

Procesamiento Digital de Imágenes

Unidad V: Nociones de segmentación

Departamento de Informática - FICH
Universidad Nacional del Litoral

7 de mayo de 2017



Temas a desarrollar

- Motivación.
- Relaciones entre píxeles:
- Detección de discontinuidades: puntos, líneas y bordes.
- Conexión de bordes:
 - Procesos locales.
 - Procesos globales: Transformada de Hough.
- Segmentación de regiones.
- Etiquetado de componentes conectadas.

Niveles de procesamiento



Motivación

- Dividir una imagen en sus regiones constituyentes y separar objetos de interés.



- Para imágenes no triviales, es una de las tareas más difíciles en procesamiento de imágenes.
- Su calidad condiciona el resultado de procesos de análisis posteriores.

Vecindad de un píxel

- Dado un píxel particular $p = (x, y)$, se definen 3 vecindades relativas a p :

- 4-vecinos (2 verticales y 2 horizontales):

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

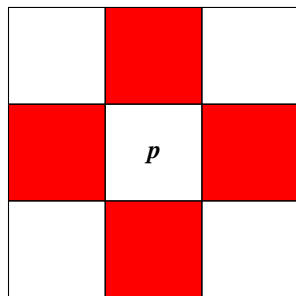
- Vecinos diagonales:

$$N_D(p) = \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\}$$

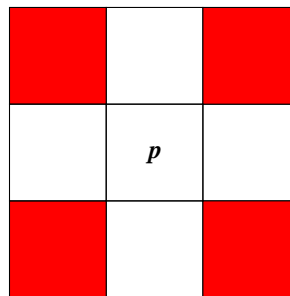
- 8-vecinos:

$$N_8(p) = N_4(p) \cup N_D(p)$$

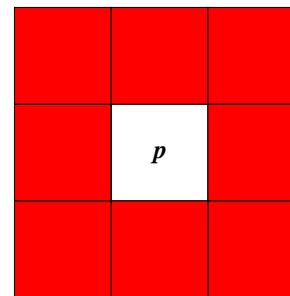
- Si p se encuentra en el borde de la imagen, algunos de sus vecinos se encuentran fuera de la misma.



$N_4(p)$



$N_D(p)$



$N_8(p)$

Detección de discontinuidades

- Discontinuidad: cambio abrupto en las intensidades de gris.
- Tipos básicos de discontinuidades:
 - Puntos
 - Líneas
 - Bordes
- Forma de detección: filtrado espacial con una máscara apropiada (ej. $w_{3 \times 3}$):

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i$$

donde w_i es el coeficiente de la máscara que afecta al valor de intensidad z_i .

- Luego se aplica umbralización para binarizar el resultado.

Detección de puntos: Procedimiento

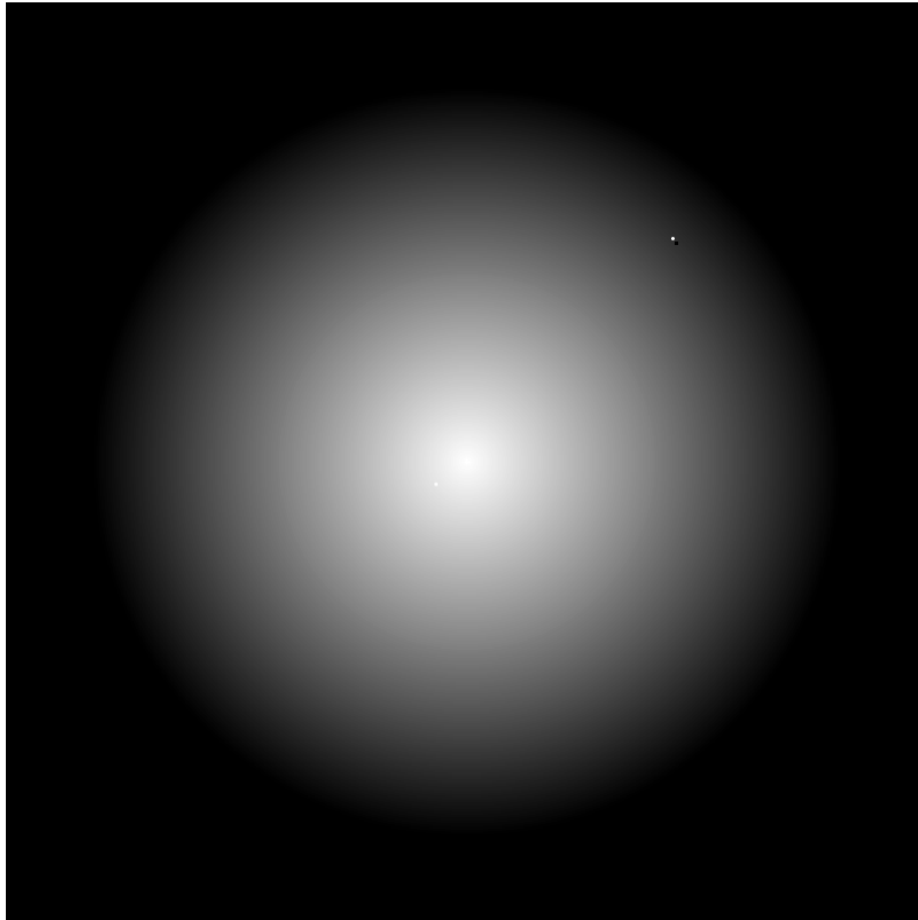
- Objetivo: Detectar puntos aislados en zonas homogéneas.
- Se aplica una máscara correspondiente a un pasa altos con suma cero:

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

- Luego se umbraliza el resultado a partir de un umbral T especificado:

$$|R| \geq T$$

Detección de puntos: Ejemplo



Original



Detección de puntos

Detección de líneas: Procedimiento

- Objetivo: Detectar líneas de grosor de un píxel con distintas orientaciones.
- Máscaras direccionales con suma 0:

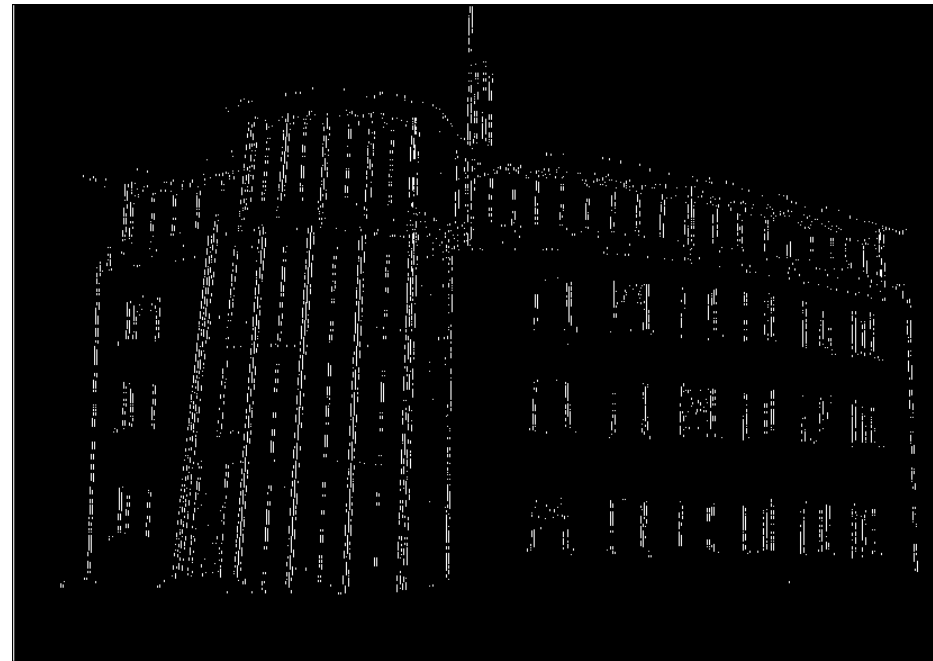
$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

- Umbralización del resultado.
- Permite detectar líneas en una determinada dirección (aplicación de una sola máscara) o en todas direcciones (suma del resultado de todas las máscaras).

Detección de líneas: Ejemplo



Original



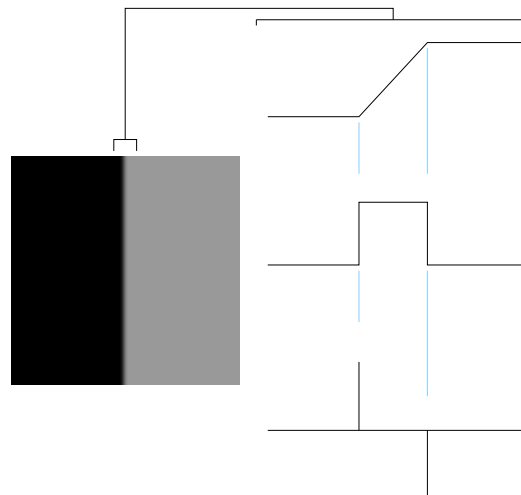
Líneas verticales

Detección de bordes: Conceptos

- Borde ideal (perfil=escalón) vs. borde real (perfil=rampa):



- Modelo real: el borde está formado por todos los píxeles conectados contenidos en la rampa.
- El grosor del borde depende de la pendiente de la rampa (grado de borrosidad).



Detección de bordes: Métodos

- Método de la primera derivada:
 - La magnitud de la primera derivada detecta la presencia de borde en un punto.
 - Detección mediante umbralización de la derivada primera. Los puntos donde la misma sea mayor que un umbral pertenecen al borde.
- Método de la segunda derivada:
 - La segunda derivada produce una respuesta doble (indeseable).
 - El signo de la segunda derivada detecta si un punto de borde está del lado claro u oscuro del borde.
 - El cruce por cero de la línea imaginaria que une los valores máximo y mínimo de la derivada segunda se encuentra aproximadamente en el centro del borde.

Derivada primera: operadores gradiente

- El gradiente de una imagen $f(x,y)$ en el punto (x,y) se define como el vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Magnitud del gradiente:

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = [G_x^2 + G_y^2]^{1/2} \approx |G_x| + |G_y|$$

- Dirección del gradiente:

$$\alpha(x,y) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

- La dirección del gradiente es perpendicular a la dirección del borde en un punto.

Operadores de gradiente cruzado de Roberts

- Dada una vecindad de la imagen:

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

- Las componentes del gradiente se aproximan mediante:

$$G_x = z_9 - z_5$$

$$G_y = z_8 - z_6$$

- Las máscaras que lo implementan son:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Operadores de Prewitt

- Las componentes del gradiente se aproximan mediante:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

- Las máscaras que lo implementan son:

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Respuesta nula en zonas de gris constante.

Operadores de Sobel

- Las componentes del gradiente se aproximan mediante:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

- Las máscaras que lo implementan son:

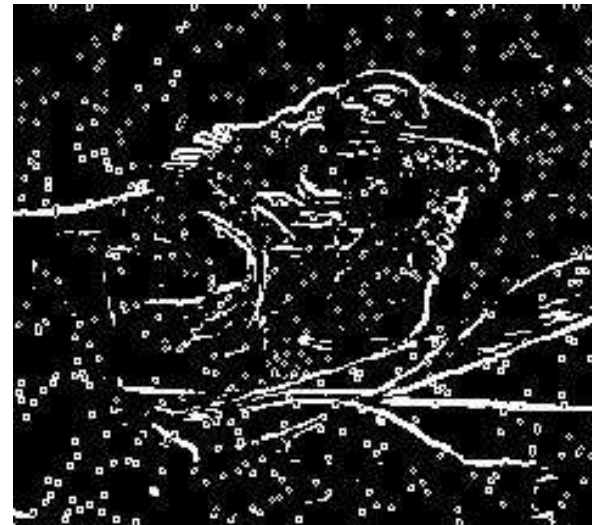
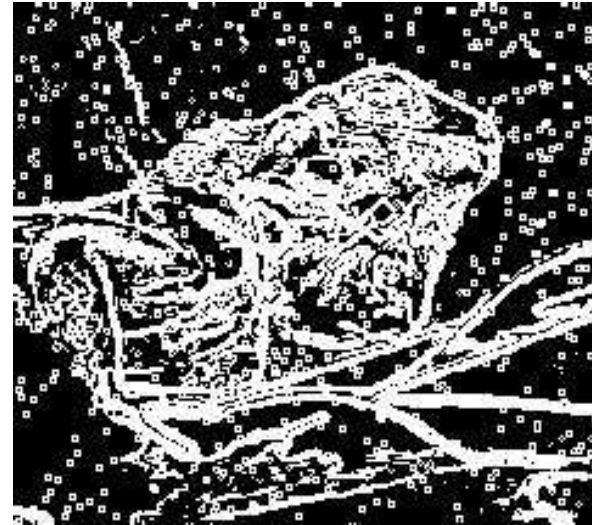
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

- Enfatiza los píxeles más cercanos al centro, consiguiendo una mejor respuesta en presencia de ruido tipo gaussiano.
- Respuesta nula en zonas de gris constante.

Comportamiento en presencia de ruido

- Alta sensibilidad de los operadores derivada a la existencia de ruido.
- Relación de compromiso entre sensibilidad al ruido y exactitud en la detección de bordes.
- Parámetros de control: talla del filtro, nivel del umbral.
- A mayor talla del filtro o del nivel del umbral se consigue una mayor remoción de ruido, pero se pierden bordes.
- Necesidad de eliminar ruido en etapas previas de procesamiento.

Comportamiento en presencia de ruido



Derivada segunda: el Laplaciano

- Laplaciano de una función 2-D:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Aproximación considerando N_4 :

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

- Máscara utilizada:

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

Derivada segunda: el laplaciano

- Aproximación considerando N_8 :

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$$

- Máscara utilizada:

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

- Problemas del laplaciano:
 - Excesivamente sensible al ruido.
 - Produce bordes dobles.
 - No detecta dirección de los bordes.
 - Se requiere determinar los cruces por cero.
 - Su utilidad se limita a clasificar los puntos que pertenecen a la zona clara y a la zona oscura a cada lado del borde.

Derivada segunda: Laplaciano del Gaussiano (LoG)

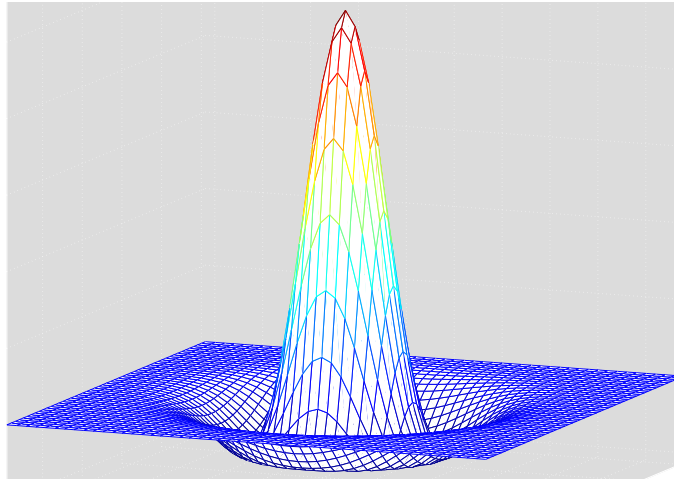
- Suavización mediante filtro gaussiano seguida de aplicación del laplaciano:

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}, \text{ con } r^2 = x^2 + y^2$$

- Calculando el laplaciano se obtiene el LoG:

$$\nabla^2 h(r) = - \left[\frac{r^2 - \sigma^2}{\sigma^4} \right] e^{-\frac{r^2}{2\sigma^2}}$$

Derivada segunda: Laplaciano del Gaussiano (LoG)

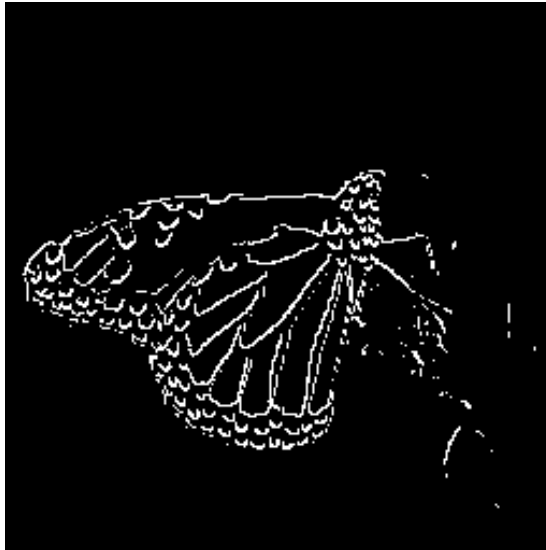


- Ejemplo de máscara utilizada:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Detectores de bordes: comparación

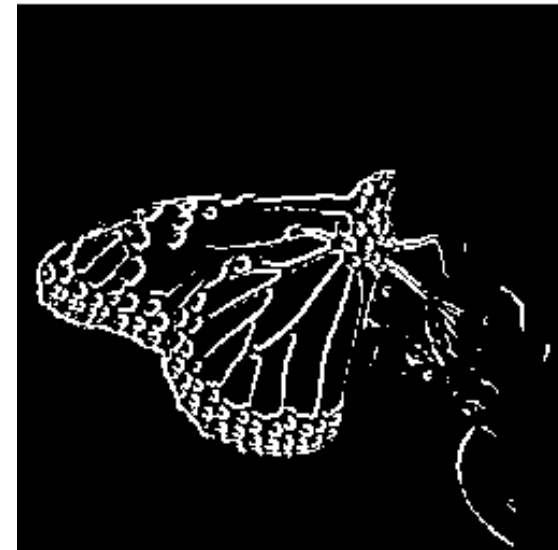
Roberts



Prewitt



Sobel



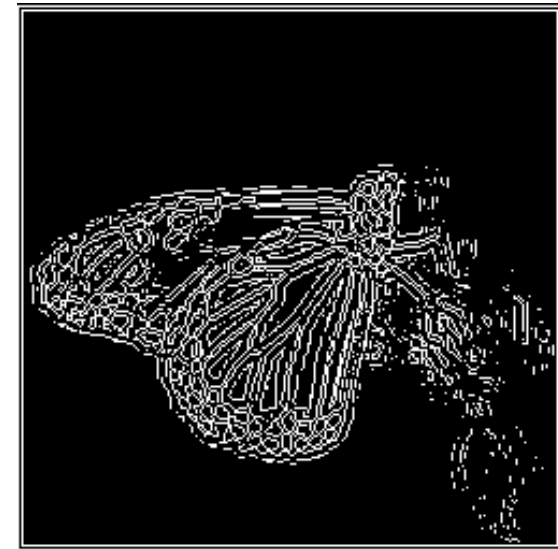
Laplaciano



LoG



Cruces por cero



Conexión de bordes: Proceso local

- De forma ideal, los puntos detectados con los métodos anteriores deberían todos pertenecer a los bordes.
- En la práctica, distintos factores (ruido, condiciones de iluminación, etc.) causan la detección de puntos espúreos y bordes discontinuos.
- Necesidad de conectar los puntos y segmentos detectados para encontrar los límites de los objetos de interés.
- Método: Recorrer la imagen de magnitud del gradiente, analizando pequeños entornos (ej. 3x3). Los puntos del entorno que tienen una magnitud y dirección de gradiente similar se etiquetan como pertenecientes al mismo borde. Por ej., el píxel de borde en (x_0, y_0) perteneciente a la vecindad de un píxel de borde (x, y) se conecta al mismo si se verifican estas dos condiciones:

$$\begin{aligned} |\nabla f(x, y) - \nabla f(x_0, y_0)| &\leq T_1 \\ |\alpha(x, y) - \alpha(x_0, y_0)| &< T_2 \end{aligned}$$

donde T_1 y T_2 son dos umbrales no negativos.

Búsqueda de líneas rectas por Transf. de Hough

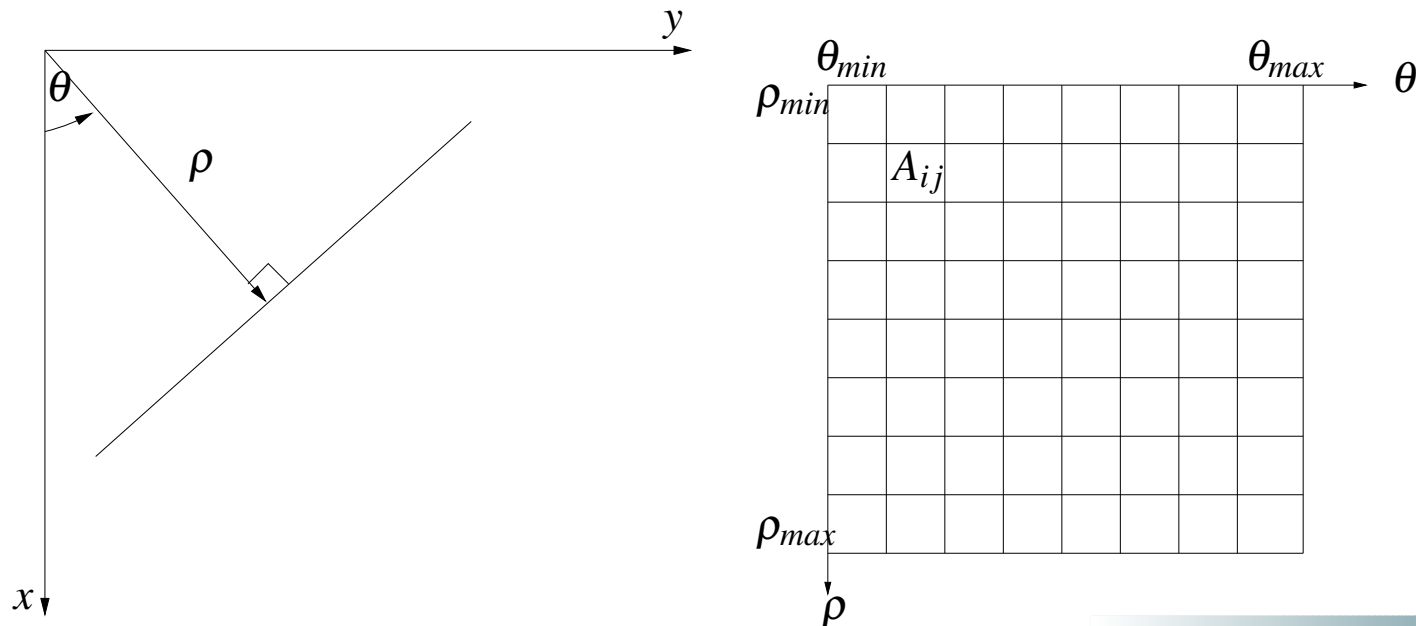
- Representación de la recta en coordenadas polares:

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

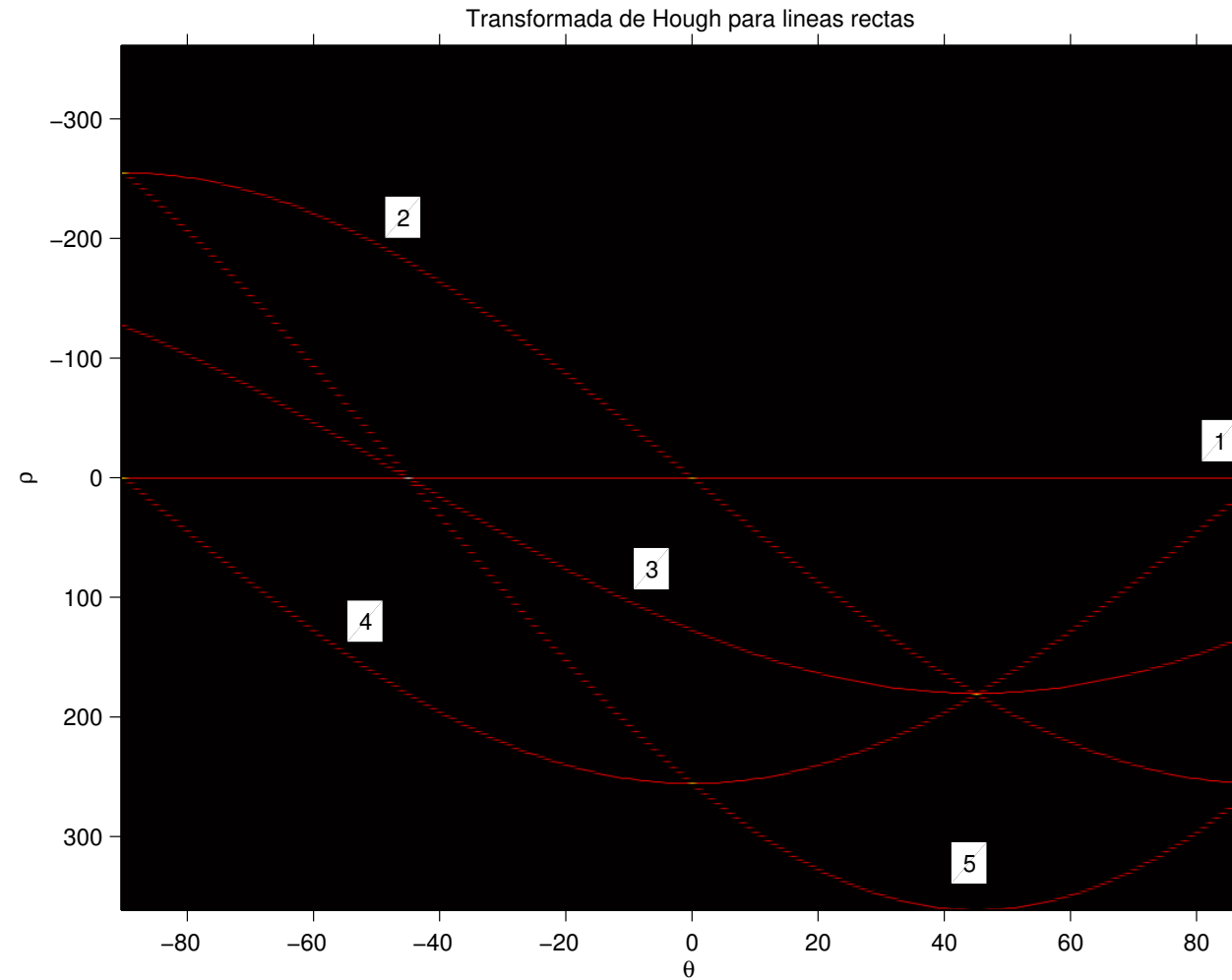
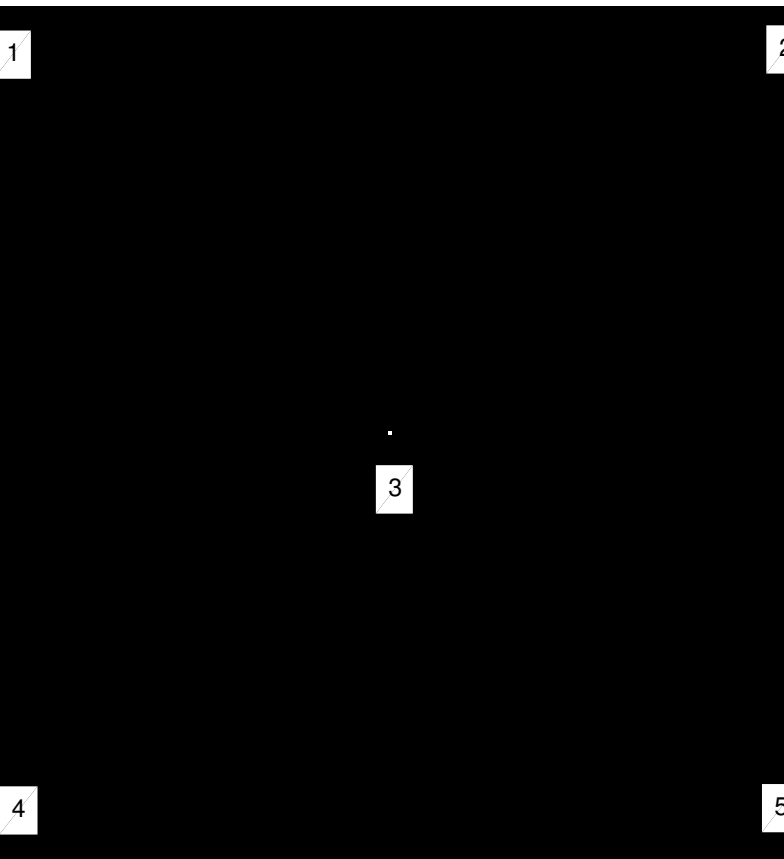
$$\text{con } -90^\circ \leq \theta \leq 90^\circ, -\sqrt{2}D \leq \rho \leq \sqrt{2}D$$

siendo D igual a la distancia entre esquinas de la imagen.

- N puntos pertenecientes a la recta $x \cos \theta_j + y \sin \theta_j = \rho_i$ producen N curvas sinusoidales que se intersectan en (ρ_i, θ_j) en el espacio de parámetros.



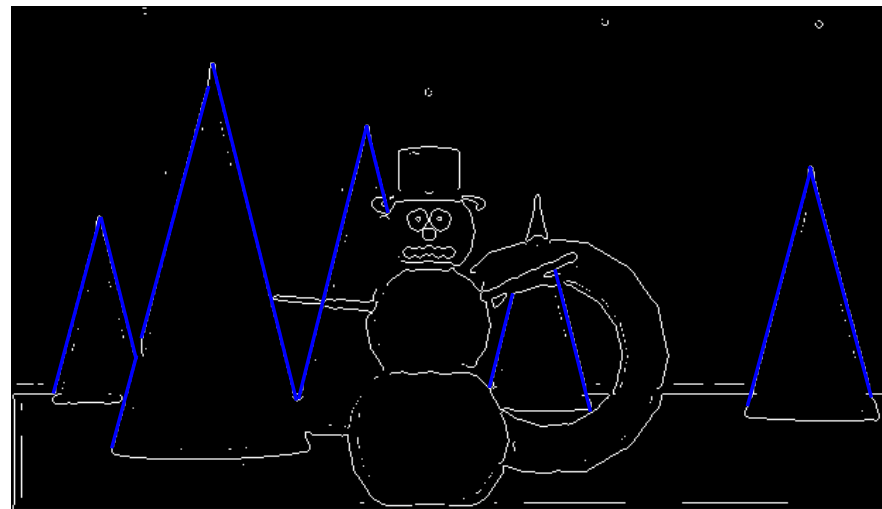
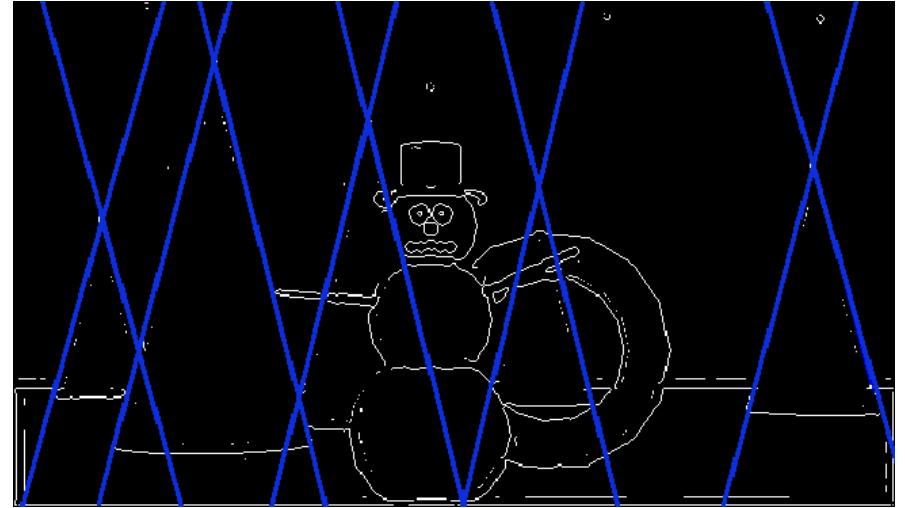
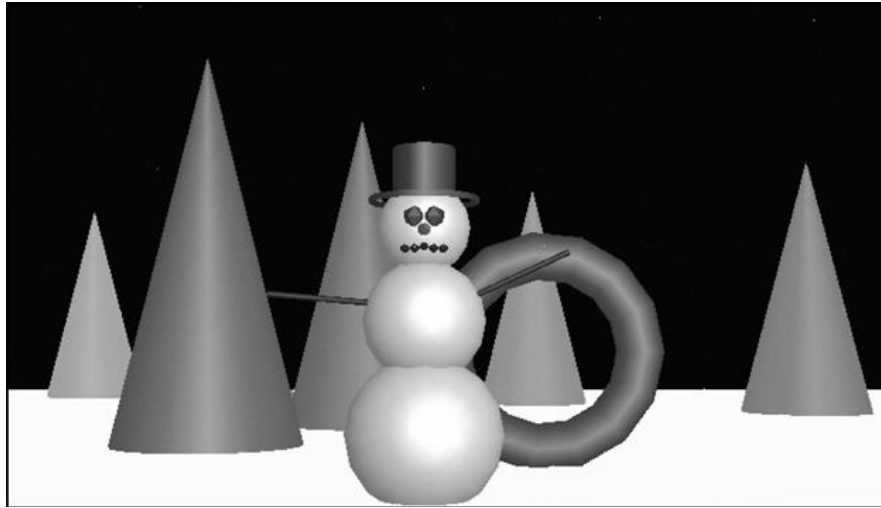
Conexión de bordes: Transformada de Hough



Transformada de Hough: Algoritmo

- Algoritmo:
 - Calcular el gradiente de la imagen y umbralizar el resultado.
 - Generar el espacio de parámetros mediante valores equiespaciados de ρ y θ , inicializándolo en cero (acumulador).
 - Para cada punto de borde de la imagen, calcular la curva correspondiente en el espacio de parámetros para cada θ_j , despejando el valor de ρ_i . Incrementar el acumulador en la posición (ρ_i, θ_j) .
 - La intersección de curvas en el espacio $\rho - \theta$ (altos valores en el acumulador) identifica la colinearidad de los puntos a los cuales corresponden tales curvas.
 - Conectar los puntos colineales cercanos (adyacentes o cuya distancia entre sí sea inferior a un umbral).

Transformada de Hough: Ejemplo



Segmentación basada en regiones

- Sea R la región correspondiente a una imagen completa. Región: conjunto de píxeles vecinos que cumplen una condición de semejanza de grises.
- El proceso de segmentación consiste en dividir R en subregiones $R_1, \dots, R_i, \dots, R_n$ tales que se verifiquen las siguientes condiciones:

1.
$$\bigcup_{i=1}^n R_i = R$$

2. R_i es una región conectada $\forall i$.

3. $R_i \cap R_j = \emptyset, \forall i \neq j$.

4. R_i cumple una cierta propiedad $P(R_i), \forall i$.

5. Dos regiones R_i y R_j verifican P , pero no la unión de ambas.

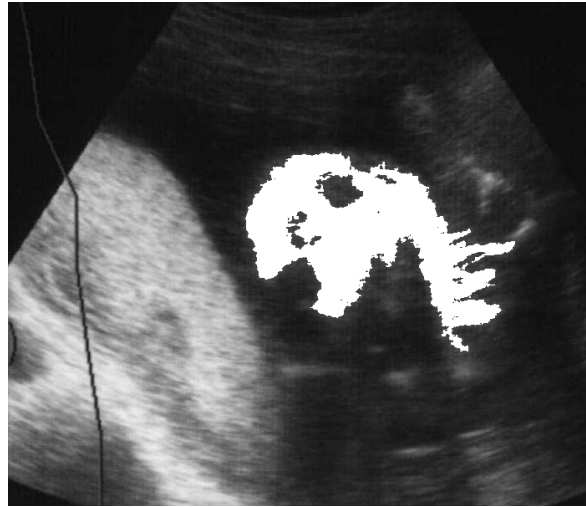
Algoritmo de crecimiento de regiones

- Consiste en agrupar píxeles o subregiones en regiones más grandes, basándose en propiedades que tengan en común (por ej. medida o criterio de similitud de los niveles de gris).
- Los píxeles deben cumplir con algún criterio determinado de conectividad o adyacencia.
- La técnica se inicia a partir de píxeles iniciales (**semillas**) y se van incorporando los píxeles vecinos que cumplen con la propiedad P seleccionada.
- El proceso termina cuando no existen más píxeles que cumplan P o cuando se verifica algún criterio especificado (tamaño máximo de la región, forma, etc.).

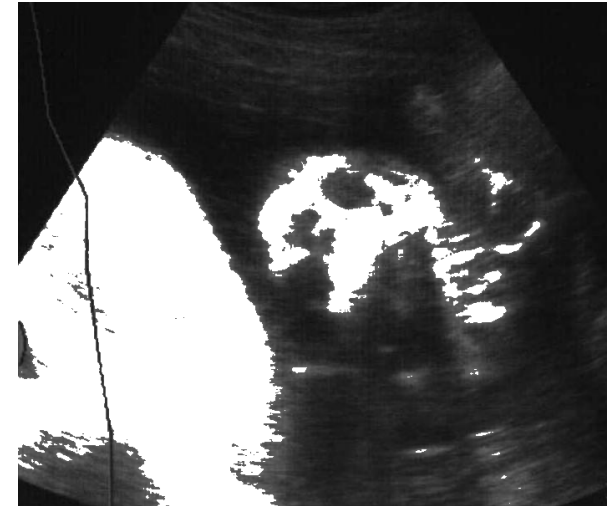
Crecimiento de regiones: Ejemplo



Original



Segmentada



Sólo umbralizada

Criterios de homogeneidad

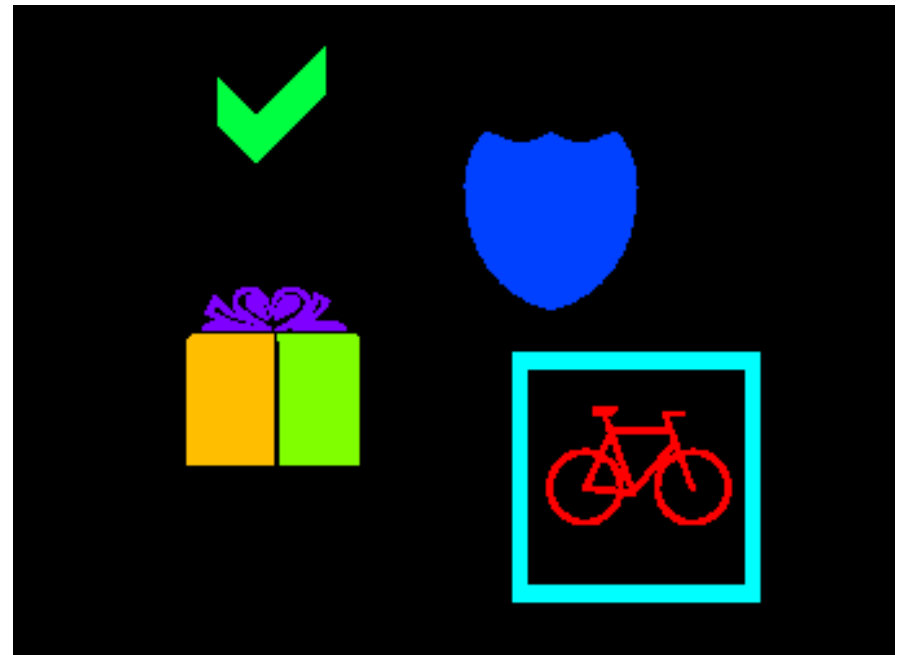
Gran diversidad, depende de la aplicación y de conocimientos previos. Algunos ejemplos:

Una región es considerada homogénea si:

- los niveles de gris son constantes;
- la media local es mayor que la media global;
- el desvío estándar local es menor que un cierto porcentaje de la media global;
- toda la región presenta la misma textura;
- cumple un criterio definido en función del color de la región;
- otros...

Etiquetado de componentes conectadas

- Componente conectada: conjunto de píxeles conectados (vecinos y semejantes).
- Etiquetar objetos en una imagen binaria asignando a los píxeles que los componen un mismo valor de gris/color (etiqueta).



Etiquetado de componentes conectadas: Ejemplo

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 1 1 0 1 0
0 1 1 1 1 1 1 0 0 1 0 0 1 0
0 0 0 0 1 0 1 0 0 0 0 0 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 1 1 1 1 1 1 1 1 1 0
0 1 1 0 0 0 1 0 1 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Imagen

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 2 0 0 3 3 0 4 0
0 5 5 5 2 2 2 0 0 3 0 0 4 0
0 0 0 0 5 0 2 0 0 0 0 0 4 0
0 6 6 5 5 5 2 2 2 2 2 4 4 0
0 0 0 0 5 5 5 2 2 2 2 2 4 0
0 7 7 0 0 0 5 0 2 0 0 2 2 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Primera pasada

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 2 0 0 1 1 0 2 0
0 2 2 2 2 2 2 0 0 1 0 0 2 0
0 0 0 0 2 0 2 0 0 0 0 0 2 0
0 2 2 2 2 2 2 2 2 2 2 2 2 0
0 0 0 0 2 2 2 2 2 2 2 2 2 0
0 3 3 0 0 0 2 0 2 0 0 2 2 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Segunda pasada

Tabla de equivalencias: 2-5, 5-6, 2-4

Etiquetado de componentes conectadas

Primera pasada:

- Recorrer la imagen R entera fila por fila, asignando una etiqueta entera $v \neq 0$ a cada píxel $R(i, j) \neq 0$.
- El valor de v depende de los vecinos de $R(i, j)$ de la siguiente manera:
 - Si todos los vecinos pertenecen al fondo (etiquetas iguales a 0), $R(i, j)$ es etiquetado con un valor aún no usado de v .
 - Si existe sólo un píxel vecino con etiqueta distinta de 0, se le asigna esta etiqueta a $R(i, j)$.
 - Si existe más de un píxel vecino con etiqueta distinta de 0, se le asigna a $R(i, j)$ la etiqueta de cualquiera de ellos. Si a su vez las etiquetas de los vecinos son distintas entre sí, se almacena la equivalencia entre etiquetas en una tabla.

Etiquetado de componentes conectadas

Segunda pasada:

- Todos los píxeles fueron etiquetados en la primera pasada, pero debido a posibles colisiones puede haber regiones con distintas etiquetas asignadas a sus píxeles.
- Se recorre la imagen nuevamente re-etiquetando los píxeles según la tabla de equivalencias.

Algunos comentarios:

- La vecindad puede corresponder a 4 u 8 vecinos.
- Es aconsejable utilizar valores de etiquetas incrementales. El número de componentes se obtiene de manera directa.
- El algoritmo puede extenderse a imágenes en grises.
- Implementación alternativa: crecimiento de regiones.

Fin de teoría

- Próxima teoría: Unidad VI - Morfología binaria