



Sistemas de Visión Artificial

Tema 3. Técnicas de Segmentación

*Autores: Sira Palazuelos, Luis M. Bergasa, Manuel Mazo,
M. Ángel García, Marisol Escudero, J. Manuel Miguel
Departamento de Electrónica. Universidad de Alcalá.*

Índice

1. **Introducción**
2. **Segmentación basada en el histograma**
 - Análisis Global frente a Análisis Local
 - Umbralización
 - Métodos de clustering (K-means)
 - Métodos estocásticos
3. **Detección de discontinuidades**
 - Detección de bordes
 - Detección de esquinas
4. **Segmentación basada en bordes**
 - Análisis local
 - Análisis global
5. **Segmentación basada en regiones**
 - Growing regions
 - Splitting & Merging regions



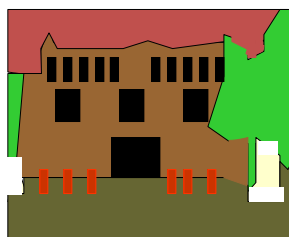
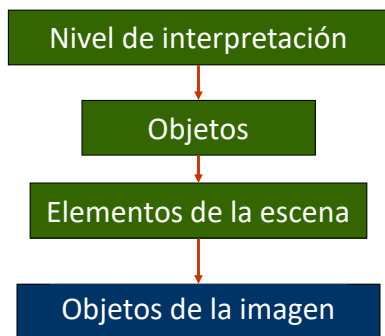
1. Introducción

Introducción

- ☐ **Segmentar** una imagen es **dividirla** en las partes u objetos de interés que la componen.
- ☐ Los **métodos de segmentación** buscan zonas que tengan características **homogéneas** dentro de ellas y que **contrasten** con las de alrededor.
- ☐ Las medidas de homogeneidad y contraste incluyen **características** como: **nivel de gris, color, textura, bordes, movimiento**, etc.
- ☐ Estas zonas con valor del atributo uniforme se considerarán **objetos** de la imagen.



- Correspondencia entre una escena real y su imagen asociada



- ¿Qué objetos de una imagen se corresponden con elementos con significado de una escena?
- La respuesta depende del **tipo y complejidad** de la imagen.
- El concepto de segmentación **no está bien definido**.

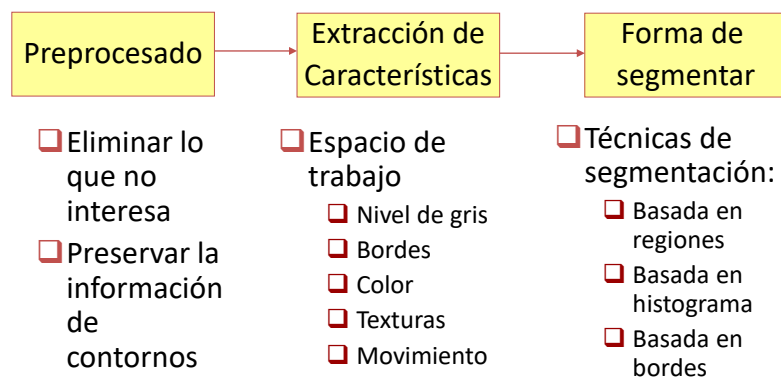


- ❑ Evaluación de la **calidad** de la segmentación



- ❑ A veces es una tarea difícil

➤ Esquema general de funcionamiento

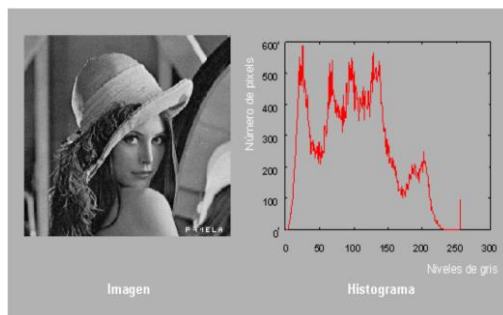
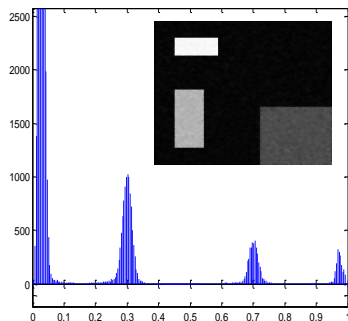




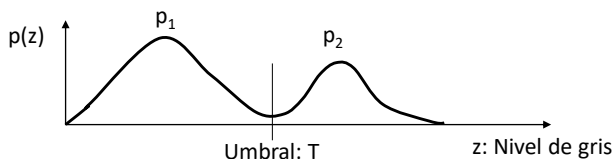
4. Segmentación basada en el histograma

Segmentación basada en histograma

- En ocasiones, los distintos elementos de interés de la imagen son identificables en el **histograma** → podemos utilizar esta característica en la **segmentación**.
- La complejidad del algoritmo necesario para dicha segmentación dependerá del grado de fusión de los histogramas de los distintos objetos.



- La segmentación basada en el **histograma** considera que cada **lóbulo** del histograma corresponde a la distribución en probabilidad de cada clase **objeto** en la imagen. Un ejemplo con 2 objetos podría ser:



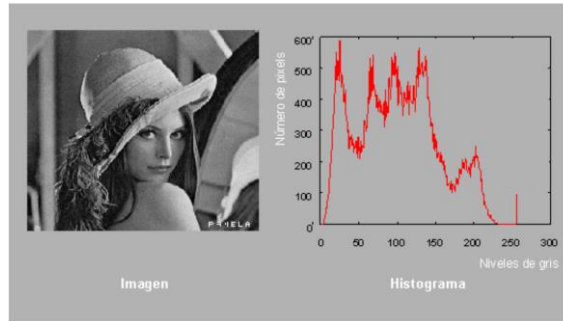
- La segmentación se basa en un proceso de umbralización:

$$g(u,v) = \begin{cases} 255 & \text{si } f(u,v) > T \\ 0 & \text{si } f(u,v) \leq T \end{cases} \quad f(u,v) \longrightarrow \text{Umbralización} \longrightarrow g(u,v)$$

- Problema: Cálculo del umbral** $T = T[u, v, s(u,v), f(u,v)]$
 - $f(u,v)$ es la intensidad del píxel (u,v) y $s(u,v)$ es alguna propiedad local del píxel, por ejemplo, la intensidad media de un entorno de vecindad centrado en (u,v) .

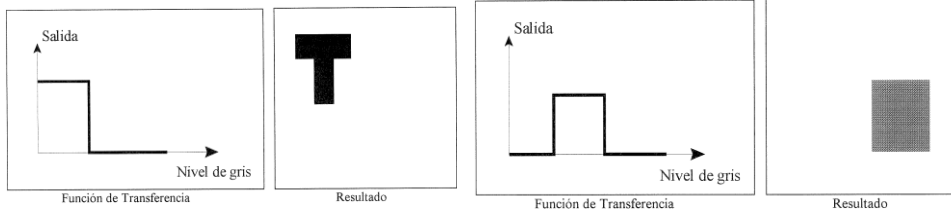
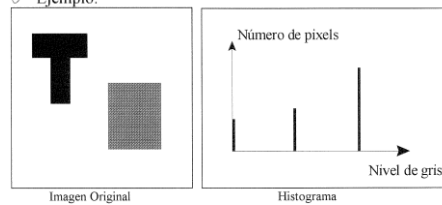
- Tipos de umbralización:
 - Según el **número de umbrales** a aplicar en la imagen:
 - Global**: mismos umbrales en toda la imagen $\rightarrow T$ depende sólo de $f(u,v)$.
 - Local**: múltiples umbrales $\rightarrow T$ depende de $f(u,v)$ y $s(u,v)$.
 - Multiespectral**: imágenes RGB \rightarrow un umbral T para cada componente de color.
 - Según la forma de cálculo:
 - Supervisados** (obtenidos manualmente):
 - Método lineal: Segmentación mediante distancia euclídea.
 - Métodos no lineales: Segmentación mediante el teorema de Bayes.
 - No supervisados** (obtenidos de forma automática)
 - Búsqueda de máximos y mínimos sobre el histograma: Poco efectivo en imágenes reales (muchos máximos y mínimos locales).
 - Métodos de *clustering* (lineal): Algoritmo K-means.
 - Métodos estocásticos (no lineal): Modelos de mezcla Gaussianos (GMM).

- A partir del histograma **global** de la imagen se obtienen los umbrales a aplicar a **todos** los píxeles de la imagen:



$$g(u,v) = \begin{cases} 1 & \text{si } T_1 > f(u,v) > T_2 \\ 0 & \text{si resto} \end{cases}$$

✓ Ejemplo:



- En ocasiones, los valores de intensidad de los píxeles pueden **variar** en función de la iluminación de la escena: Es muy difícil/imposible segmentar adecuadamente **todas** las zonas de la imagen utilizando un umbral global.
- Los resultados mejoran calculando un **umbral local, adaptativo**, diferente para cada entorno, definido para una vecindad del píxel a umbralizar, en función de algún estadístico del entorno como: media, mediana, etc.

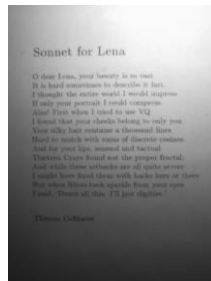


Imagen original



Imagen umbralizada
con umbral global

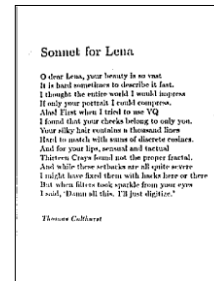
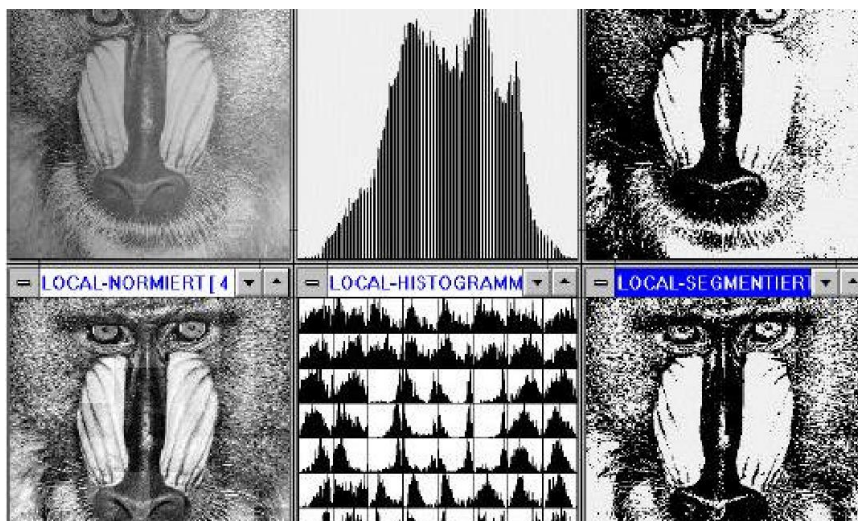
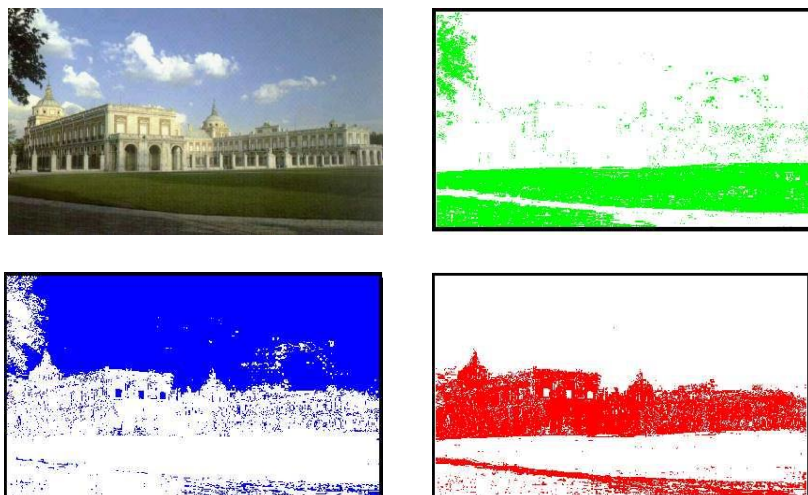
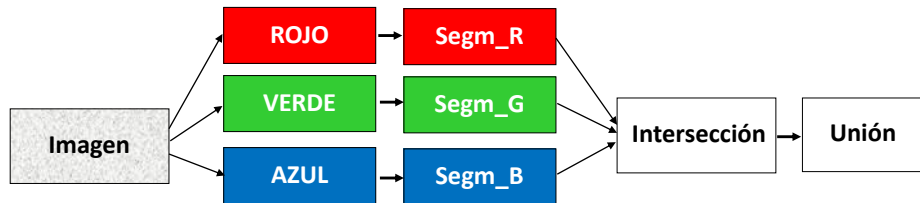


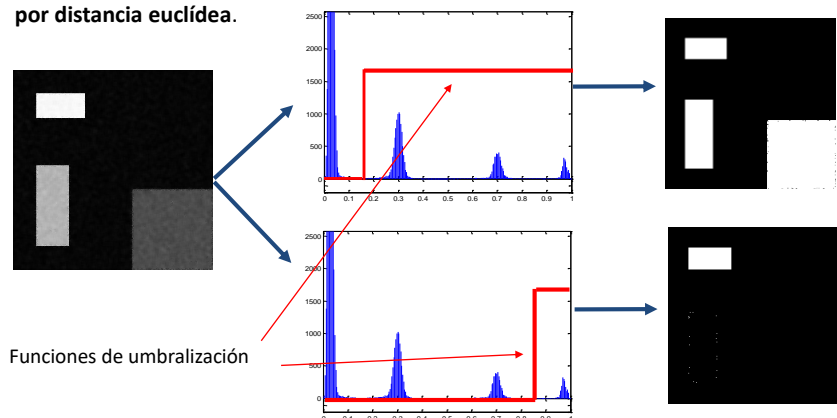
Imagen umbralizada
con umbral local



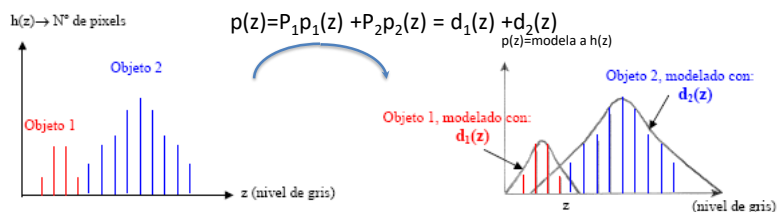
- Mediante la union (OR) o intersección (AND) de las imágenes umbralizadas para cada componente R, G, B se puede obtener una segmentación en color.



- **Umbralización supervisada:** hay un **entrenamiento** previo en que se proporciona al sistema información sobre las clases y sus características.
- **Umbralización supervisada lineal:** Se obtienen los umbrales a partir de los centroides de las clases buscando (manualmente) los puntos **medios** entre ellos: **segmentación por distancia euclídea**.



- **Umbralización supervisada no lineal:** hay veces que el umbral óptimo **no** es el punto medio.
- Se aproxima el histograma mediante una **suma ponderada de funciones de densidad de probabilidad** (pdf) correspondientes a los niveles de gris de los objetos presentes en la imagen.
- Para el caso de un histograma bimodal (dos objetos):



donde:

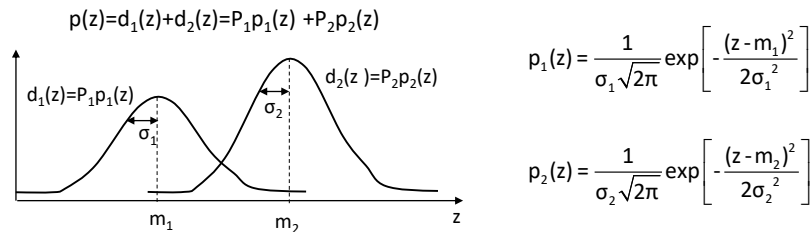
z = variable aleatoria que representa intensidad luminosa (nivel de gris),

$p_1(z)$ y $p_2(z)$ = pdf asociadas a los dos objetos del histograma,

P_1 y P_2 = probabilidades a priori de que un píxel pertenezca al objeto 1 y al 2 ($P_1 + P_2 = 1$).

$d_1(z) = P_1 p_1(z)$ y $d_2(z) = P_2 p_2(z)$: modelos del histograma referentes al objeto 1 y 2

- Es frecuente hacer la **aproximación por funciones Gaussianas**:



donde:

m_1, σ_1 y m_2, σ_2 = media y desviación típica de las fdp's asociadas a los objetos 1 y 2.

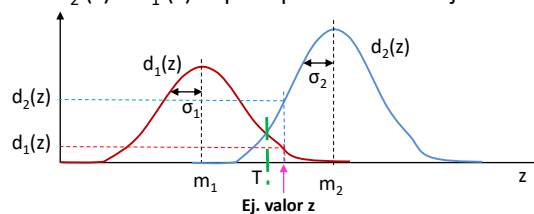
- Los **pesos**, P_1 y P_2 , que ponderan las fdp serán proporcionales al área que ocupa la parte del histograma referente al objeto 1 y 2:

$$\left. \begin{array}{l} \text{- más probabilidad de objeto 2 que de objeto 1: } P_2 > P_1 \\ \text{- clases equiprobables: } P_1 = P_2 \dots \end{array} \right\} \boxed{P_1 + P_2 = 1}$$

- Cada píxel se clasifica en la clase en la que la **pertenencia** al modelo de su nivel de gris es **máxima**: se calcula $d_i(z)$ para cada fdp y se asigna el píxel a la clase que proporcione un valor mayor:

Si $d_1(z) > d_2(z)$ el píxel pertenece al objeto 1

Si $d_2(z) > d_1(z)$ el píxel pertenece al objeto 2



- El **umbral óptimo** se selecciona para aquel nivel $z = T$ que cumple:

$$d_1(z) = d_2(z) \rightarrow P_1 * p_1(z) = P_2 * p_2(z)$$

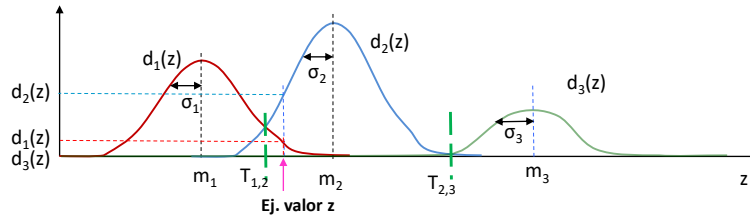
- Si $\sigma_1 = \sigma_2 = \sigma$ (histograma bimodal modelado con funciones gaussianas):

$$T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \frac{P_2}{P_1}$$

- Para **histogramas multimodales** (con m objetos en la imagen):

$$p(z) = d_1(z) + d_2(z) + \dots d_m(z) = P_1 * p_1(z) + P_2 * p_2(z) + \dots P_m * p_m(z)$$

$$\text{donde } P_1 + P_2 + \dots P_m = 1$$



- Cada píxel se clasifica como perteneciente a la clase para la cual la pertenencia al modelo de su nivel de gris (z) sea **máxima**.

El píxel de nivel de gris z pertenece al objeto o clase k -ésima si:

$$d_k(z) > d_j(z) \text{ siendo } j=1,2,\dots,m \text{ y distinto de } k$$

- Los **umbrales** entre clase y clase se seleccionan de forma que:

$$P_k * p_k(T_{kj}) = P_j * p_j(T_{kj})$$

siendo T_{kj} el umbral de separación entre la clase k y j

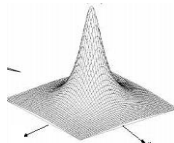
- ❑ Este procedimiento se puede generalizar para un **histograma n-dimensional**, (que representa el nº de puntos de la imagen con un vector n-dimensional de características de valor igual a \bar{z}), **de m objetos**, que se modelaría mediante:

$$p(\bar{z}) = d_1(\bar{z}) + d_2(\bar{z}) + \dots d_m(\bar{z}) = P_1 * p_1(\bar{z}) + P_2 * p_2(\bar{z}) + \dots P_m * p_m(\bar{z})$$

siendo: $P_1 + P_2 + \dots P_m = 1$

$p_1(\bar{z}), p_2(\bar{z}) \dots p_m(\bar{z})$ fdp n-dimensionales asociadas con los objetos 1, 2, ..., m :

Ejemplo: Gaussiana bidimensional



$$p_1(\bar{z}) = \frac{1}{(2\pi)^{\frac{n}{2}} |C_1|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\bar{z} - \bar{\mu}_1)^T C_1^{-1} (\bar{z} - \bar{\mu}_1) \right]$$

donde $\bar{\mu}_1, C_1$ representan los vectores medias y matrices de covarianza de la fdp asociada con la parte del histograma n-dimensional del objeto 1.

- ❑ El píxel de vector de características \bar{z} pertenecería a la clase k -ésima si:

$$d_k(\bar{z}) > d_j(\bar{z}) \text{ siendo } j = 1, 2, \dots m \text{ y distinto de } k.$$

- Si se desea **segmentar un único objeto** con un color uniforme en la imagen se puede modelar la pdf de este color mediante una función Gaussiana definida en el espacio de color (espacio RG):
 - Cálculo de los parámetros del modelo a priori
 - Segmentación mediante umbral



Imagen original



Objeto a segmentar

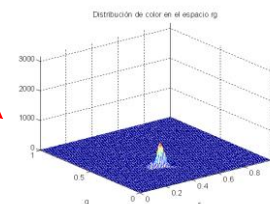


Imagen segmentada

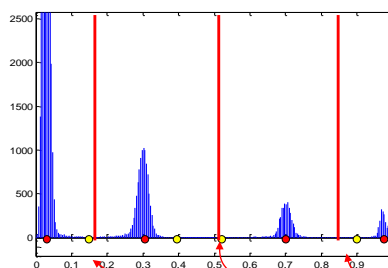
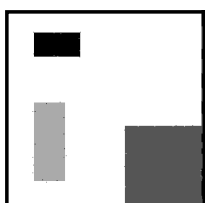
$$f(x_j / \text{skin}) = \frac{1}{2\pi |C_s|^{0.5}} e^{-0.5(x_j - m_s)^T C_s^{-1} (x_j - m_s)}$$

$$x_j \in \text{skin} \text{ si } f(x / \text{skin}) > Th$$

Segmentación

Parámetros del modelo

- Algoritmos de clustering:** clasificación automática, no supervisada, de los píxeles: el sistema aprende automáticamente, a partir de un conjunto de entrenamiento en que no le damos previamente clasificados los niveles.
- Umbralización no supervisada lineal:**
 - Número de clases conocido a priori,
 - Entrenamiento no supervisado de centroides,
 - Clasificación mediante distancia euclídea (lineal).



- Posición inicial de los centroides (antes del entrenamiento, aleatoria)
- Posición final de los centroides, después del entrenamiento
- Umbral: punto medio entre centroides consecutivos

Umbral obtenidos

➤ **Algoritmo k-means:**

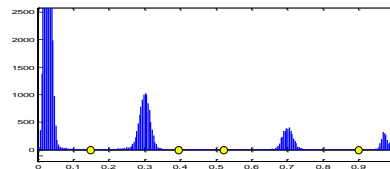
- Su objetivo es clasificar los píxeles $\{X\}$ de la imagen en m clases $\{\alpha_1, \alpha_2, \alpha_m\}$
- Es un algoritmo de clasificación **no** supervisada lineal, que asume que **se conoce el número de clases** a clasificar (m), pero nada más: en la etapa de entrenamiento no se conoce a qué clase pertenece cada nivel de gris.

➤ **Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (I):**

- **Paso 1:** Se toman al azar entre los elementos a agrupar m vectores, centroides de las m clases.

$$\alpha_1: z_1(0); \alpha_2: z_2(0); \dots; \alpha_m: z_m(0)$$

Las componentes de los vectores son características de los píxeles: intensidad, color, etc.



● Posición inicial de los centroides (antes del entrenamiento, aleatoria)

➤ **Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (II):**

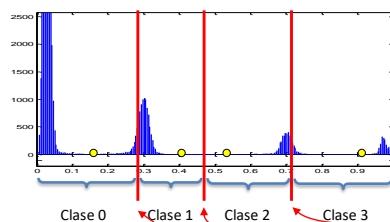
- **Paso 2:** En la iteración t se distribuyen todas las muestras entre las m clases

For $i = 1$ to N

$$x_i \in \alpha_j(t) \quad \text{si} \quad \|x_i - z_j(t)\| < \|x_i - z_k(t)\|$$

$$\forall k = 1, 2, \dots, m / k \neq j$$

End{For}



● Posición inicial de los centroides (antes del entrenamiento, aleatoria)

▮ Umbral: punto medio entre centroides consecutivos

Umbral correspondientes con esos centroides

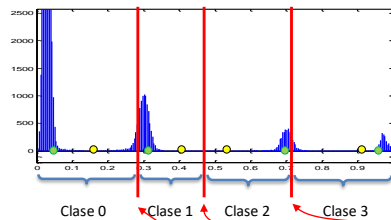
➤ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (III):

- **Paso 3:** Se actualizan los centroides de las clases con objeto de minimizar la siguiente función de coste:

$$J_k = \sum_{x \in \alpha_k(t)} \|x - z_k(t)\|^2 \quad k = 1, 2, \dots, m \quad (m \text{ es el n}^\circ \text{ de clases})$$

Con esto conseguimos maximizar la variación intercluster y minimizar la variación intracluster. Esta minimización se logra utilizando la media aritmética de $\alpha_k(t)$.

$$z_k(t+1) = \frac{1}{N_k(t)} \sum_{i=1}^{N_k(t)} x_i \quad ; k = 1, 2, \dots, m \quad ; N_k(t) : \text{elementos de } \alpha_k(t)$$



- Posición inicial de los centroides (antes del entrenamiento, aleatoria)
- Nuevos centroides calculados como media de los clusters generados

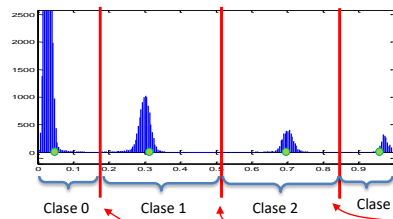
Umbral calculado con los centroides anteriores

➤ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (IV):

- **Paso 4:** Se vuelve al paso 2 hasta que los vectores prototipos mantengan su posición:

$$\sum_{j=1}^m \|z_j(t+1) - z_j(t)\| < \epsilon$$

También podemos decir que repetimos el proceso hasta que ya no haya más reasignaciones de centroides en el paso 2.

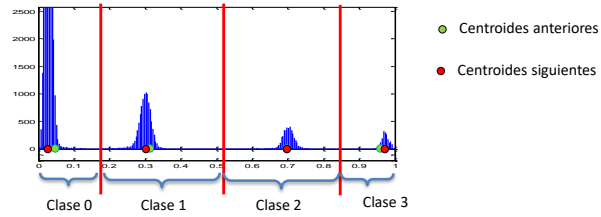


- Nuevos centroides calculados como media de los clusters generados

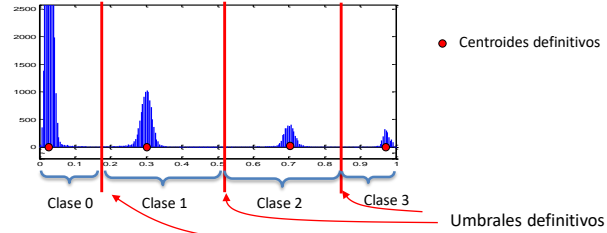
Nuevos umbrales

➤ Algoritmo de entrenamiento recursivo para encontrar el centroide de cada clase (V):

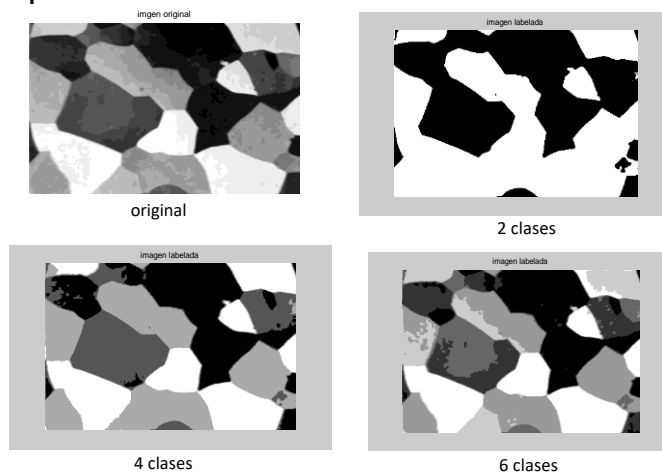
➤ Siguiete iteración:



➤ Valores definitivos: en la siguiente iteración no se modifica la posición de los centroides ni de los umbrales.



➤ Ejemplos:



➤ **Ventajas:**

- Algoritmo sencillo y muy eficiente.

➤ **Inconvenientes:**

- Se necesita saber el **número de clases** a priori.
- El método de cálculo de clases no es óptimo ya que considera que todas tienen la misma dispersión estadística para cada dimensión (d) de X, y esto no es cierto en todos los casos.
- Aprendizaje local y dependiente de la inicialización, en cada iteración únicamente se mueve el centroide de la clase ganadora.

➤ **Segmentación basada en histograma**➤ **Ventajas:**

- Aplicación **simple y rápida**
- Puede obtenerse de forma **no** supervisada
- Buenos resultados con imágenes **simples** e iluminación controlada
- Aplicable para aplicaciones de **seguimiento** de objetos en secuencias

➤ **Inconvenientes:**

- Usa un umbral **único** para toda la imagen: malos resultados con iluminación no uniforme
- **No controla la conectividad espacial**
 - **Agujeros** en las clases
 - Píxeles **aislados** de una clase dentro de otras

- [Bergasa et al.,00] L.M. Bergasa, M. Mazo, A. Gardel, M. A. Sotelo, L. Boquete, "Unsupervised and adaptive Gaussian skin-color model", Image and Vision Computing (Editorial Elsevier), Number 18, pages 987 to 1003, September of 2000.
- [Campbell et al. 96] N.W. Campbell, B.T. Thomas and T. Troscianko. "Segmentation of Natural Images using Self-Organising Feature Maps". In British Machine Vision Conference, pages 223_232. British Machine Vision Association, September 1996.
- [Gath et al.,89] I. Gath and B. Geva. "Unsupervised Optimal Fuzzy Clustering". IEEE Trans. Pattern Analysis and Machine Intelligence", Vol.11, Nº7, pp.773-781, July 1989.
- [Kohonen, 97] T. Kohonen, "Self Organizing Maps", Springer, Berlin, 1997.
- [Lagan et al.,98] D.A. Lagan, J.W. Modestino and J.Zhang. "Cluster Validation for Unsupervised Stochastic Model-Bases Image Segmentation". IEEE Transactions on Image Processing, Vol. 7, Nº2, pp. 180-195, February, 1998
- [Littmann&Ritter, 97] E. Littmann and H. Ritter, "Adaptive Color Segmentation-A comparison of Neural and Statistical Methods", IEEE Transactions on Neural Networks, VOL 8, NO 1, January 1997.
- [Oliver et al.,00] N. Oliver, A. Pentland, F. Berard, "LAFTER: a real-time face and lips tracker with facial expression recognition". Pattern Recognition 33, pp:1369-1382, 2000.
- [Rissanen, 78] J. Rissanen. "Modeling by Shortest Data Description", Automatica, Vol.14, pp. 465-471, 1978.
- [Roberts et al., 98] S.J. Roberts, Dirk Husmeier, lead Rezek and William Penny. "Bayesian Approaches to Gaussian Mixture Modeling". IEEE Transactions on Image Processing, Vol. 20, Nº11, pp.1133-1142, November, 1998
- [Stiefelhaven et al., 97a] Stiefelhaven, J. Yang and A. Waibel "A model based gaze tracking system". IEEE International Joint Symposia on Intelligence and Systems-Image, Speech&Natural Language Systems. 1997
- [Yamazaki, 98] T. Yamazaki, "Introduction of EM algorithm into color image segmentation", Proc. ICIP'98, pp:368-371, August, 1998.
- [Yang&Waibel, 97], J. Yang, A. Waibel, "Skin-Color Modelong and Adaptation", Technical Report CMU-CS-97-146. School of Computer Science. Carnegie Mellon University, 1997.
- [Zhu et al., 00] X. Zhu, J. Yang, A. Waibel, "[Segmenting Hands of Arbitrary Color](#)", Fourth IEEE International Conference on Automatic Face and Gesture Recognition (F&G), pp: 446-453, 2000



2. Detección de discontinuidades

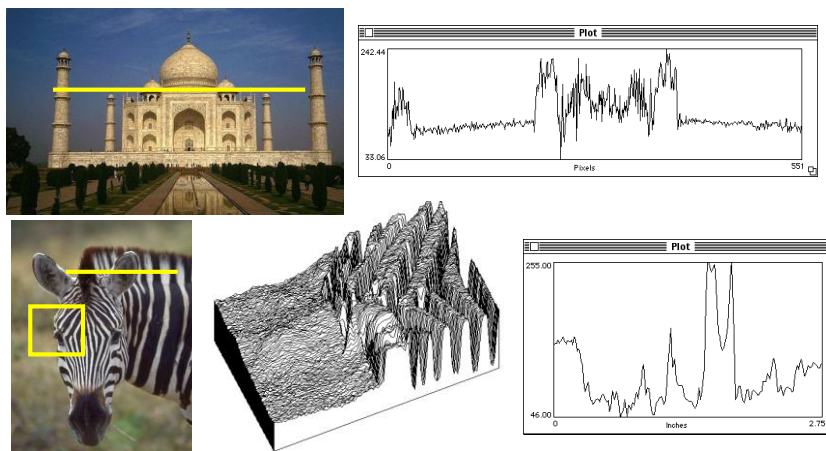
Detección de bordes

Objetivo

- ❑ Los **bordes** de la imagen son los cambios de intensidad significativos que hay en ella.
- ❑ Las causas suelen ser:
 - ❑ Discontinuidades en la superficie.
 - ❑ Discontinuidades en la profundidad.
 - ❑ Discontinuidades en el color o en la textura de la superficie.
 - ❑ Discontinuidades en la iluminación, sombras, reflexiones...
- ❑ Nuestro objetivo será crear algoritmos para la **detección de bordes (edges) significativos** de una imagen, ya que de estos bordes pueden extraerse características importantes de la imagen que nos ayuden en las siguientes etapas del Sistema de vision artificial (reconocimiento..).



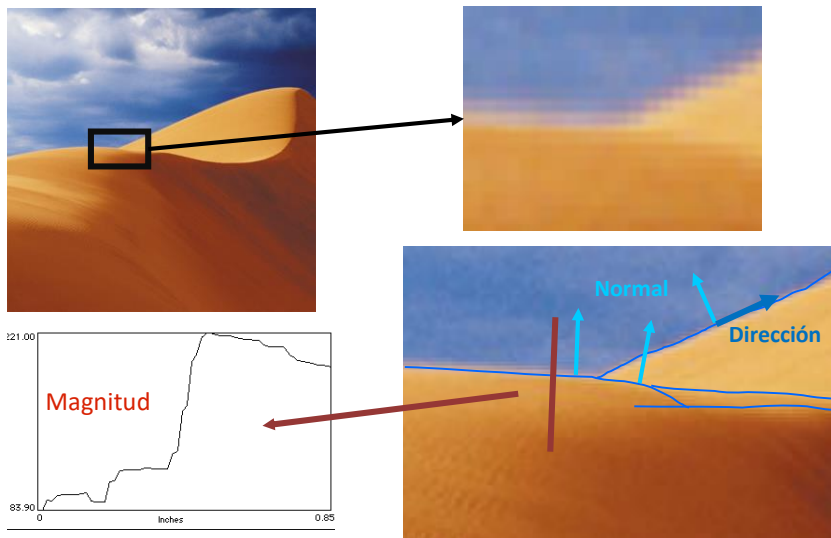
- Ejemplos donde se pueden ver los cambios de intensidad de las imágenes.

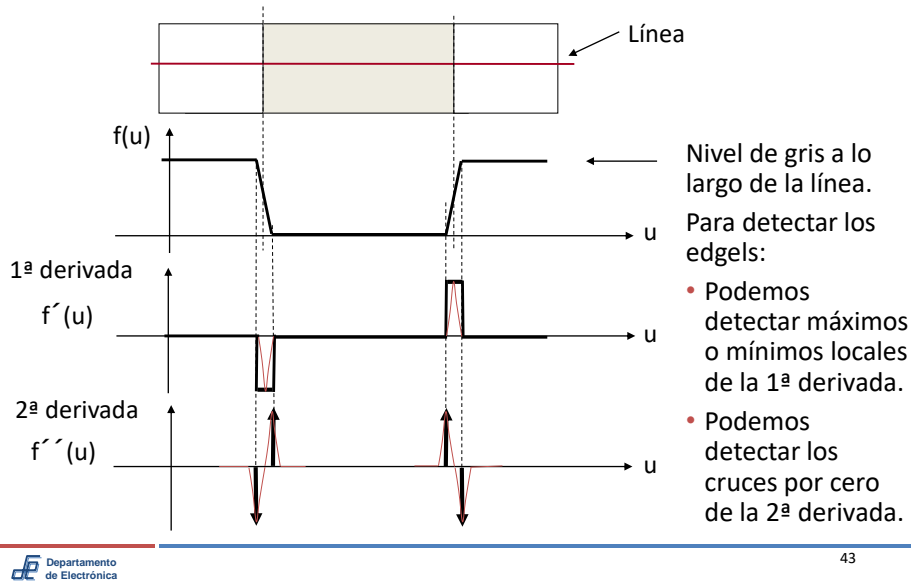


- No está claro qué es un *borde significativo*.
 - En parte definido por el contexto en el que se están aplicando los detectores de bordes.

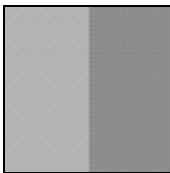


- ❑ Los **edgels** (*edge+pixels*) son los **píxeles** que forman parte de un borde ("edge").
- ❑ Un **borde local** implica que hay un cambio rápido en una imagen dentro de un área pequeña.
- ❑ Edgels son píxeles, **no** son contornos, límites ni líneas.
 - ❑ Los "edgels" sirven de soporte para definir contornos, límites o líneas.
 - ❑ Contornos, límites o líneas se construyen a partir de los "edgels".
- ❑ **Propiedades de los bordes:**
 - ❑ **Posición o centro:** posición de la imagen en la cual se localizan los bordes.
 - ❑ **Normal:** Vector unitario en la dirección de máxima variación de intensidad (máximo gradiente de intensidad).
 - ❑ **Dirección:** Vector unitario perpendicular al vector normal.
 - ❑ **Magnitud:** relacionado con el gradiente local (indica cómo de rápido varía la intensidad en borde en la dirección de la normal).

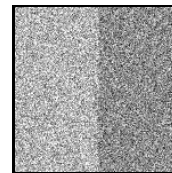
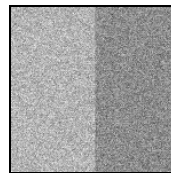
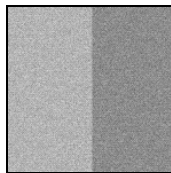




Borde ideal



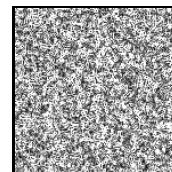
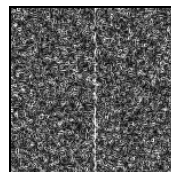
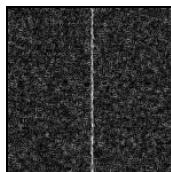
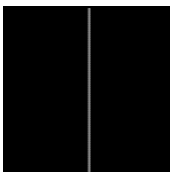
Bordes + ruido



Aumento de ruido



Bordes detectados



☐ **Eliminación de ruido**

- ☐ Suprimir todo lo que se pueda el ruido mientras se mantengan los bordes.
- ☐ En ausencia de otro tipo de información, se supone que el ruido es “blanco” con distribución gaussiana.

☐ **Realce de bordes**

- ☐ Diseñar filtros que respondan bien a los bordes: Filtros que tengan respuestas elevadas en donde existan bordes y respuestas bajas en el resto.

☐ **Localización de bordes**

- ☐ Determinar qué píxeles deberían ser descartados por representar ruido y cuáles se deben mantener.
 - ☐ Bordes delgados (de un píxel de ancho): máxima supresión.
 - ☐ Establecer un valor mínimo para considerar un máximo local de un filtro como un borde (umbralización).

☐ **Estimación de la primera derivada:**

- ☐ Detectores de bordes tipo “*gradiente*”.
- ☐ Detectores de bordes tipo “*orientación*” (*brújula o compás*).
- ☐ Detectores de bordes tipo “*Canny*”.

☐ **Segunda derivada:**

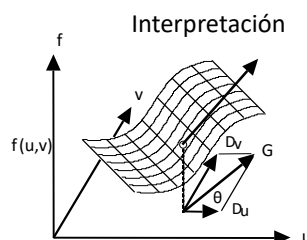
- ☐ Laplaciana.
- ☐ Laplaciana de la Gaussiana.

- El **gradiente** de una imagen $f(u,v)$ en un punto (u,v) se define como un **vector bidimensional** (vector perpendicular al borde):

$$G[f(u,v)] = \begin{bmatrix} G_u \\ G_v \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial u} f(u,v) \\ \frac{\partial}{\partial v} f(u,v) \end{bmatrix} \quad \left\{ \begin{array}{l} |G| = \sqrt{G_u^2 + G_v^2}, \quad |G| \approx |G_u| + |G_v| \\ \theta(u,v) = \tan^{-1} \frac{G_v}{G_u} \end{array} \right.$$

- Se considera que existe borde si la magnitud del gradiente **supera un determinado umbral** → !Se debe fijar un umbral T!:

$$g(u,v) = \begin{cases} 255 & \text{si } |G[f(u,v)]| > T \\ 0 & \text{si } |G[f(u,v)]| \leq T \end{cases}$$



- Las derivadas se pueden **aproximar** por:

$$G_u = \frac{\partial f(u,v)}{\partial u} \approx f(u,v) - f(u-1,v) \quad \xrightarrow{\text{máscara}} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline -1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad h_u(u,v)$$

$$G_v = \frac{\partial f(u,v)}{\partial v} \approx f(u,v) - f(u,v-1) \quad \xrightarrow{\text{máscara}} \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad h_v(u,v)$$

$$G_u = \frac{\partial f(u,v)}{\partial u} \approx \frac{[f(u+1,v-1) + 2f(u+1,v) + f(u+1,v+1)] - [f(u-1,v-1) + 2f(u-1,v) + f(u-1,v+1)]}{2} \quad \xrightarrow{\text{máscara}} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \quad h_u(u,v)$$

$$G_v = \frac{\partial f(u,v)}{\partial v} \approx \frac{[f(u+1,v+1) + 2f(u,v+1) + f(u-1,v+1)] - [f(u+1,v-1) + 2f(u,v-1) + f(u-1,v-1)]}{2} \quad \xrightarrow{\text{máscara}} \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad h_v(u,v)$$



$f(u,v)$



$g_1(u,v) = f(u,v) * h_u(u,v)$

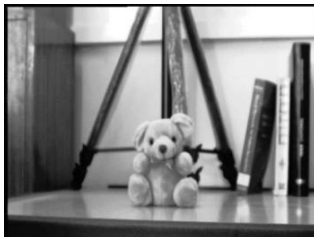


$g_2(u,v) = f(u,v) * h_v(u,v)$

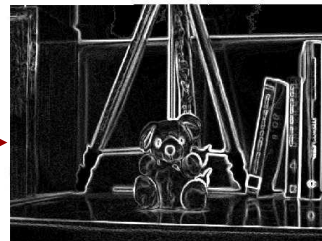


$g(u,v) = g_1(u,v) + g_2(u,v)$

$f(u,v)$



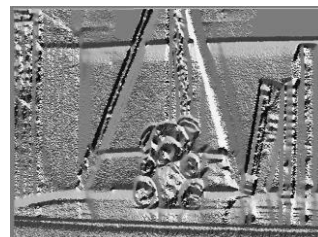
$g_m(u,v) = |G[f(u,v)]|$



$g(u,v)$, con
 $T = 30$



$\theta(u,v)$



Roberts

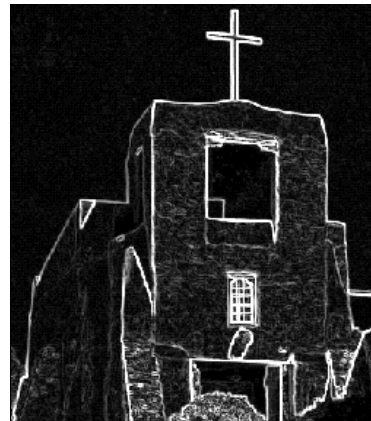
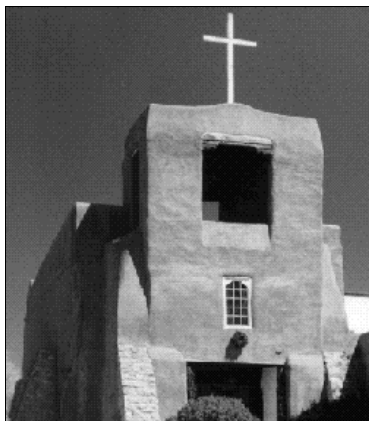
$$G_u = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G_v = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Prewitt

$$G_u = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad G_v = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Sobel

$$G_u = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



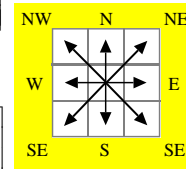
❑ Máscaras de Kirsch:

$$k_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, k_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}, k_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, k_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$0^\circ \qquad 45^\circ \qquad 90^\circ \qquad 135^\circ$

$$k_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, k_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}, k_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, k_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

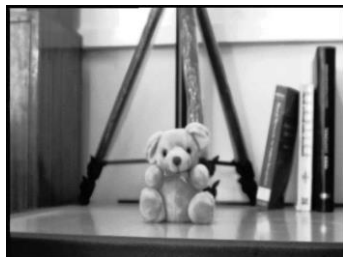
$180^\circ \qquad 225^\circ \qquad 270^\circ \qquad 315^\circ$



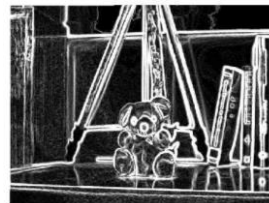
❑ Para cada píxel $f(u,v)$:

❑ $|G[f(u,v)]| = \max\{|k_0 * f(u,v)|, |k_1 * f(u,v)|, \dots, |k_7 * f(u,v)|\} = \max\{|k_i * f(u,v)|\}$

❑ $\Theta = \text{ángulo de la dirección correspondiente a } \max\{|k_i * f(u,v)|\}$



$|G|$



θ

