

- ❑ El **operador de Canny** se fundamenta en los operadores derivada pero introduce otras fases para mejorar la detección.
- ❑ Resulta especialmente interesante **porque extrae bordes y cierra los contornos**.
- ❑ Se desglosa en las siguientes fases:
 - ❑ **Suavizado** de la imagen original con filtro gaussiano (σ).
 - ❑ Obtención del **gradiente** ($|G|$ y θ) en cada píxel.
 - ❑ **Adelgazamiento** del ancho de los bordes, obtenidos con el gradiente, hasta lograr **bordes de un píxel de ancho** (supresión no máxima).
 - ❑ Umbralizamos: si los píxeles superan cierto umbral serán considerados como bordes. Pero hay problemas:
 - ❑ Si imponemos un umbral muy alto perdemos parte de los bordes,
 - ❑ Si usamos un umbral bajo aparecería ruido.

- ❑ **Solución: Histéresis de umbral.** Fijar **dos umbrales** T_1 y T_2 con $T_1 < T_2$
 - ❑ Umbralizar todos los puntos de $I_N(u,v)$ usando T_1 y T_2 y explorando en un orden fijo:
 - ❑ Localizar los puntos tal que:
 - ❑ $I_N(u,v) > T_2$ (**strong edge pixels**) y
 - ❑ $T_1 < I_N(u,v) < T_2$ (**weak edge pixels**)
 - ❑ Consideraremos **bordes válidos**:
 - ❑ todos los **píxeles fuertes** más
 - ❑ los **píxeles débiles que sean vecinos de los píxeles fuertes**.
 - ❑ De esta forma, obtenemos una detección mucho más **limpia** y en la que desaparecen muchos de los píxeles interiores a los bordes.
 - ❑ La salida es un conjunto de **bordes conectados de contornos** de la imagen junto a su **magnitud y orientación**, que describen las propiedades de los puntos de borde.
 - ❑ Frecuentemente, es común realizar un último paso consistente en **cerrar los contornos** que pudiesen haber quedado abiertos por problemas de ruido.

Ejemplos de aplicación sobre la misma imagen del filtro de Canny con diferentes valores para el filtro gaussiano (σ) y para los umbrales T_1 , T_2 .

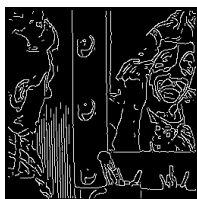
Imagen original



$\sigma = 1, T_2 = 255, T_1 = 1$



$\sigma = 1, T_2 = 255, T_1 = 220$



$\sigma = 1, T_2 = 128, T_1 = 1$



$\sigma = 2, T_2 = 128, T_1 = 1$



Probar distintos pasos en: <http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/>

$$\nabla^2 f(u,v) = \frac{\partial^2 f(u,v)}{\partial u^2} + \frac{\partial^2 f(u,v)}{\partial v^2} \approx f(u,v-1) + f(u-1,v) + f(u+1,v) + f(u,v+1) - 4f(u,v)$$

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$h = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$h = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

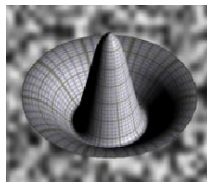


Máscara 5x5

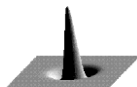
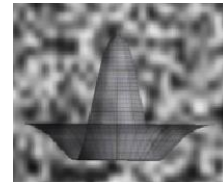


Máscara 9x9

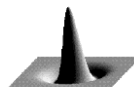
$$G(u,v) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(u^2 + v^2)}{2\sigma^2}\right] \longrightarrow \nabla^2 G(u,v) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{(u^2 + v^2)}{2\sigma^2}\right] \exp\left[-\frac{(u^2 + v^2)}{2\sigma^2}\right]$$



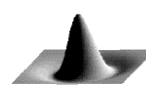
Operador
Sombrero Mexicano



$$\sigma^2 = 0.5$$



$$\sigma^2 = 1.0$$



$$\sigma^2 = 2.0$$

$$\nabla^2 \{f(u,v) * G(u,v)\} = \nabla^2 G(u,v) * f(u,v)$$

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$
 Un ejemplo:
máscara de 5x5

$$\begin{bmatrix} h_{11} & h_{12} & \dots & h_{1N} \\ h_{21} & h_{22} & \dots & h_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \dots & h_{NN} \end{bmatrix}$$
 Convolución
bidimensional

$$\nabla^2 \{f(u,v) * G(u,v)\} = G(u) * \left\{ f(u,v) * \frac{\partial^2}{\partial v^2} G(v) \right\} + G(v) * \left\{ f(u,v) * \frac{\partial^2}{\partial u^2} G(u) \right\}$$

Cuatro convoluciones unidimensionales

Máscara 13x13



❑ ¿Cómo se generan las máscaras bidimensionales?

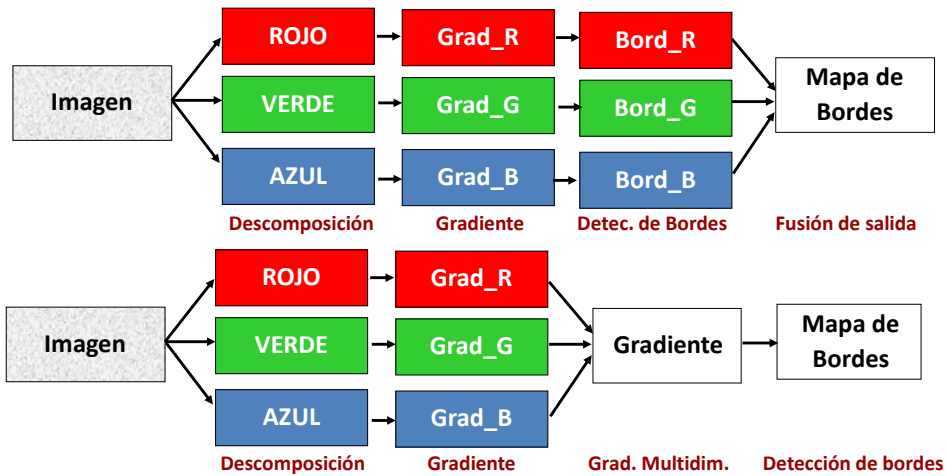
$$\nabla^2 G(u,v) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{(u^2 + v^2)}{2\sigma^2} \right] \exp \left[-\frac{(u^2 + v^2)}{2\sigma^2} \right]$$

1. Fijar el valor de la desviación típica (σ).
2. Determinar el valor de la ecuación anterior para los diferentes valores de (u,v): $u=0, 1, 2, \dots$ y $v=0, 1, 2, \dots$. Dada la simetría sólo hay que calcular en un cuadrante.
3. Escalar los valores y redondear los valores al entero más próximo.
4. Extender el ancho de la máscara de forma que contenga todos los valores distintos de cero.
5. Ajustar de forma simétrica los valores, mediante la adición o substracción de valores pequeños hasta conseguir que todos los valores de la máscara sumen cero.

Máscara de 17x17

0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0

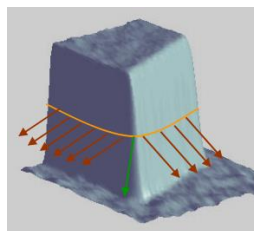
❑ Dos alternativas frecuentes:



Detección de esquinas

- ❑ En las esquinas **cambia la dirección del gradiente a lo largo del borde.**
- ❑ Un método habitual es el uso de **derivadas de segundo orden**, para medir la **razón de cambio de la dirección del gradiente** (E) con la **magnitud del gradiente** ("mg").
- ❑ Una esquina se declara como tal si: $E \geq T_1$
y/o $mg \geq T_2$, siendo T_1 y T_2 dos umbrales predeterminados.
- ❑ Detector de esquinas de Kitchen y Rosenfeld:

$$E = \frac{f_{uu} + f_v^2 + f_{vv} + f_u^2 - 2f_{uv}f_u f_v}{(f_u^2 + f_v^2)^{3/2}} \geq T_1$$



<http://www.disc.ua.es/tavarca/ponencias/esquinas.pdf>

$$f_u = \frac{\partial f(u,v)}{\partial u}, f_v = \frac{\partial f(u,v)}{\partial v}, f_{uu} = \frac{\partial}{\partial u} \left(\frac{\partial f(u,v)}{\partial u} \right), f_{vv} = \frac{\partial}{\partial v} \left(\frac{\partial f(u,v)}{\partial v} \right), f_{uv} = \frac{\partial}{\partial u} \left(\frac{\partial f(u,v)}{\partial v} \right)$$

Donde:

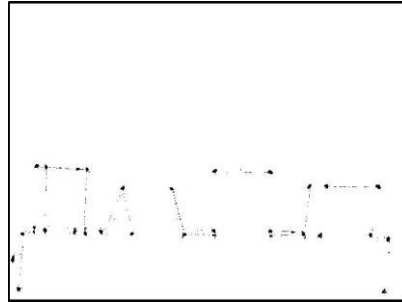
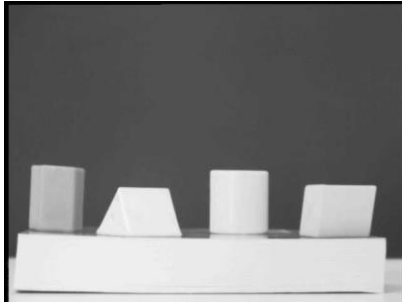
$$\frac{\partial}{\partial u} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial}{\partial v} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Por tanto:

$$f_u = \frac{\partial f(u,v)}{\partial u} = \left[\frac{\partial}{\partial u} \right] * f(u,v), f_v = \frac{\partial f(u,v)}{\partial v} = \left[\frac{\partial}{\partial v} \right] * f(u,v)$$

$$f_{uu} = \left[\frac{\partial}{\partial u} \right] * \left(\left[\frac{\partial}{\partial u} \right] * f(u,v) \right), f_{vv} = \left[\frac{\partial}{\partial v} \right] * \left(\left[\frac{\partial}{\partial v} \right] * f(u,v) \right), f_{uv} = \left[\frac{\partial}{\partial u} \right] * \left(\left[\frac{\partial}{\partial v} \right] * f(u,v) \right)$$

❑ Ejemplo de detección de esquinas

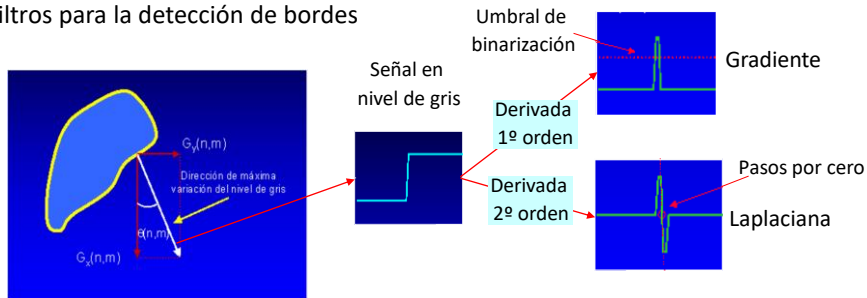


3. Segmentación Basada en Bordes

❑ Proceso:

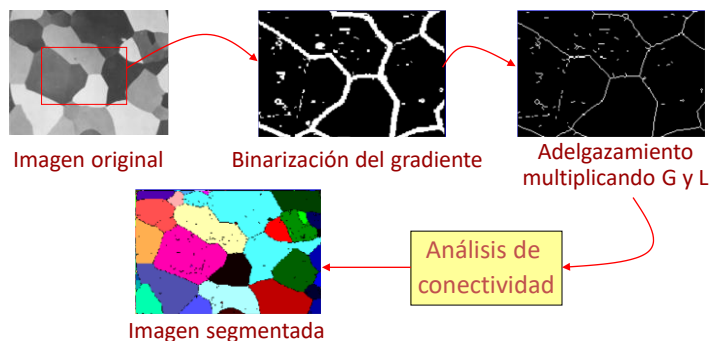
- ❑ Se pasa un filtro **paso alto** por la imagen para detectar los bordes.
- ❑ Se **umbraliza** la imagen de bordes.
- ❑ Se busca **conectividad** entre píxeles para detectar **contornos cerrados**.
- ❑ **Cada contorno cerrado define un objeto.**

Filtros para la detección de bordes



❑ Binarización de la imagen

- ❑ G: Gradiente
$$g_1(x, y) = \begin{cases} 1 & \text{si } G[I(x, y)] > T \\ 0 & \text{si } G[I(x, y)] \leq T \end{cases}$$
 - ❑ L: Laplaciana
$$g_2(x, y) = \begin{cases} 1 & \text{si } L[I(x, y)] = 0 \\ 0 & \text{si } L[I(x, y)] \neq 0 \end{cases}$$
- $\Rightarrow g(x, y) = g_1(x, y) \& g_2(x, y)$



❑ Ventajas:

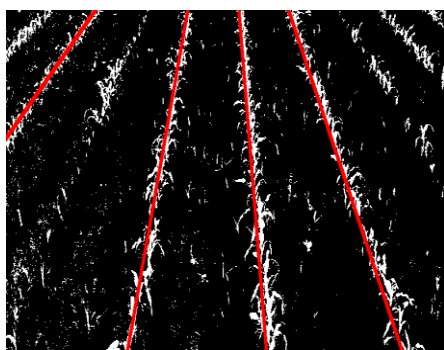
- ❑ Definición exacta de los contornos de los objetos
- ❑ Facilidad de implementación

❑ Inconvenientes:

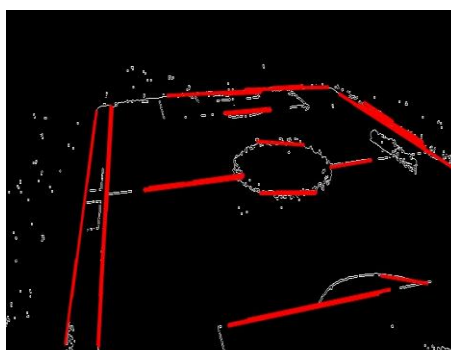
- ❑ Detección por umbral
- ❑ Análisis muy local: definición de regiones depende del cierre de contornos

Segmentación basada en bordes
Análisis Global (detección de líneas)

- ❑ Dado un conjunto local de elementos de borde (obtenidos mediante técnicas de detección de bordes), con o sin información de orientación ¿Cómo se pueden extraer las **líneas rectas** que forman el borde?
- ❑ Idea general:
 - ❑ Encontrar un espacio alternativo en el cual las líneas se “*mapean*” como puntos.
 - ❑ Cada elemento de borde “*vota*” por una línea recta a la cual puede pertenecer.
 - ❑ Aquellos puntos (pares: *pendiente, desplazamiento*) que en el espacio alternativo reciban un alto número de *votos* se corresponden con líneas en la imagen.

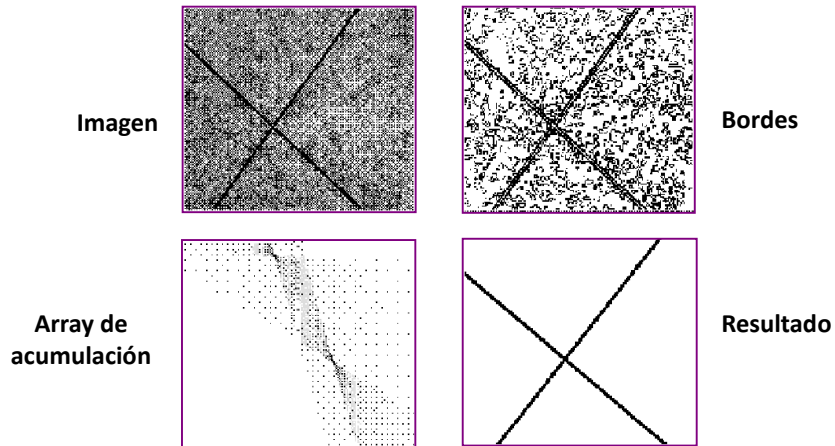


<https://unoyunodiez.files.wordpress.com/2012/03/output.png>

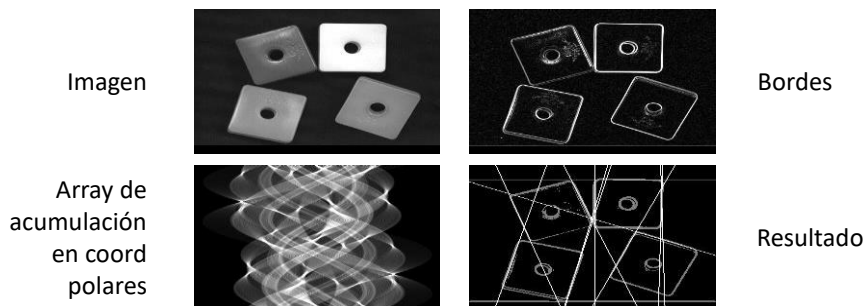


<https://nachorodriguezpfc.files.wordpress.com/2009/12/resultado18.jpg>

- Ejemplo de transformada de Hough en coordenadas cartesianas (cada punto de borde se mapea como una recta en el array de acumulación)

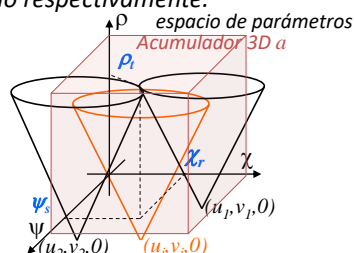
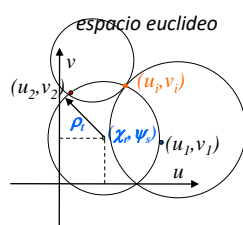


- Un problema de la representación cartesiana de la recta es que tanto la pendiente (a) como la ordenada en el origen (b) tienden a **infinito** conforme la recta se acerca a posiciones **verticales**.
- Para evitar este problema se usan las **coordenadas polares**. La forma de construir el **acumulador** en el plano ρ - θ es similar al primer algoritmo, la única diferencia está en que, **en vez de líneas rectas, se obtendrán curvas sinusoidales**.



□ Ecuación de la circunferencia: $h(x, p); (u - \chi)^2 + (v - \psi)^2 - \rho^2 = 0$

□ donde $p = (\chi, \psi, \rho)$ son el centro y el radio respectivamente.



- Las infinitas circunferencias que pasan por (u_p, v_p) en el espacio de parámetros se representan por curvas cónicas $(\chi - u_i)^2 + (\psi - v_i)^2 - \rho^2 = 0$
- El plano $\chi\psi\rho$ se cuantifica con un array de acumulación 3D (a).
- Un elemento $a(i, j)$ se incrementa en 1 si: $|(\chi - u_i)^2 + (\psi - v_i)^2 - \rho| < \varepsilon$
- Si las curvas cónicas tienen un punto de intersección se obtendrá un máximo en el acumulador (a) correspondiente a la circunferencia (χ_p, ψ_p, ρ_p)

□ Transformada de Hough: generalización a otras formas geométricas

□ **Detección de líneas.** Parámetros: (r, θ)

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

□ **Detección de círculos:** Parámetros: (r, x_0, y_0) 2 para el centro + radio

$$r^2 = (x - x_0)^2 + (y - y_0)^2$$

□ **Detección de elipses:** Parámetros: $(r_x, r_y, \theta, x_0, y_0)$ centro + orientación + eje mayor + eje menor

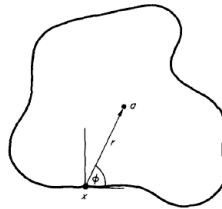
$$1 = \frac{(x - x_0)^2}{r_x^2 \cos^2 \theta} + \frac{(y - y_0)^2}{r_y^2 \sin^2 \theta}$$

□ **Ballard: Transformada de Hough generalizada**

- Transformada de Hough para la detección de elipses
 - 5 parámetros: eje mayor (r_x), eje menor (r_y), orientación (θ), centro (x_0, y_0)

$$1 = \frac{(x - x_0)^2}{r_x^2 \cos^2 \theta} + \frac{(y - y_0)^2}{r_y^2 \sin^2 \theta}$$

- Transformada de Hough generalizada (Ballard)
 - Generalización para formas no analíticas con lookup table

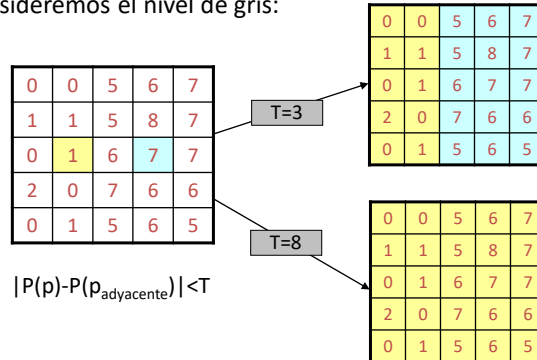


2. Segmentación basada en Regiones

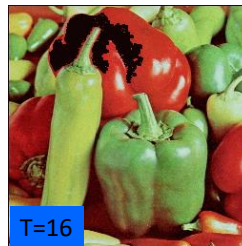
- ❑ La **segmentación en regiones** de una imagen, $f(u,v)$ representada mediante la región R , es un proceso de partición de R en un conjunto de K regiones $\{R_i\}$, $1 \leq i \leq K$, tal que:
 - ❑ La unión de todas las regiones forman la imagen $\rightarrow R = \bigcup_{i=1}^K R_i$
 - ❑ Ningún píxel pertenece a más de una region:
$$\rightarrow R_i \cap R_j = \emptyset \quad \forall i, j; i \neq j$$
 - ❑ Coherencia espacial
 $\rightarrow R_i$ es una región conectada
 - ❑ Coherencia de características \rightarrow Para algún predicado P :
$$P(R_i) \text{ es CIERTO para } i = 1, 2, \dots, K$$
$$P(R_i \cup R_j) \text{ es FALSO para } R_i, R_j \text{ adyacentes e } i \neq j$$

- ❑ **Método de crecimiento de regiones:** Para **segmentar** la imagen se comienza con un conjunto de puntos **semilla** y se van haciendo crecer las regiones a su alrededor, añadiendo a cada punto semilla aquellos píxeles vecinos que tengan **propiedades similares**.
- ❑ **Idea general:**
 - ❑ Un píxel se añade a una región si se cumplen dos condiciones:
 - ❑ El píxel debe ser **adyacente** a la región.
 - ❑ El píxel debe ser **similar** a los píxeles de esa región.
 - ❑ El proceso continua hasta que **no** se puedan añadir más puntos.
 - ❑ Se elige **otro punto de semilla**, no perteneciente a ninguna de las regiones previas, y el proceso se repite hasta que se segmenta la totalidad de la imagen.

- ❑ **Criterio de similitud:** ¿Cuándo se considera un píxel similar para ser añadido a una región?
- ❑ Cuando la **diferencia de características (predicados)** con respecto a los píxeles de la región es inferior a un determinado **umbral (T)**.
- ❑ Ej: consideremos el nivel de gris:



Punto semilla

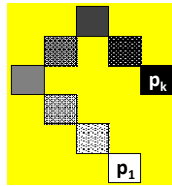


❑ Problemas:

- ❑ Selección de las semillas iniciales (número y ubicación)
- ❑ Criterio de similitud
 - ❑ Características a tener en cuenta
 - ❑ Fijación del umbral

❑ Debilidades:

- ❑ Deriva



Secuencia de píxeles:

p_1, \dots, p_k

p_j, p_{j+1} entorno

$|P(p_j) - P(p_{j+1})| < T$ pero $|P(p_1) - P(p_k)| > T$

- ❑ Los umbrales no tienen en cuenta las características de la distribución espacial global

- ❑ El método de **división y fusión de regiones (splitting and merging)**: divide la imagen en un conjunto de regiones y después las fusiona (*merge*) y/o divide (*split*) para satisfacer algún criterio de similitud. La fase de *merging* evita la **sobresegmentación** producida en la fase de *splitting*.

❑ Proceso:

- ❑ Partición inicial: 1 región (R)
- ❑ **Criterio de similitud: distancia** de características de los píxeles (predicados) respecto a un modelo

$$C = \sum_{p \in R_i} \|P(p) - M[P(R_i)]\|$$

p : píxel

$P(\cdot)$: predicados calculados

R_i : región

$M[\cdot]$: modelo de la región (media, ...)

- ❑ Si $C(R_i) = \text{falso}$, $C > T$ → “split”: dividir de forma geométrica



- ❑ Si $C(R_i \cup R_j) = \text{verdadero}$, $C < T$ → “merge”: fusionar R_i y R_j



- ❑ Parar cuando no sea posible realizar más fusiones ni divisiones.

□ Ejemplo de funcionamiento: $C_{R_i} = |I(p_j) - I(R_i)_{max}|; \forall p_j \in R_i \text{ (Th = 3)}$

Split if $C_{R_i} \geq Th$

Quadimages

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_R = 8$

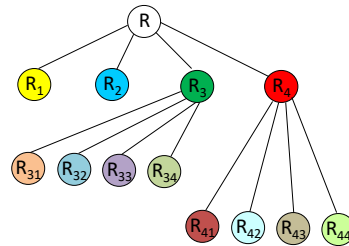
0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_1} = 2$ $C_{R_3} = 4$

$C_{R_2} = 1$ $C_{R_4} = 8$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

Quadtree



Merge if $C_{R_i} \cap C_{R_j} \leq Th$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_1} \cap C_{R_2} = 3$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_1} \cap C_{R_{43}} = 2$

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_1} \cap C_{R_{44}} = 3$

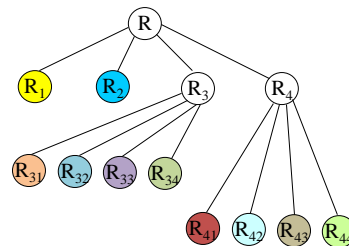
0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_{34}} \cap C_{R_{33}} = 1$

...

0	1	4	8
1	2	7	7
3	2	8	7
2	2	1	0

$C_{R_{34}} \cap C_{R_{41}} = 0$



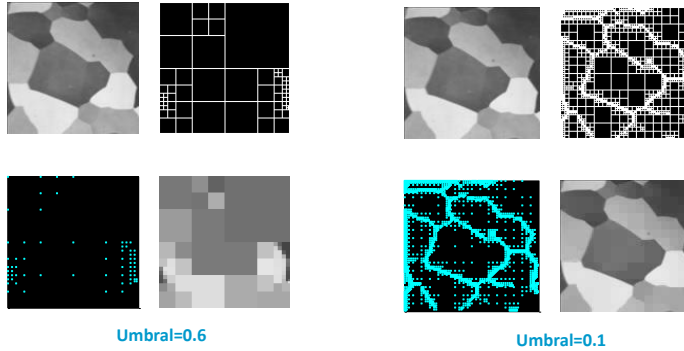
La fase de merging evita la sobresegmentación
producida en la fase de splitting

❑ **Ventajas:**

- ❑ Simple
- ❑ Sin partición inicial fija
- ❑ Visión global

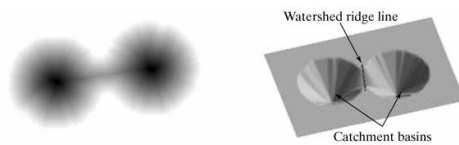
❑ **Inconvenientes:**

- ❑ Dependencia del umbral
- ❑ División puramente geométrica
- ❑ Contornos poco naturales

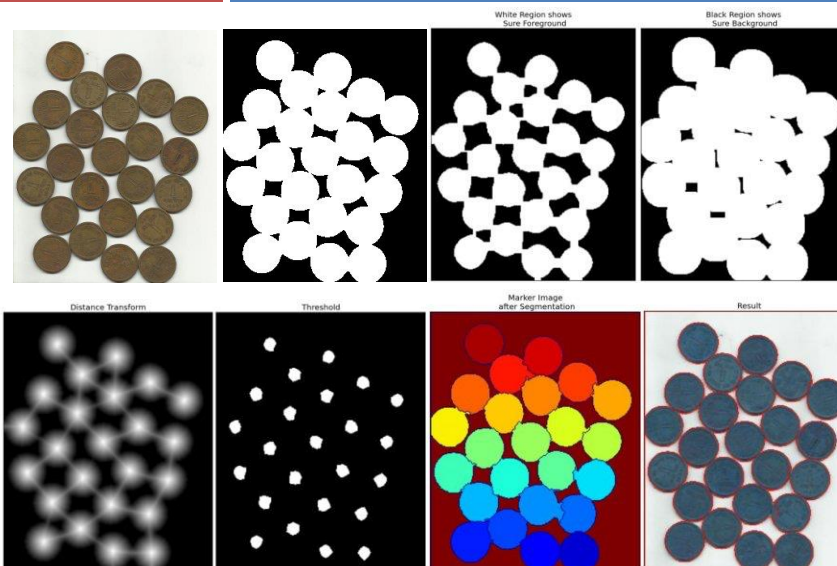


- ❑ <http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html> (Computer Vision Homepage CMU)
- ❑ <http://svr-www.eng.cam.ac.uk/milab.html> (Machine Intelligence Laboratory. Cambridge University)
- ❑ http://www.inria.fr/recherche/equipes/listes/theme_3.en.html (INRIA Vision)
- ❑ <http://www-white.media.mit.edu/vismod/> (MIT Media Lab)
- ❑ <http://www.cvc.uab.es> (Centro de Visión Artificial. Barcelona)

- ❑ **Transformada Watershed.** Es útil para separar **objetos que se tocan** en una imagen.
- ❑ Aplica conceptos **topológicos** para segmentar. Se puede interpretar como:
 - ❑ Se parte de una imagen donde su nivel de gris representa la distancia de cada pixel de la imagen original al borde más cercano del objeto o su inversa.
 - ❑ Niveles de gris altos indican picos mientras que niveles bajos indican valles.
 - ❑ Se llenan los valles (**catchment basins**) con agua de diferentes colores.
 - ❑ Cuando el agua subiera obviamente diferentes colores se fusionarían.
 - ❑ Para evitarlo se construyen barreras (**watershed ridge**).
 - ❑ Se termina cuando todos los picos están bajo el agua.
 - ❑ Las barreras creadas son el resultado de la segmentación.



- ❑ **Problema:** sobresegmentación (a veces, se segmentan muchos más objetos de los necesarios)
- ❑ **Solución** a la sobresegmentación: marcar a "grosso modo" los objetos y el fondo de cara a hacer la transformada.



☐ **Ventajas:**

- ☐ Buen funcionamiento para segmentar múltiples objetos que se tocan aunque todos tengan similares niveles de gris

☐ **Inconvenientes:**

- ☐ Método complejo con gran contenido heurístico
- ☐ Baja precisión en la posición de los contornos