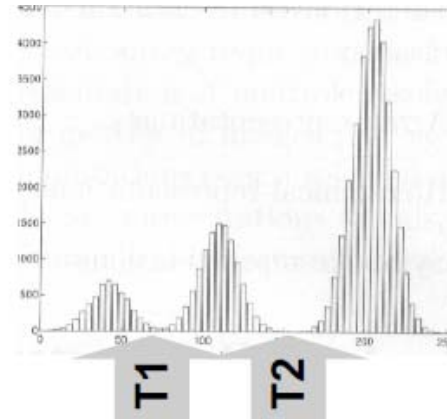
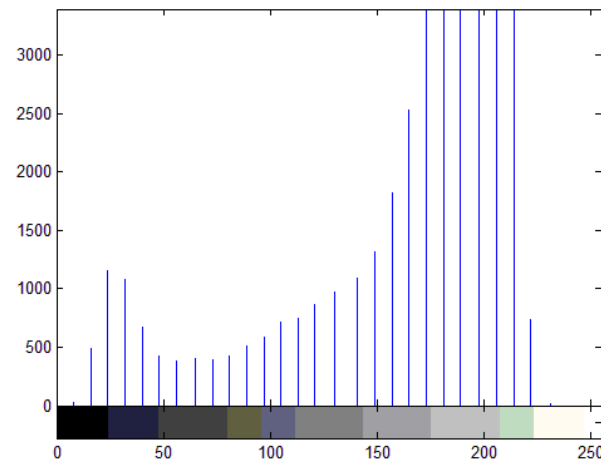


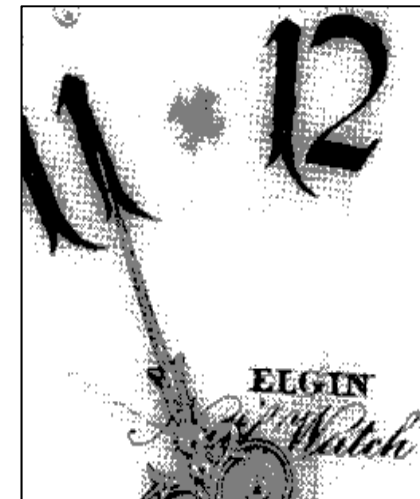
Imagen Original



Histograma



$t1 = 89$   $t2 = 165$



# [ Algoritmo de multithresholding ]

Supongamos que queremos dividir la imagen en  $k+1$  regiones, para ello necesitamos  $k$  umbrales.

- (A) Seleccionar el valor de  $K$ .
- (B) Seleccionar  $\varphi$  y  $REF$  para construir los conjuntos fuzzy.
- (C) Para todas las posibles combinaciones de  $K$  valores de intensidad  $\in \{0, 1, \dots, L-1\}$ .

1. Calcular

$$m_0(t) = \frac{\sum_{q=0}^{t_i} q \cdot h(q)}{\sum_{q=0}^{t_i} h(q)} \quad m_1(t) = \frac{\sum_{q=t_i+1}^{t_j} q \cdot h(q)}{\sum_{q=t_i+1}^{t_j} h(q)}, \dots, m_k(t) = \frac{\sum_{q=t_{k-1}+1}^{L-1} q \cdot h(q)}{\sum_{q=t_{k-1}+1}^{L-1} h(q)}$$

2. Construir el conjunto fuzzy asociado a los  $K$  umbrales:

$$Q_{t_i, t_j, \dots, t_k}(q) = \begin{cases} \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_0(t)}{L-1} \right) \right), & q \leq t_i \\ \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_1(t)}{L-1} \right) \right), & t_i < q \leq t_j \\ \dots \\ \varphi \left( REF \left( \frac{q}{L-1}, \frac{m_k(t)}{L-1} \right) \right), & q > t_k \end{cases}$$

(D) Calcular  $SM(Q_{t_i, t_j, \dots, t_k}, \tilde{1}) = M_{q=0}^{L-1} REF_2 \left( 1, \mu_{Q_{t_i, t_j, \dots, t_k}}(q) \right)$

(E) Tomar la combinación de  $k$  umbrales tales que se correspondan con el mayor valor de similitud calculado en (D).

# [ Algoritmo de multithresholding ]

---

- Practicar con las imágenes y el código

similaproxdos(*imagen*, *tabladosumbrales*)

donde *imagen* es cualquier imagen a segmentar en tres zonas  
*tabla* es tab12.bmp y es necesario indicar la ruta completa

**Ejemplo:**

```
similaproxdos('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\Segmentacion\00.bmp',  
              'D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\Segmentacion\FS\tab12.bmp')
```

# [ Generalización del algoritmo de umbralización: IVFSs ]

- **Definición.** un IVFS  $A$  sobre un universo de discurso finito  $U$  es un mapeo  $U \rightarrow L([0, 1])$ .

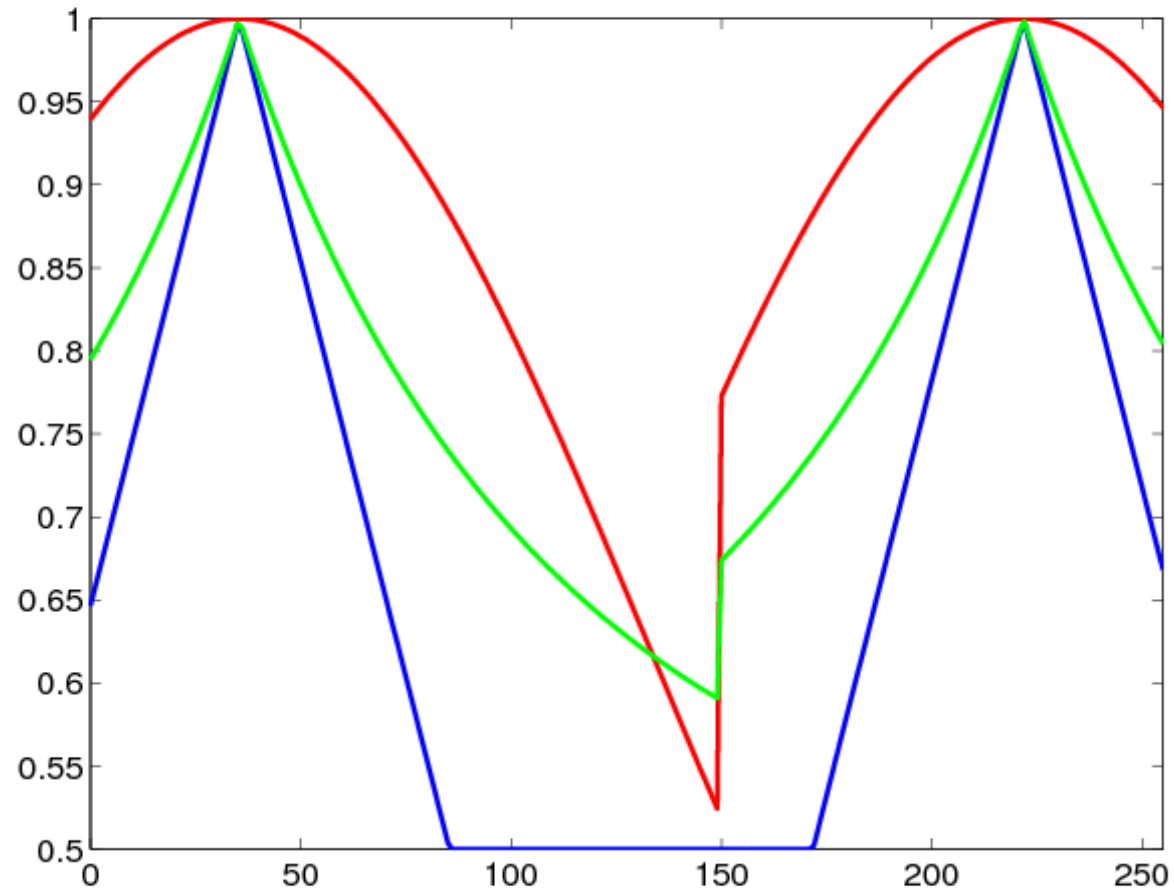
$$A = \{(u_i, A(u_i)) = [\underline{A}(u_i), \bar{A}(u_i)] \mid u_i \in U\} \in IVFSs(U)$$

$L([0, 1])$  es el conjunto de todos los subintervalos cerrados del intervalo unidad  $[0, 1]$ .

- Extendemos el algoritmo de umbralización fuzzy considerando que un expert puede elegir varios conjuntos difusos para representar una imagen  $Q^1, \dots, Q^k$ .
- A partir de estos conjuntos fuzzy construimos un IVFS tal que la longitud de cada intervalo de pertenencia represente la incertidumbre/desconocimiento del experto en elegir dichas funciones:
  - Si el experto está totalmente seguro de que un pixel pertenece al fondo (objecto), entonces la longitud del interval debe ser cero (caso FS).
  - Si el experto tiene la máxima duda de de si un pixel pertenece al fondo (objecto), entonces la longitud del interval debe ser la maxima posible.

# [ Generalización del algoritmo de umbralización: IVFSs ]

- Tres funciones de pertenencia diferentes para representar la imagen a partir de 3 REFs diferentes.



# [ Generalización del algoritmo de umbralización: IVFSs ]

- **Input:** una imagen en escala de grises  $Q$ .
- **Output:** un umbral  $t$  para binarizar la imagen

(A) Fijar un número  $k$ ;

(B) Para cada nivel de intensidad  $t \in \{0, 1, \dots, L - 1\}$

(B1) Asignar  $k$  conjuntos fuzzy,  $Q_t^1, \dots, Q_t^k$ , a la image  $Q$ ;

(B2) Construir un IVFS,  $\tilde{Q}_t$ , a partir de  $Q_t^1, \dots, Q_t^k$ ;

(C) Calcular la entropía intervalar de  $\tilde{Q}_t$ .

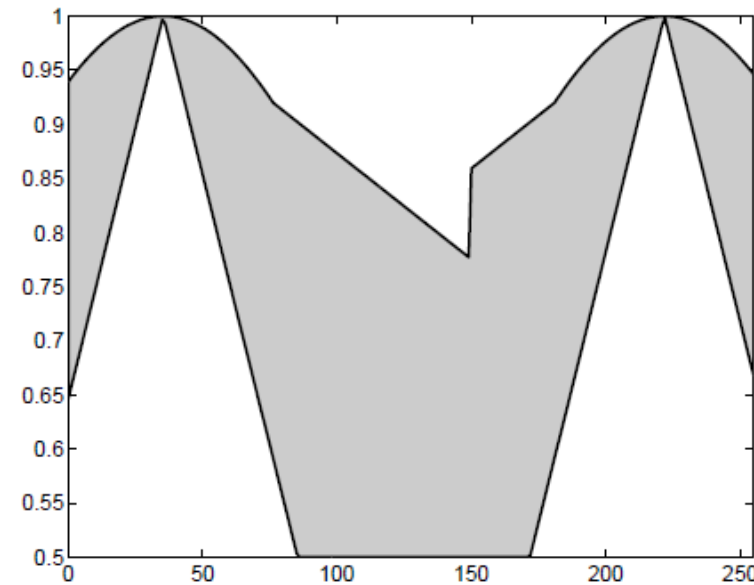
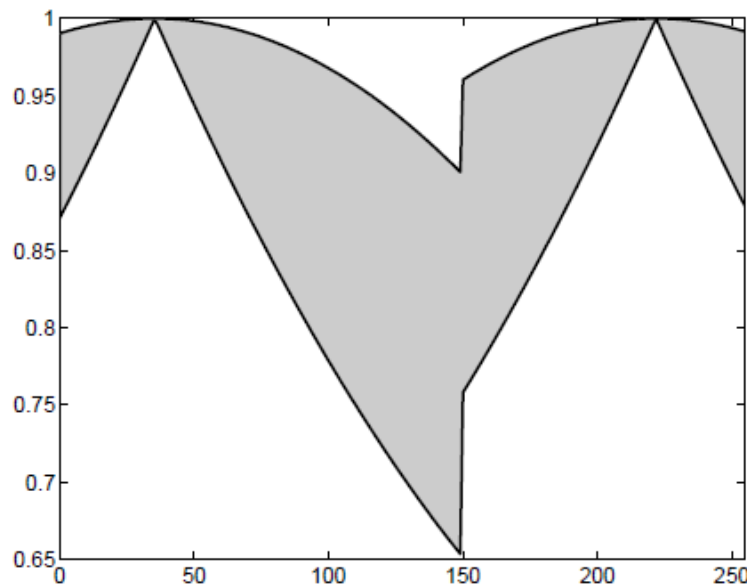
(D) Tomar como **mejor umbral** el valor de  $t$  asociado con el conjunto  $\tilde{Q}_t$  de menor entropía.

# [ Generalización del algoritmo de umbralización: IVFSs ]

- (B1) Asignar  $k$  conjuntos fuzzy,  $Q_t^1, \dots, Q_t^k$ , a la imagen  $Q$ ;
- (B2) Construir un IVFS,  $\tilde{Q}_t$ , a partir de (B1):  

$$\delta: FSS(U) \times \dots \times FSS(U) \rightarrow IVFSs(U)$$

$$\delta(Q_t^1, \dots, Q_t^k)(u) = [T(\mu_{Q_t^1}(u), \dots, \mu_{Q_t^k}(u)), S(\mu_{Q_t^1}(u), \dots, \mu_{Q_t^k}(u))]$$



# [ Generalización del algoritmo de umbralización: IVFSs ]

---

(C) Calcular la entropía de cada  $\tilde{Q}_t$ .

Índice de indeterminación de Sambuc:

$$E(\tilde{Q}_t) = \frac{1}{N} \sum_i^N \bar{\mu}_{\tilde{Q}}(u_i) - \underline{\mu}_{\tilde{Q}}(u_i)$$

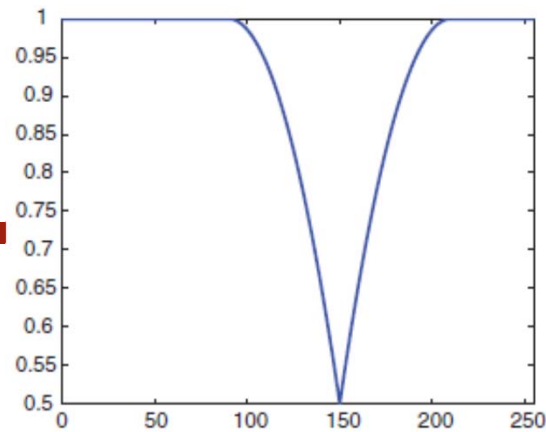
(D) Tomar como **mejor umbral** el valor de  $t$  asociado con el conjunto  $\tilde{Q}_t$  de menor entropía.



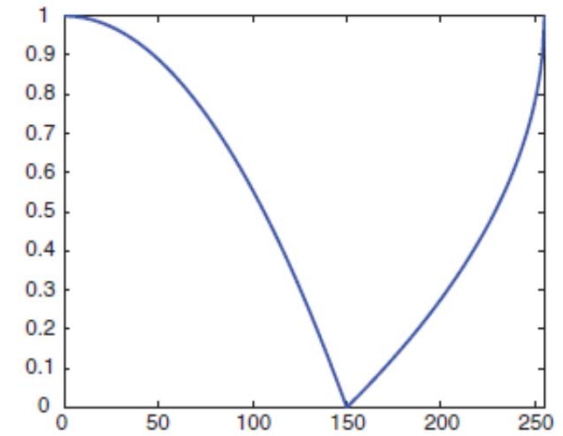


(a) S- and Z-functions  $t = 150$ ,  $a = t$ ,  $b = a + 60/2$ , and  $c = a + 60$ .

(b) Tizhoosh-based function,  $t = 150$ ,  $k1 = 2$ , and  $k2 = 1/2$ .



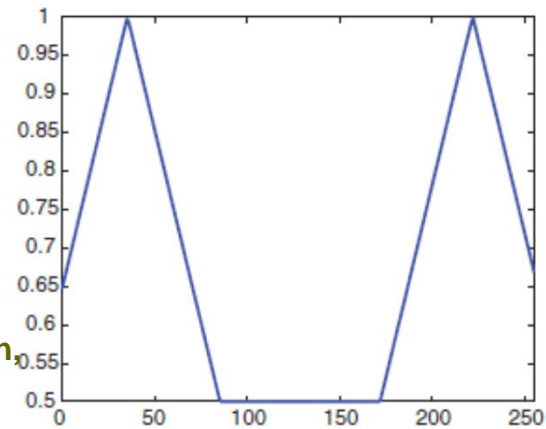
(a)



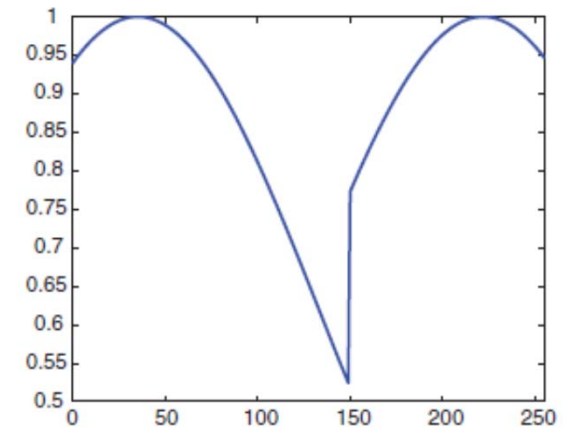
(b)

(c) Triangular-based function,  $t = 150$ , base width = 100.

(d) Normal distribution membership function,  $t = 150$ , std. dev. = 100.



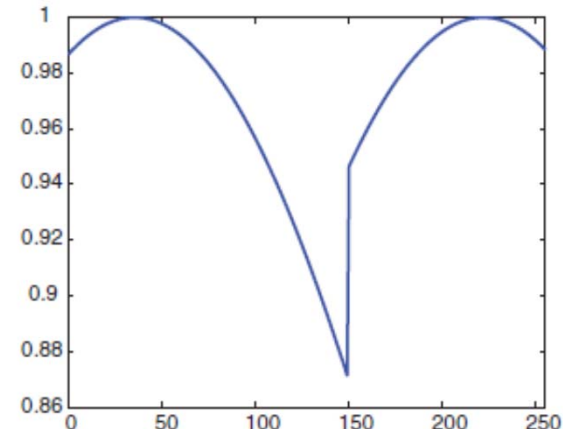
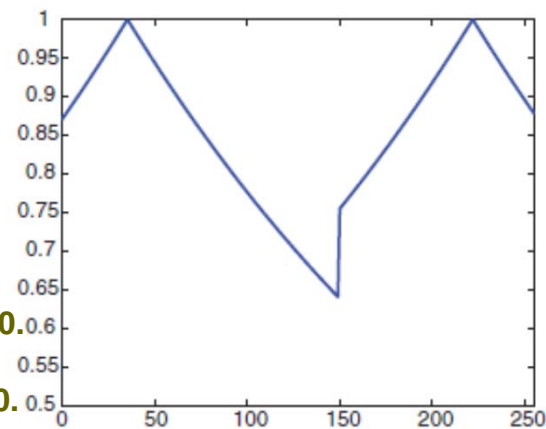
(c)



(d)

(e) Exponential membership function,  $t = 150$ .

(f) Logarithmic membership function,  $t = 150$ .



# [ Generalización del algoritmo de umbralización: IVFSs ]

---

- Practicar con las imágenes y el código:

```
IVFSC2algv2(imagen,'1')  
IVFSC2algv2(imagen,'1', '2', '4')
```

Indicando la ruta completa de la imagen, por ejemplo,

```
'D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\00.bmp'
```

y con los siguientes parámetros de números se escogen las diferentes funciones de pertenencia.

Ejemplo:

```
IVFSC2algv2('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\00.bmp','1')
```

```
IVFSC2algv2('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\00.bmp','1','2','4')
```

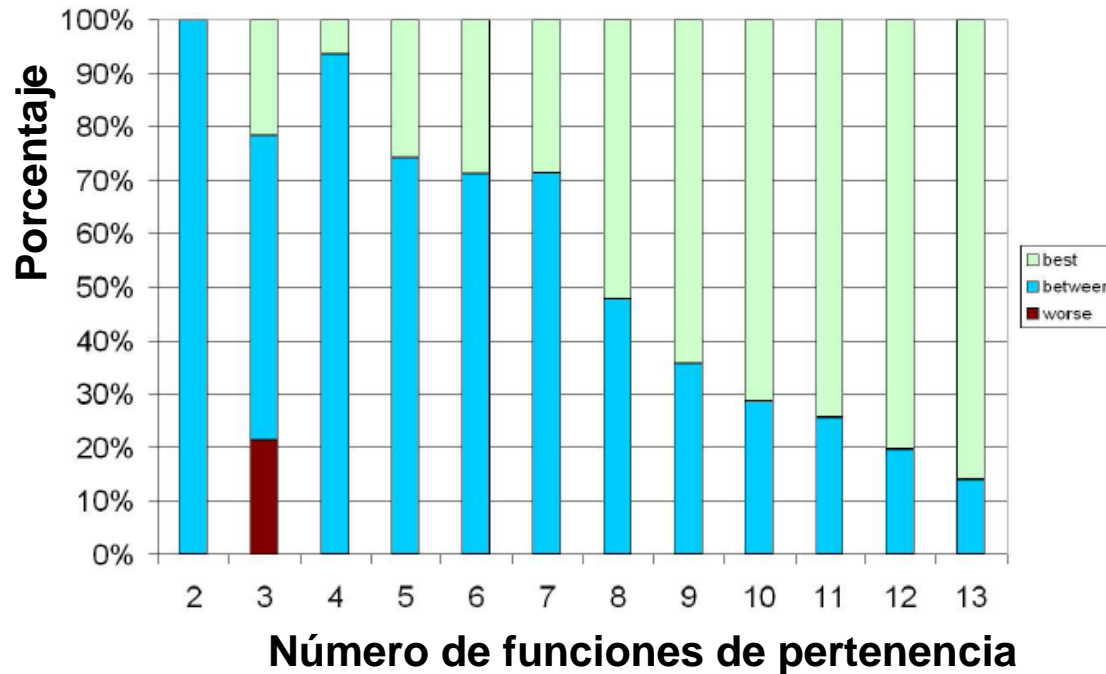
# [ Generalización del algoritmo de umbralización: IVFSs ]

---

## Experimento

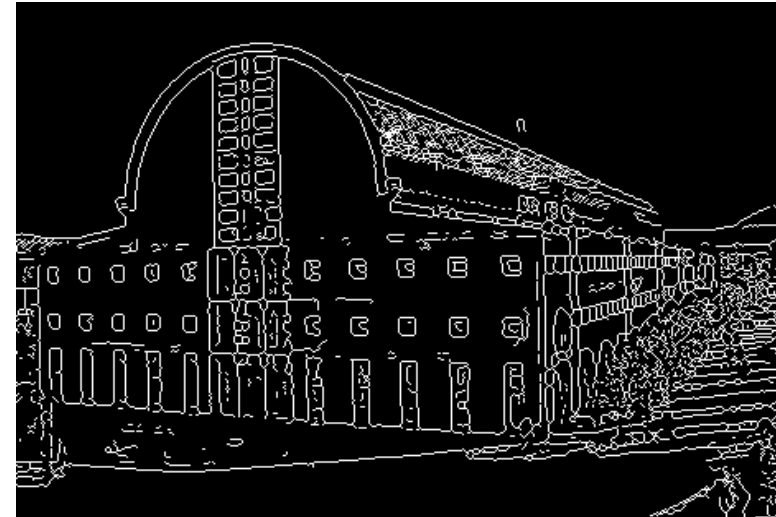
- Tomar  $K=14$  funciones de pertenencia diferentes;
- Siendo  $k$  el número de funciones elegidas, tomar  $k = 2, k = 3, \dots, k = 14$ , calculamos cada possible combinación de funciones de pertenencia ;
- $14!/((14-k)!k!)$  casos
- Para **cada combinación**, calculamos la solución con el primer algoritmo fuzzy (para cada función de pertenencia) y la solución con el algoritmo IVFSs.
- **Entropía**: tomamos la suma de las amplitudes de los intervalos.

# Generalización del algoritmo de umbralización: IVFSs



- % de casos en los que el algoritmo IVFS es **mejor** que el mejor algoritmo fuzzy, **peor** que el peor algoritmo fuzzy, y **entre**.
- Los resultados muestran que, en la mayoría de los casos, la solución obtenida con el algoritmo IVFS o es **similar a la mejor solución** obtenida algoritmo fuzzy o **está en el rango entre la peor y la mejor** solución obtenida algoritmo fuzzy.

# [ Algoritmos de detección de bordes ]



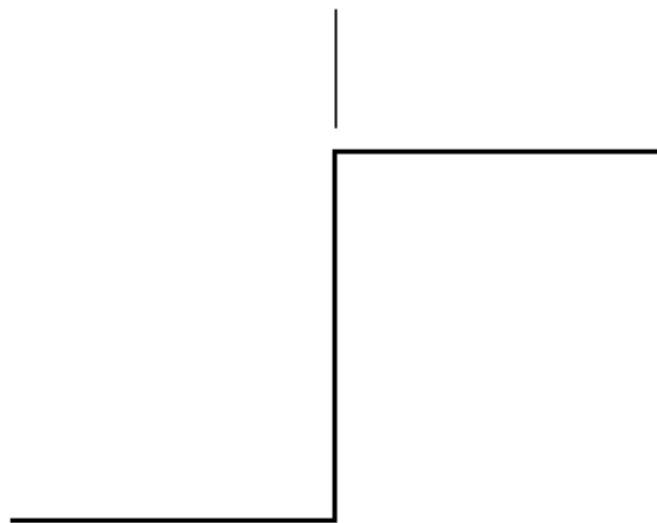
**Borde:** un salto de intensidad lo suficientemente grande entre la intensidad de un pixel y las intensidades de sus píxeles vecinos.



# [ Detección de bordes ]

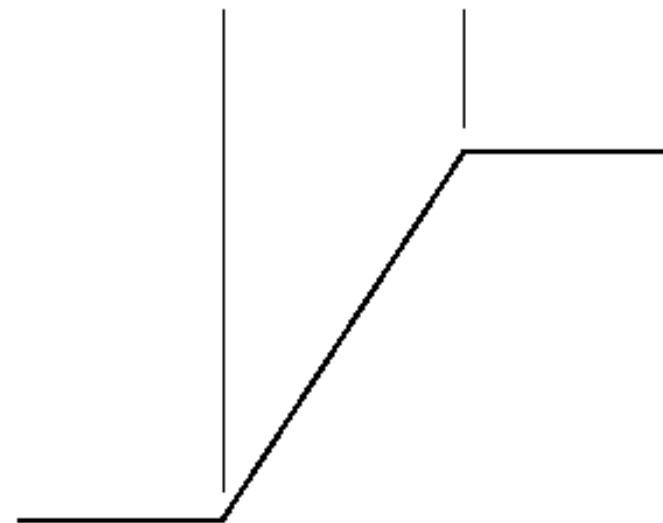
- Un borde es un conjunto de píxeles conectados que se encuentran en la frontera entre dos regiones.

Model of an ideal digital edge



Gray-level profile of a horizontal line through the image

Model of a ramp digital edge



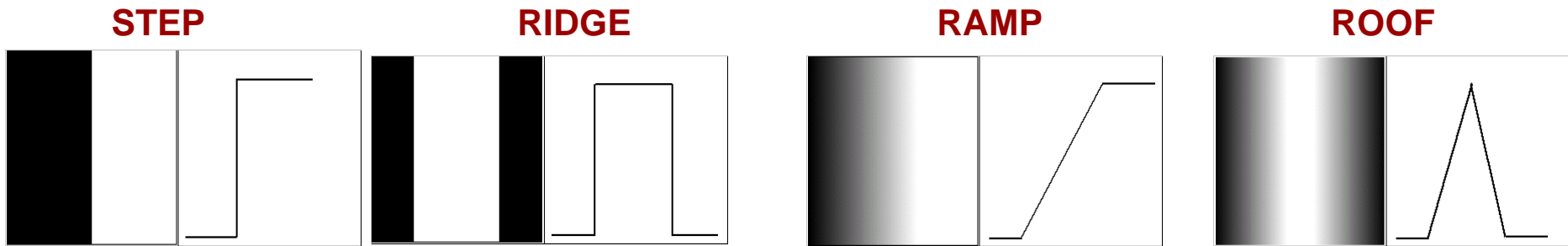
Gray-level profile of a horizontal line through the image



# [ Detección de bordes ]

---

- Tipos de bordes:



# [ Detector de Canny ]

---

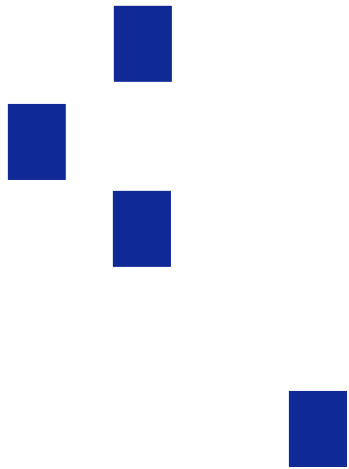
- La idea esencial en la detección de bordes es encontrar puntos en la imagen que tengan localmente máximo gradiente.
- Las imágenes son distorsionadas por el ruido.
- Cualquier proceso de detección de bordes basado en fronteras debe por una parte **suprimir el ruido** y por otra **localizar los bordes** tan preciso como sea posible.
- **El detector de bordes de Canny** tiene en cuenta ciertas condiciones:
  1. **Error.** Se deben detectar todos y sólo los bordes.
  2. **Localización:** La distancia entre el píxel señalado como borde y del borde real debe ser tan pequeña como se pueda
  3. **Respuesta:** No se deben identificar varios píxeles como bordes cuando sólo exista uno.



# [ Detector de Canny ]



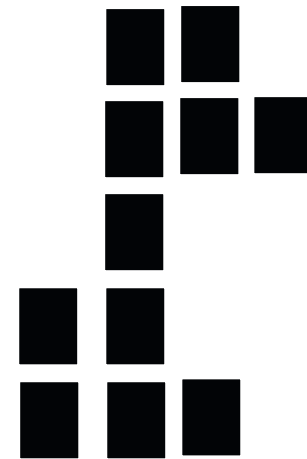
Borde real



Poco robusto frente al ruido



Localización pobre



Demasiadas respuestas

# [ Detector de Canny ]

Ejemplo de aplicación del algoritmo de Canny



# [ Detectores de bordes difusos ]

---

- ¿Qué es un borde? Definición difusa.
- Detectores con conjuntos difusos: Fire, Fedge, Takani-Sugeno Approach, Perception\_Based Approach,...
- *Detectores con conjuntos intervalo-valorados difusos:*
  - *Manejar la amplitud de los intervalos que nos indiquen los saltos de intensidad lo suficientemente grandes entre la intensidad de un pixel y las intensidades de sus píxeles vecinos.*

# [ T-normas y T-conormas ]

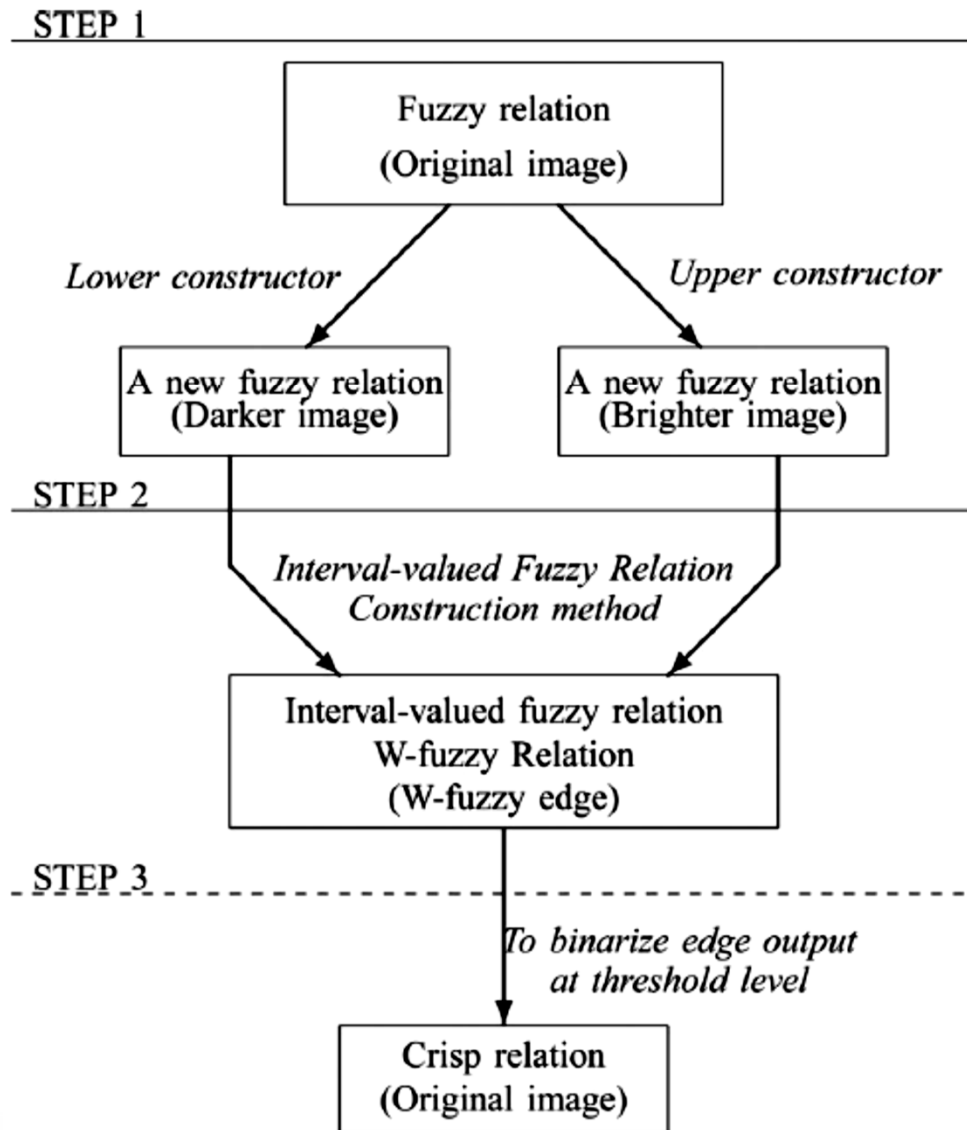
Una función  $T : [0, 1]^2 \rightarrow [0, 1]$  es una t-norma si cumple las siguientes propiedades:

- ❑ Conmutativa,
- ❑ Asociativa,
- ❑ Creciente (en ambos argumentos)
- ❑ Elemento neutro el 1, es decir,  $T(x, 1) = x$  para todo  $x \in [0, 1]$
- Ejemplos:
  - ❑  $T(x, y) = \min(x, y)$ ;  $T(x, y) = x * y$ ;  $T(x, y) = \max(x + y - 1; 0)$ .

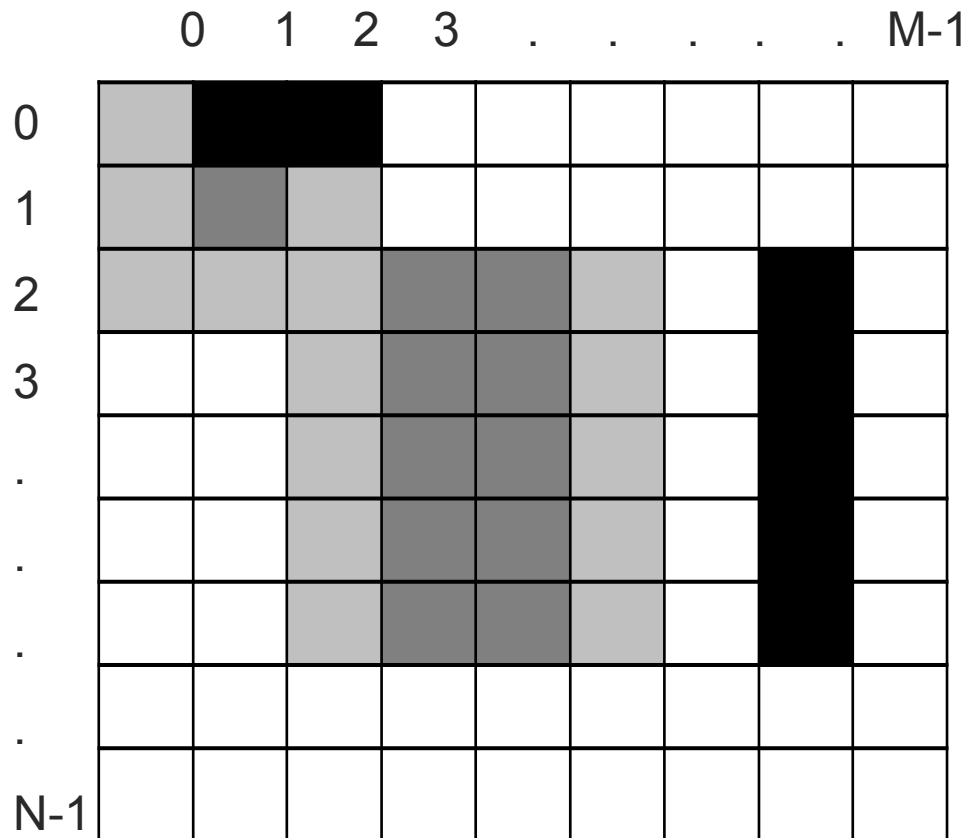
Una función  $S$  es una t-conorma si cumple las mismas propiedades que una t-norma, excepto que el elemento neutro es el 0, es decir,  $S(x, 0) = x$  para todo  $x \in [0, 1]$ .

- Ejemplos:
  - ❑  $S(x, y) = \max(x, y)$ ;  $S(x, y) = x + y - x * y$ ;  $S(x, y) = \min(x + y, 1)$ .

# [ Detección de bordes ]



# [(Step1): Lower y upper constructors ]



Original Image

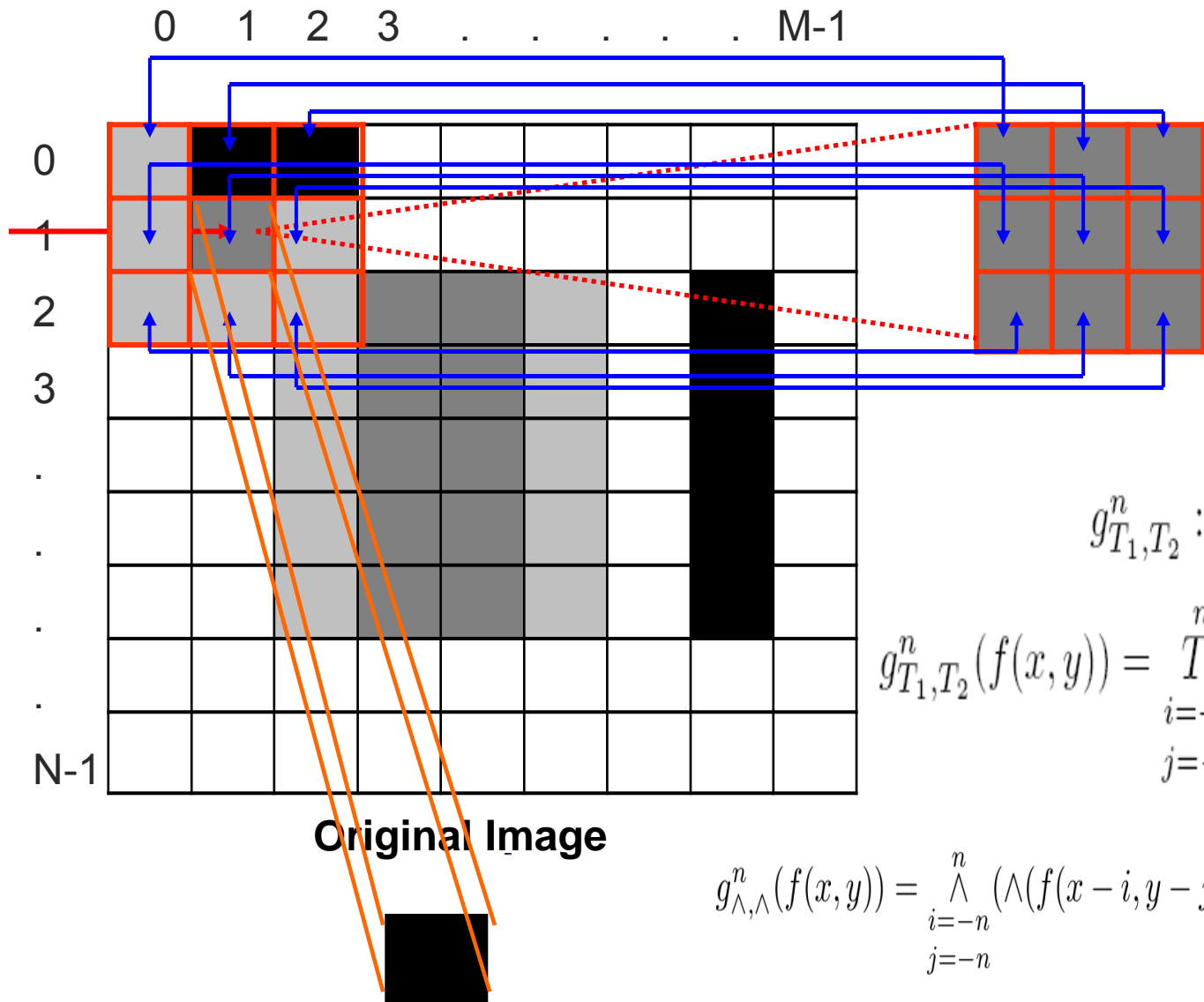
$f$

$$f : X' \times Y' \rightarrow \{0,1,\dots,L-1\}$$

$$f = \begin{pmatrix} f(0,0) & \dots & f(0,M-1) \\ f(1,0) & \dots & f(1,M-1) \\ \dots & \dots & \dots \\ f(N-1,0) & \dots & f(N-1,M-1) \end{pmatrix}$$

$$f \in \mathcal{M}$$

# [ (Step1): Upper constructor: tn-procesamiento ]



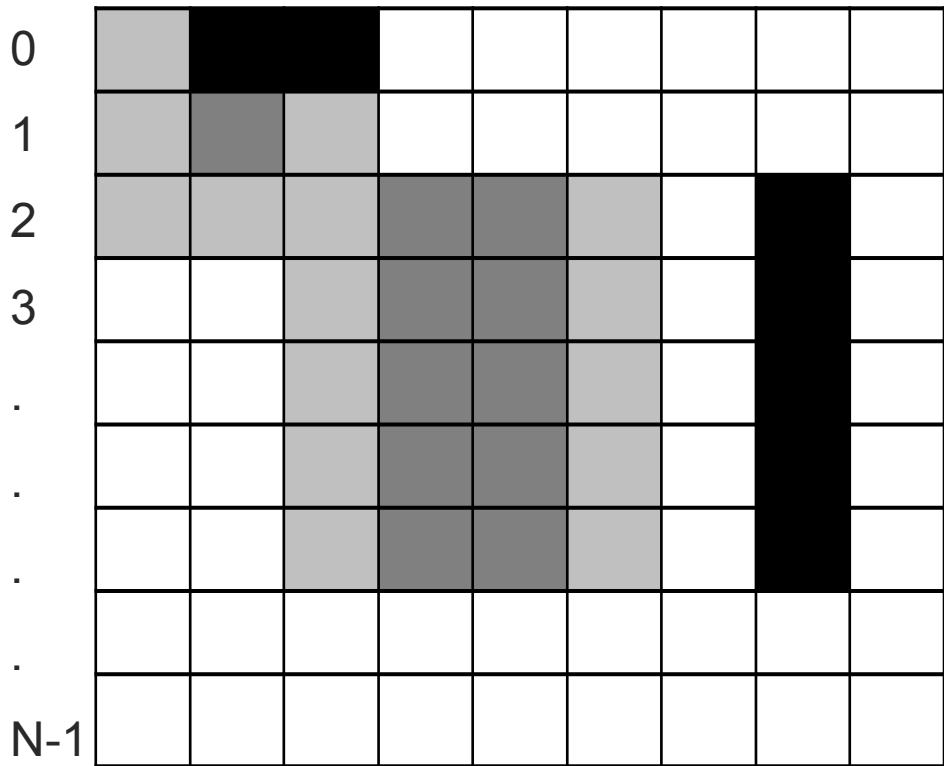
$g_{T_1, T_2}^n : \mathcal{M} \rightarrow \mathcal{M}_n$  given by

$$g_{T_1, T_2}^n(f(x, y)) = \bigwedge_{\substack{i=-n \\ j=-n}}^n (T_2(f(x-i, y-j), f(x, y)))$$

$$g_{\wedge, \wedge}^n(f(x, y)) = \bigwedge_{\substack{i=-n \\ j=-n}}^n (\wedge(f(x-i, y-j), f(x, y))) = \bigwedge_{\substack{i=-n \\ j=-n}}^n (f(x-i, y-j)).$$

**(Step1): Upper constructor:**  
**tn-procesamiento**

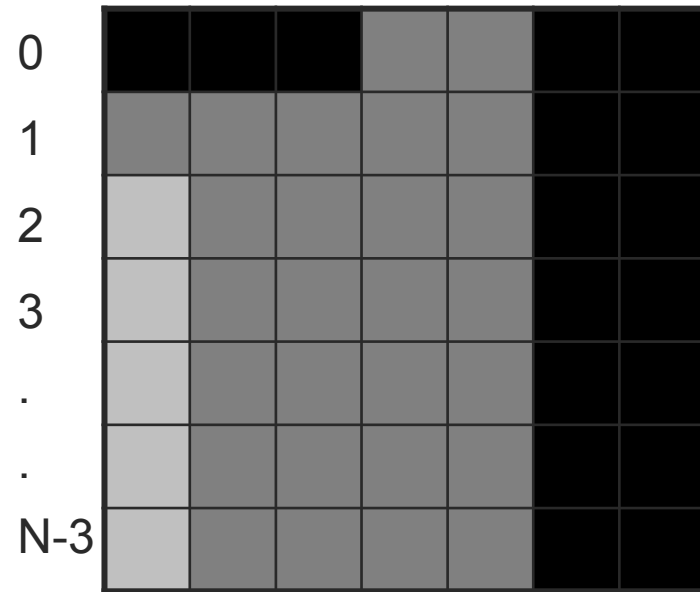
0 1 2 3 . . . M-1



**Original Image**

$f$

0 1 2 3 . . M-3



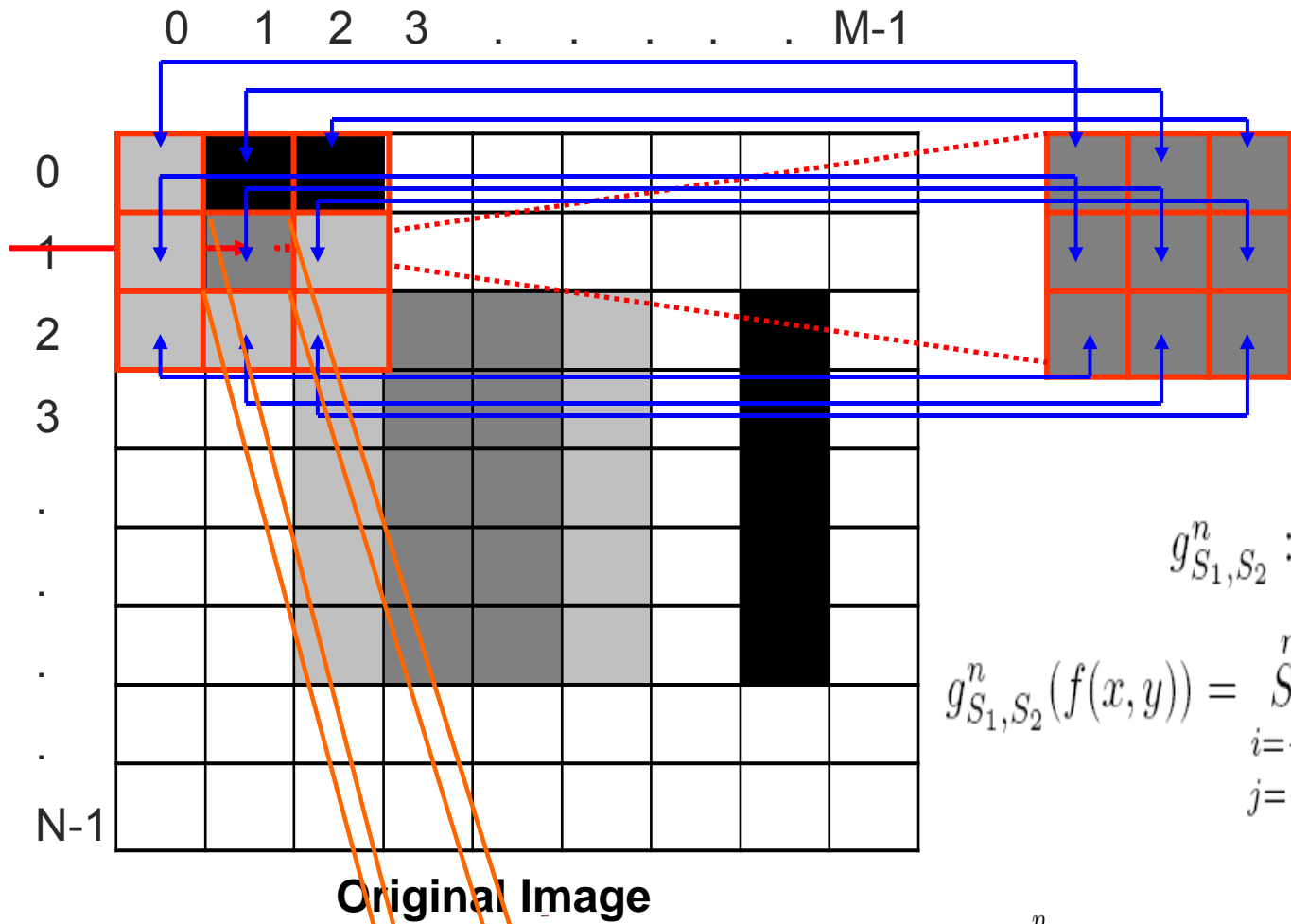
**tn-processing**

$$g_{T_1, T_2}^n(f(x, y))$$

**Darker Image**



# [ (Step1): Upper constructor: sn-procesamiento ]



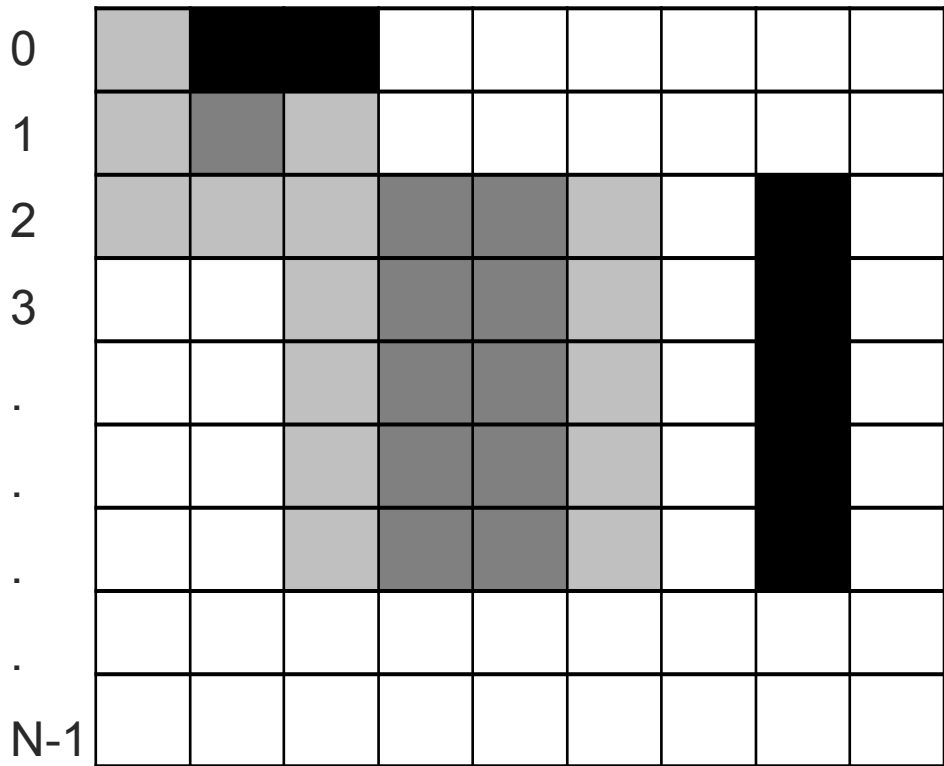
$g_{S_1, S_2}^n : \mathcal{M} \rightarrow \mathcal{M}_n$  given by

$$g_{S_1, S_2}^n(f(x, y)) = \bigvee_{\substack{i=-n \\ j=-n}}^n (S_2(f(x-i, y-j), f(x, y)))$$

$$g_{V, V}^n(f(x, y)) = \bigvee_{\substack{i=-n \\ j=-n}}^n (V(f(x-i, y-j), f(x, y))) = \bigvee_{\substack{i=-n \\ j=-n}}^n (f(x-i, y-j)).$$

**(Step1): Upper constructor:  
sn-procesamiento**

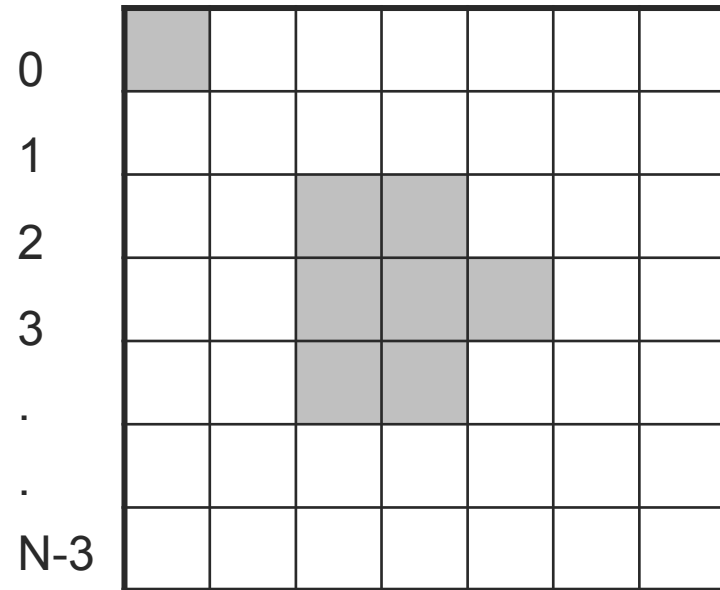
0 1 2 3 . . . M-1



**Original Image**

*f*

0 1 2 3 . . M-3

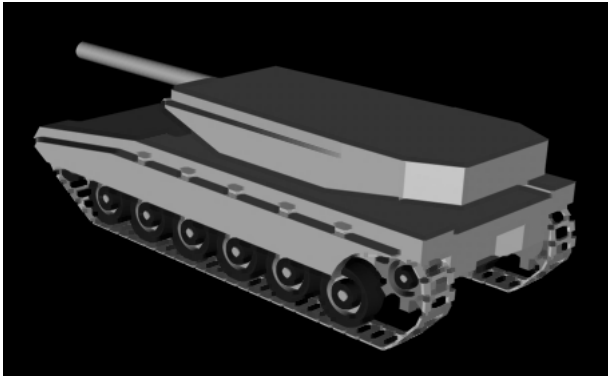


**sn-processing**

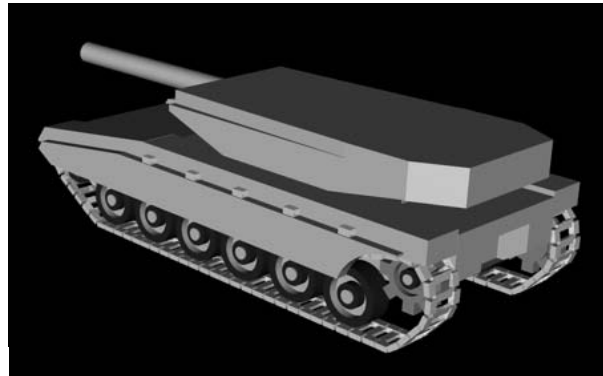
$g_{S_1, S_2}^n(f(x, y))$   
**Brighter Image**

# [Ejemplo (Step1): Lower y upper constructors]

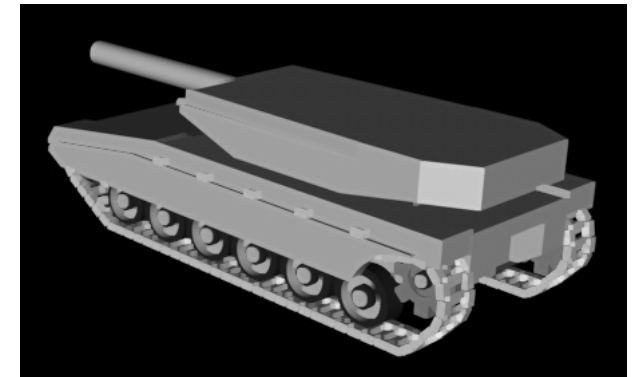
$$T_1 = T_2 = \wedge$$



Original Image

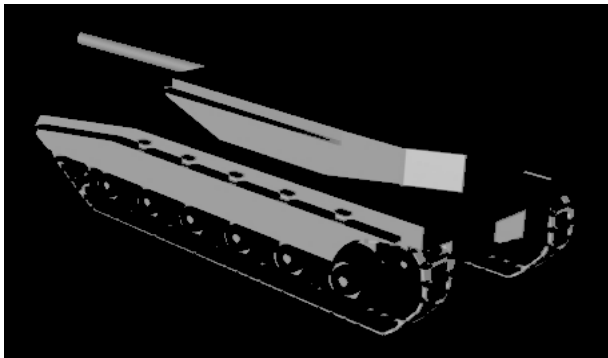


$$S_1 = S_2 = \vee$$



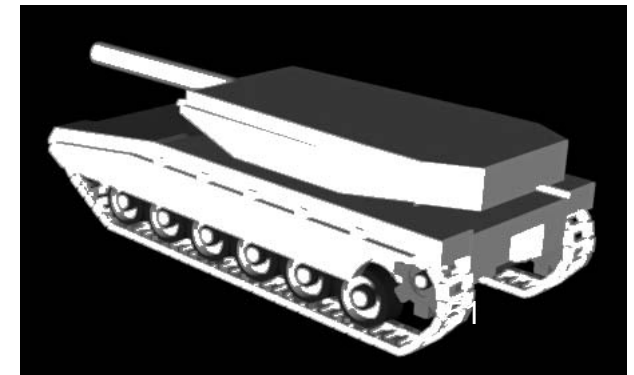
$$T_1 = \wedge$$

$$T_2(x, y) = \begin{cases} \wedge(x, y) & \text{if } x + y > 1 \\ 0 & \text{otherwise} \end{cases}$$



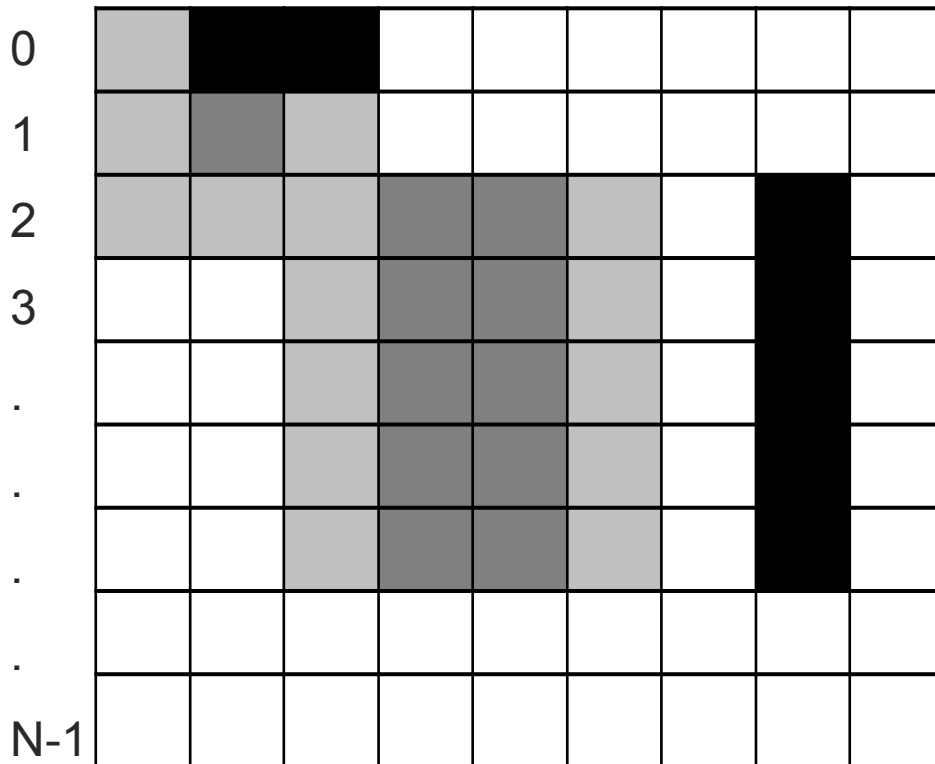
$$S_1 = \vee$$

$$S_2(x, y) = \begin{cases} \vee(x, y) & \text{if } x + y < 1 \\ 1 & \text{otherwise} \end{cases}$$



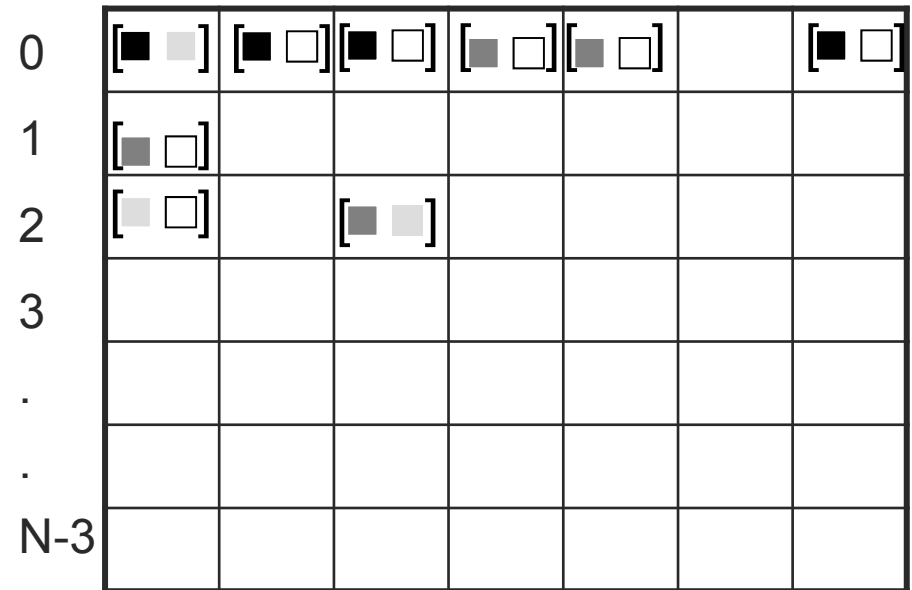
# [ (Step2): Interval-valued fuzzy image ]

0 1 2 3 . . . M-1



Original Image

0 1 2 3 . . . M-3



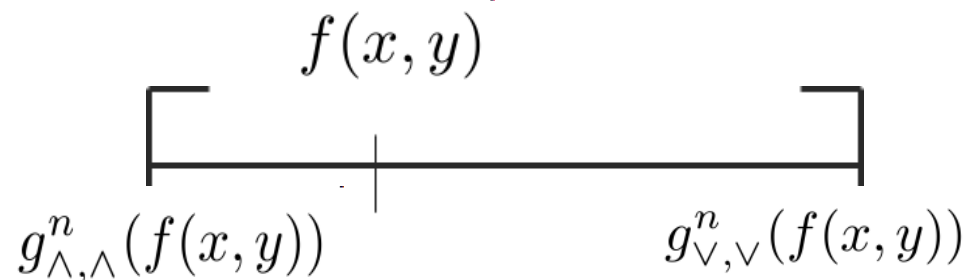
Interval-valued Fuzzy Image

# [tn-procesamiento y sn-procesamiento]

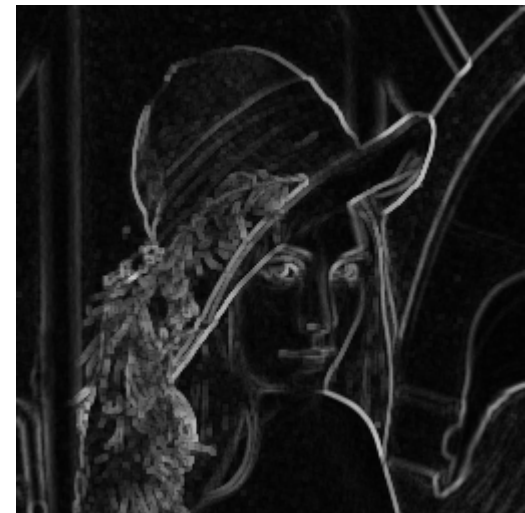
$$F^n = \{((x, y), M_{F^n}(x, y) = [g_{T_1, T_2}^n(f(x, y)), g_{S_1, S_2}^n(f(x, y))] \in L([0, 1])) | x \in X, y \in Y\} \in IVFs(X \times Y)$$

$$\mathbf{f} = \{((x, y), M_{\mathbf{f}}(x, y) = [f(x, y), f(x, y)] \in L([0, 1])) | x \in X, y \in Y\} \in IVFs(X \times Y)$$

$$W(F^n) = \{((x, y), \mu_{W(F^n)}(x, y) = g_{S_1, S_2}^n(f(x, y)) - g_{T_1, T_2}^n(f(x, y))) | (x, y) \in X \times Y\} \in FSs(X \times Y)$$



# [ tn-procesamiento y sn-procesamiento ]



# [tn-procesamiento y sn-procesamiento]

*IF en la ventana considerada todos los píxeles tienen la misma intensidad*

*THEN*

*la amplitud debe ser cero (el pixel no pertenece al borde)*

$W(F^n(x, y)) = 0$  if and only if  $T_1 = T_2 = \wedge$  and  $S_1 = S_2 = \vee$ .

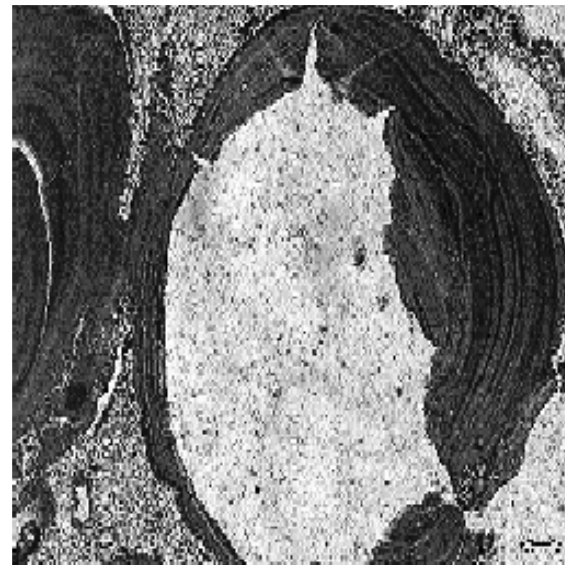
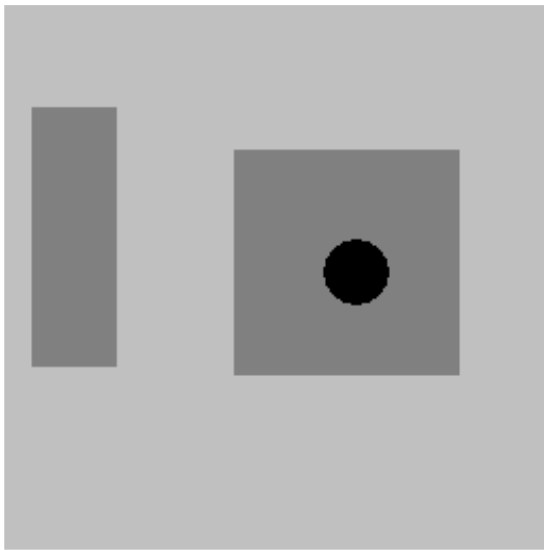
# [ Bordes de imágenes en escala de grises ]

---

**IF** *un pixel pertenece al borde*

**THEN**

*su correspondiente amplitud debería ser  
suficientemente grande.*



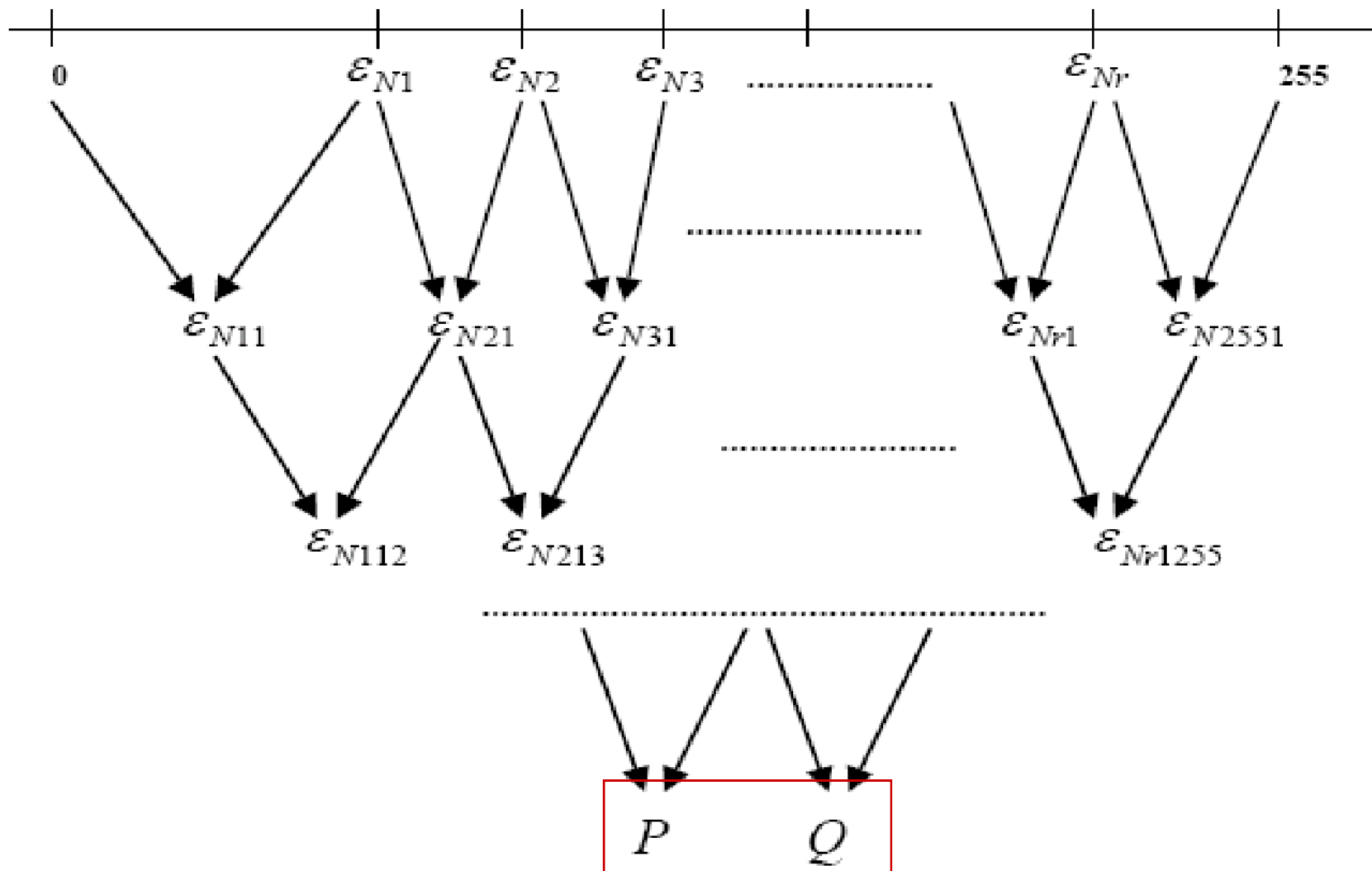


# Entropía intervalar

$$\mathcal{E}_{\mathcal{N}}(F^n) = \frac{\sum_{\substack{n \leq x \leq N-(n+1) \\ n \leq y \leq M-(n+1)}} g_{S_1, S_2}^n(f(x, y)) - g_{T_1, T_2}^n(f(x, y))}{(N - (n + 1)) \times (M - (n + 1))}$$

Entropía intervalar  Proporción de píxeles borde

# Entropía intervalar



# [ Bordes de imágenes en escala de grises ]

- Construcción de tres subimágenes

IVn imagen  
 $F^n$

$$\left\{ \begin{array}{l} F_p^n = \{((x, y), M_{F_p^n}(x, y)) | (x, y) \in X \times Y\}, \text{ where} \\ M_{F_p^n}(x, y) = \begin{cases} M_{F^n}(x, y) & \text{if } 0 < W(F^n(x, y)) < p \\ 0 & \text{otherwise} \end{cases} \\ F_{p,q}^n = \{((x, y), M_{F_{p,q}^n}(x, y)) | (x, y) \in X \times Y\}, \text{ where} \\ M_{F_{p,q}^n}(x, y) = \begin{cases} M_{F^n}(x, y) & \text{if } p \leq W(F^n(x, y)) < q \\ 0 & \text{otherwise} \end{cases} \\ F_q^n = \{((x, y), M_{F_q^n}(x, y)) | (x, y) \in X \times Y\}, \text{ where} \\ M_{F_q^n}(x, y) = \begin{cases} M_{F^n}(x, y) & \text{if } q \leq W(F^n(x, y)) \\ 0 & \text{otherwise} \end{cases} \end{array} \right.$$

# Bordes de imágenes en escala de grises

Original image 654x388: D.Z.



F1



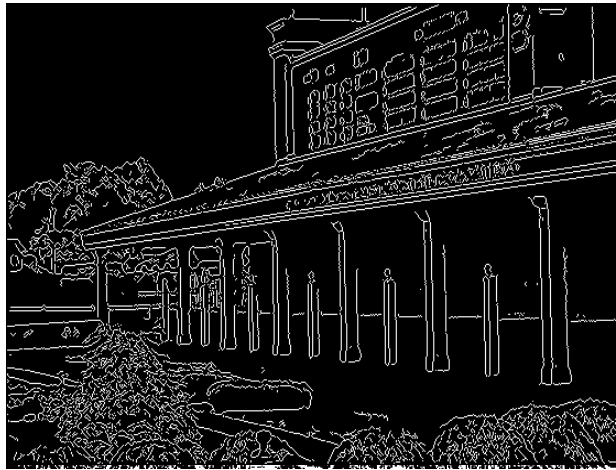
F2



F3



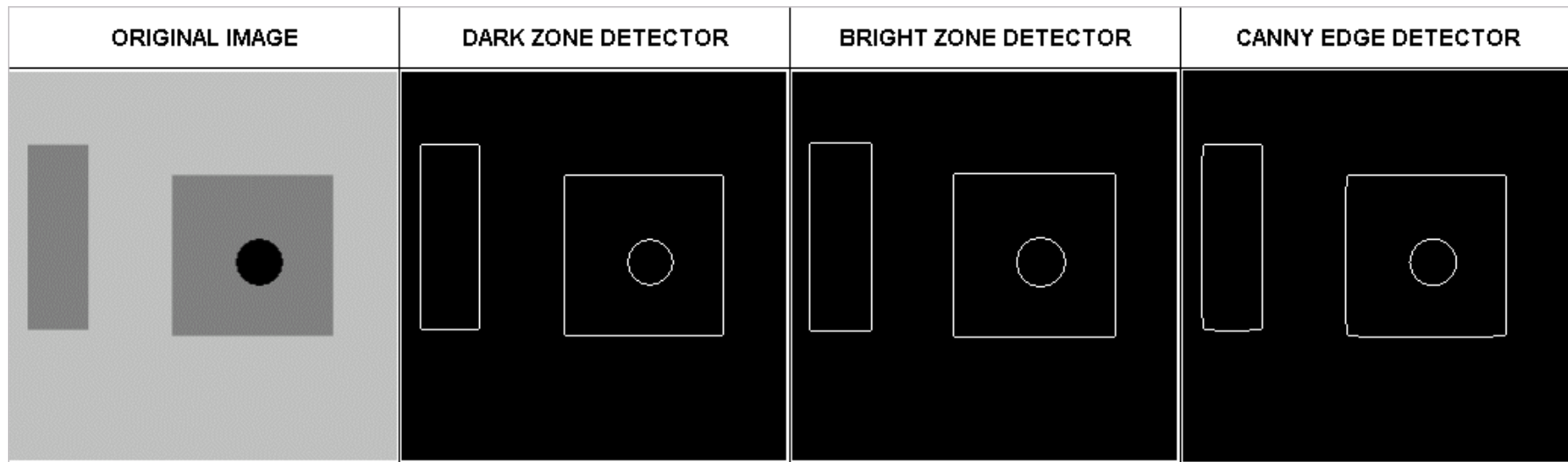
Canny



THIN & CLEAN



# [ Bordes de imágenes en escala de grises ]

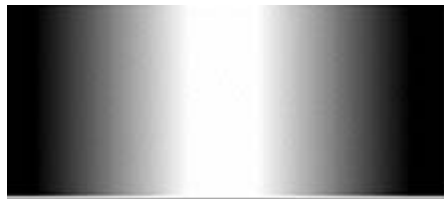


*H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Interval-valued fuzzy sets constructed from matrices: Application to edge detection, Fuzzy Sets and systems, 160, 1819–1840, 2009*

*E. Barrenechea, H. Bustince, B. De Baets, C. Lopez-Molina, Construction of Interval-Valued Fuzzy Relations With Application to the Generation of Fuzzy Edge Images, IEEE Transactions On Fuzzy Systems, 19, 819-830, 2011.*

# [ Bordes de imágenes en escala de grises ]

Original Image



Our approach



Canny



*H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Interval-valued fuzzy sets constructed from matrices: Application to edge detection, Fuzzy Sets and systems, 160, 1819–1840, 2009*

*E. Barrenechea, H. Bustince, B. De Baets, C. Lopez-Molina, Construction of Interval-Valued Fuzzy Relations With Application to the Generation of Fuzzy Edge Images, IEEE Transactions On Fuzzy Systems, 19, 819-830, 2011.*

# Bordes de imágenes en escala de grises

Original Image



Our approach



Canny



# [ Bordes de imágenes en escala de grises ]

---

- Practicar con las imágenes y el código

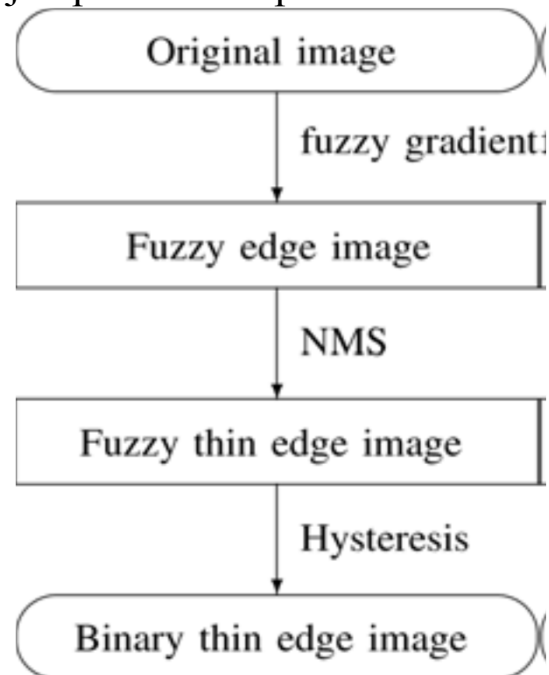
`borzonclara(imagen,tamañoventana)`

`borzonclara('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo_Images\EdgeDetection\IVFS\circuit.bmp',3)`



# Bordes de imágenes en escala de grises

- Otros métodos fuzzy que normalmente se utilizan los siguientes pasos (Bezdek et al):
  1. **Acondicionar**: Suavizado de la imagen original con un filtro Gaussiano.
  2. **Extracción** de características: obtener el gradiente.
  3. **Adelgazamiento** de la imagen de características utilizando el non-maximum suppression.
  4. **Binarizar** la imagen aplicando hysteresis. Seleccionar los umbrales de este proceso por medio de diferentes técnicas (por ejemplo la no-supervisada de Medina-Carnicer et al.)



*J. Bezdek, R. Chandrasekhar, Y. Attikouzel, A geometric approach to edge detection, IEEE Transactions on Fuzzy Systems 6 (1), 52–75, 1998.*

*R. Medina-Carnicer, F. Madrid-Cuevas, A. Carmona-Poyato, R. Muñoz Salinas, On candidates selection for hysteresis thresholds in edge detection, Pattern Recognition 42 (7), 1284–1296, 2009*

# Bordes de imágenes en escala de grises

- **Fuzzy Morphology**

*Definition 5:* The fuzzy dilation  $D_C(A, B)$  and the fuzzy erosion  $E_I(A, B)$  of  $A$  by  $B$  are the gray-level images defined by

$$D_C(A, B)(y) = \sup_{x \in d_A \cap T_y(d_B)} C(B(x - y), A(x))$$
$$E_I(A, B)(y) = \inf_{x \in d_A \cap T_y(d_B)} I(B(x - y), A(x))$$

La diferencia entre la dilatación y la erosión fuzzy en niveles de grises,  $D_C(A, B) \setminus E_I(A, B)$ , se conoce con el operador gradient fuzzy morfológico y puede usarse en detección de bordes.

Sustituir el Sup y el Inf por t-conormas y t-normas

# [ Bordes de imágenes en escala de grises ]

- **Fuzzy Morphology**

FuzzyMorph (imagen,C,Imp,TS)

- imagen: original image;
- C: conyuntor: 'T\_LK' para Dilatation
- Imp: implicación 'I\_KD' para Erosion
- TS:
  - 'B': min y max,
  - 'SS': Schweizer-Sklar

FuzzyMorph('D:\Fuzzy\CursoVeranoAEPIA2016\Codigo\_Images\EdgeDetection\ImagenesEdges\fo  
tografo.bmp', 'T\_nM', 'I\_KD', 'B');