

UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS
PROCESAMIENTO DIGITAL DE IMÁGENES

Resumen de Teoría - Parcial II

Autores:

Mario ROSALES
Franco SANTELLÁN
Carlos GENTILE

E-mail:

mariorosales941@gmail.com
fransantellan@gmail.com
csgentile@gmail.com

JUNIO, 2017



Índice

1. UNIDAD 4 - Restauración de Imágenes	2
1. Motivación	2
2. Estudio de los Modelos Posibles	2
2.1. Modelos de Degradación	2
2.2. Modelos de Ruido	2
3. Restauración Mediante Filtrado Espacial	4
3.1. Filtros de Medias	4
3.2. Filtros de Orden	6
3.3. Filtros Adaptativos	9
4. Ruido Periódico y Filtrado Frecuencial	9
4.1. Ruido Periódico	9
4.2. Restauración Mediante Filtrado Frecuencial	10
5. Estimación de la Degración	11
5.1. Estimación de H	11
5.2. Filtrado Inverso	12
6. Filtrado de Mínimos Cuadrados	13
7. Filtro de la Media Geométrica	14
2. UNIDAD 5 - Segmentación	15
1. Detección de Discontinuidades	15
1.1. Detección de Puntos	15
1.2. Detección de Líneas	15
1.3. Detección de Bordes	16
2. Conexión de bordes: Proceso local	19
3. Segmentación basada en regiones	20
4. Criterios de homogeneidad	21
5. Etiquetado de componentes conectadas	21
3. UNIDAD 6 - Morfología	22
1. Usos de la Morfología Matemática	22
2. Contenido	22
3. Morfología matemática en imágenes	22
4. Erosión Binaria	23
5. Dilatación binaria	24
6. Cierre	25
7. Apertura	25
8. Propiedades	26
9. Transformación de localización (Hit-or-Miss)	26
10. Algoritmos	27
10.1. Extracción de contornos (gradiente morfológico)	27
10.2. Relleno de agujeros semi-automático (dilatación condicionada)	28
10.3. Extracción de componentes conectadas	28
10.4. Envoltura convexa (Convex Hull)	28
10.5. Adelgazamiento (Thinning)	29
10.6. Espesamiento (Thickening)	29
10.7. Esqueletos (Skeletons)	30
10.8. Reconstrucción morfológica	30
10.9. Aplicaciones de Reconstrucción Morfológica	32

1. UNIDAD 4 - Restauración de Imágenes

1. Motivación

El **Realce** o mejoramiento de imágenes tiene como objetivo manipular la imagen para ajustarla a la psicofísica del sistema visual humano, a partir de la aplicación de heurísticas y procesamiento subjetivo.

La **Restauración**, por su parte, tiene como objetivo recuperar una imagen que ha sido degradada por algún fenómeno, como por ejemplo: contaminación por ruido, problemas de enfoque, distorsiones geométricas o artefactos de movimiento. Esto se logra a partir de la aplicación de modelos que ajustan la degradación y procesos inversos para recuperar la imagen original, y procesamiento objetivos.

2. Estudio de los Modelos Posibles

2.1. Modelos de Degradación

La imagen original es afectada por una **función de degradación** mas el aporte de **ruido**.

$$\text{Dominio Espacial} \rightarrow g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$\text{Dominio Frecuencial} \rightarrow G(u, v) = H(u, v)F(u, v) + N(u, v)$$

En la figura (1), se puede esquematizar el modelo de degradación planteado.

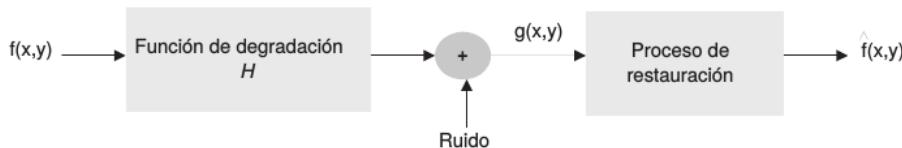


Figura 1: Modelo de Degradación.

Una forma de abordar el degradado es suponer $H = 1$ y estudiar el ruido, luego estudiamos H y la restauración en presencia de ambos componentes.

2.2. Modelos de Ruido

El **Ruido** es información no deseada que contamina la imagen. Algunas fuentes de ruido suelen darse en la adquisición de la imagen, como defectos en sensores, temperatura, ruido de línea en el sistema de captura, etc. O pueden darse en la transmisión de la imagen, como las interferencias en el canal.

Asumimos que el ruido es independiente de las coordenadas espaciales (salvo el ruido periódico) y que además no está correlacionado con la imagen. El ruido queda descripto mediante variables estadísticas, en particular la **PDF** (*Función de Densidad de Probabilidad*).

1. Ruido Impulsivo

El ruido impulsivo esta presente en elementos electrónicos de dispositivos CCD, errores en transmisión o ruido externo que contamina la conversión A/D. Ver figura (2)

Casos particulares:

- Ruido Bipolar (de disparo o sal y pimienta): $P_a \approx P_b$
- Ruido Unipolar: $P_a = 0$ ó $P_b = 0$

2. Ruido Uniforme

Utilizado para simulación de valores aleatorios. Ver figura (3)

3. Ruido Gaussiano

Generado por componentes electrónicos. Ampliamente utilizado por su versatilidad matemática, aún en situaciones donde no se ajusta a los principios físicos del ruido. Ver figura (4)

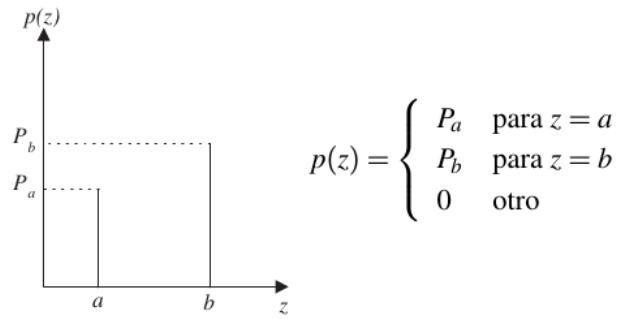


Figura 2: Ruido Impulsivo.

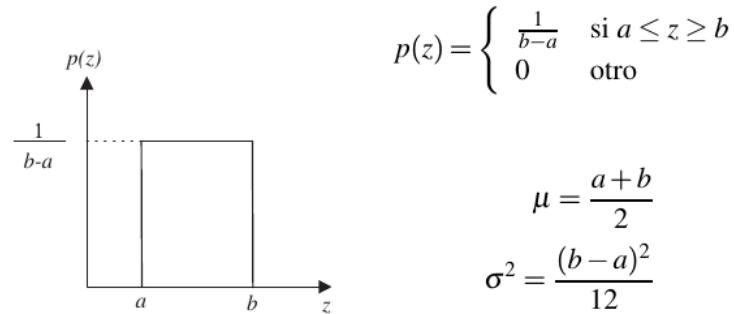


Figura 3: Ruido Uniforme.

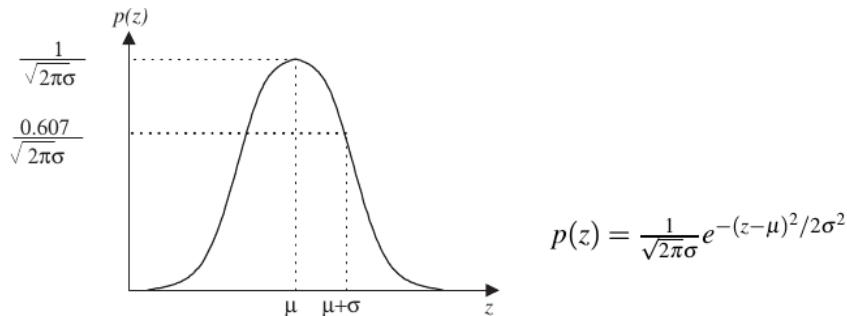


Figura 4: Ruido Gaussiano.

4. Ruido Gamma

Presente en imágenes generadas por láser. Ver figura (5)

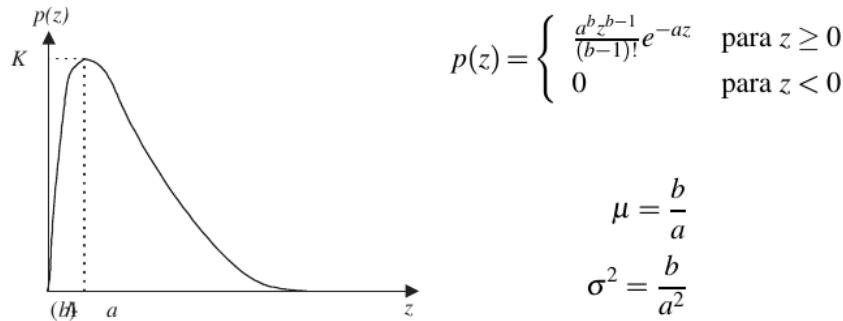


Figura 5: Ruido Gamma.

5. Ruido Exponencial

Presente en imágenes generadas por láser. Caso particular de ruido gamma con $b = 1$. Ver figura (6)

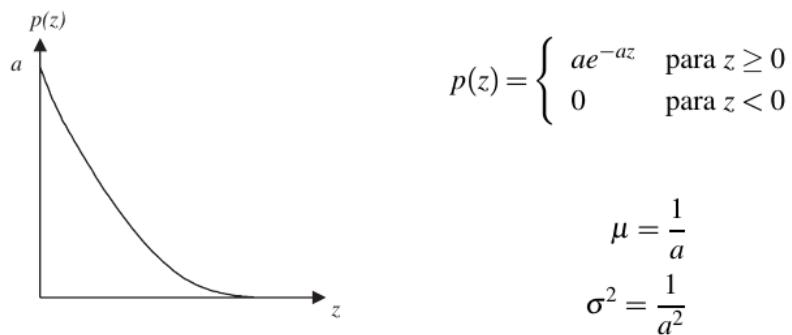


Figura 6: Ruido Exponencial.

En las figuras presentes en (7) se muestran los modelos de ruido antes mencionados.

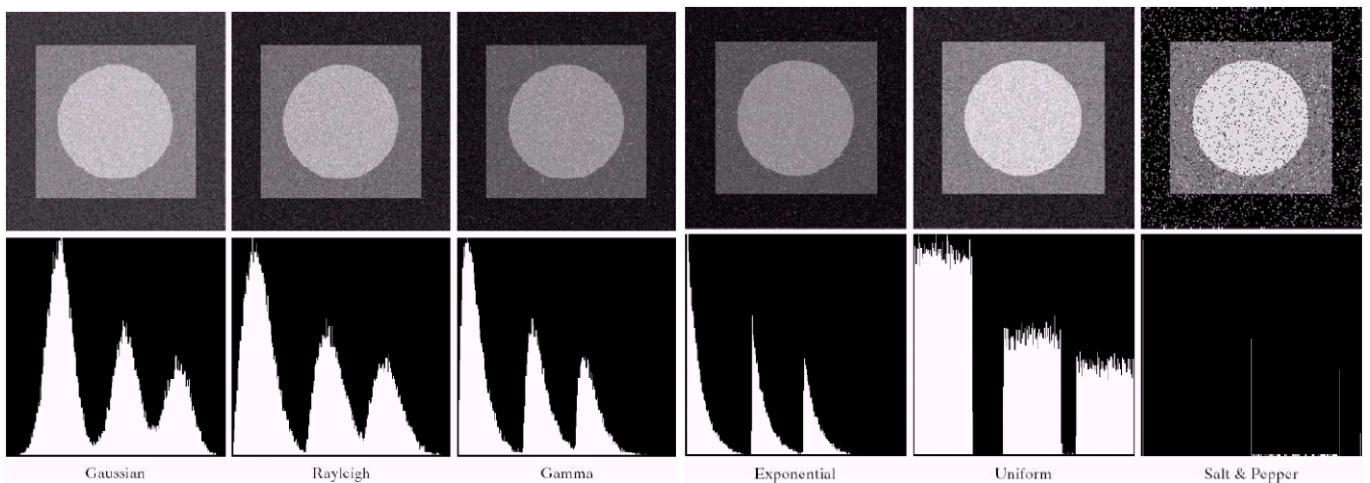


Figura 7: Modelos de Ruidos

La estimación de parámetros estadísticos del ruido se puede abordar según:

- Hoja de datos de sensores: dependencia con el sistema de adquisición.
- Prueba con imágenes: utilización del histograma como estimador de la distribución del ruido. Patrón de áreas de intensidad constante o recortes de áreas constantes en imágenes adquiridas.
- Cálculo de: media y varianza, Parámetros a y b . Patrón de gris medio constante para estimar ruido impulsivo.

3. Restauración Mediante Filtrado Espacial

Cuando $H = 1$ sólo tenemos ruido aditivo:

$$\begin{aligned} g(x, y) &= f(x, y) + \eta(x, y) \\ G(u, v) &= F(u, v) + N(u, v) \end{aligned}$$

Generalmente (salvo en el ruido periódico), la forma de N no es conocida, por lo que la resta $G - N$ no es posible. El método adecuado es el filtrado espacial.

3.1. Filtros de Medias

Promediado sobre una vecindad.

1. **Filtro de Media Aritmética:** Reduce el ruido por desenfoque, al convolucionar la imagen con una máscara de $m \times n$

coeficientes con valor $1/mn$.

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- 2. Filtro de Media Geométrica** Suaviza la imagen con menor pérdida de detalles. Bueno para ruido gaussiano, falla con ruido impulsivo. Ver figura (8).

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

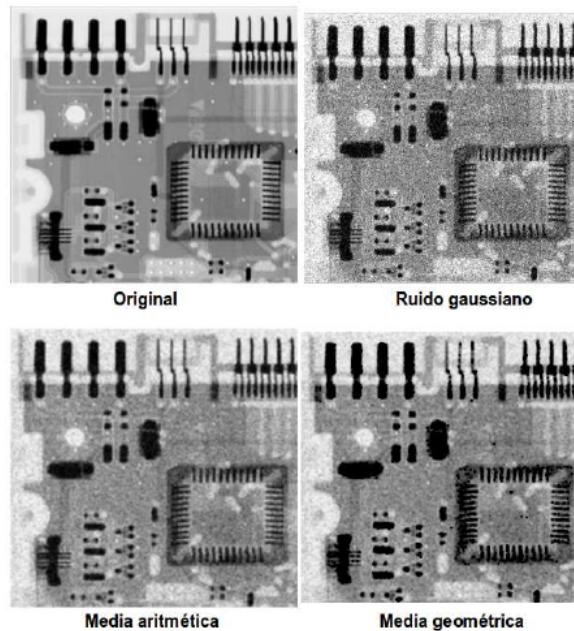


Figura 8: Modelos de Ruido: Ejemplos.

- 3. Filtro de Media Armónica** Bueno para ruido sal, malo para ruido pimienta. Bueno para ruido gaussiano, preservando detalles de la imagen. Ver figuras (9) y (10)

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t)} \frac{1}{g(s, t)}}$$

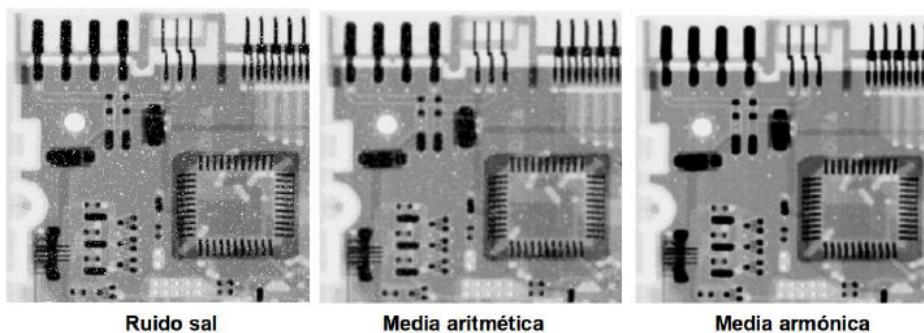


Figura 9: Modelos de Ruido: Ejemplos.

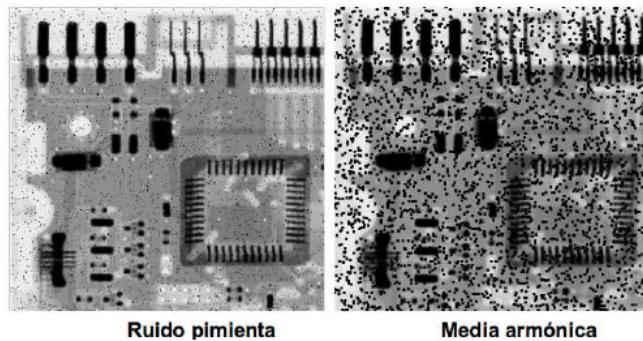


Figura 10: Modelos de Ruido: Ejemplos.

4. **Filtro de Media Contra-Armónica** Q : orden del filtro. Bueno para ruido sal y pimienta, $Q > 0$: elimina pimienta, $Q < 0$: elimina sal. $Q = 0$: media aritmética, $Q = -1$: media armónica. Ver figuras (11), (12) y (13).

Figura 11: Filtro contra-armónico con $Q = -3$ sobre imagen con ruido tipo sal.Figura 12: Filtro contra-armónico con $Q = 3$ sobre imagen con ruido tipo pimienta.

3.2. Filtros de Orden

Ordenamiento ascendente sobre una vecindad y reemplazo de grises.

1. **Filtro de la Mediana** Elección del valor medio de la vecindad. Buena reducción de ruido impulsivo sin el desenfoque de un filtro lineal de la misma talla. Ver figura (14)

$$\hat{f}(x, y) = \text{mediana}_{(s,t)} \{g(s, t)\}$$

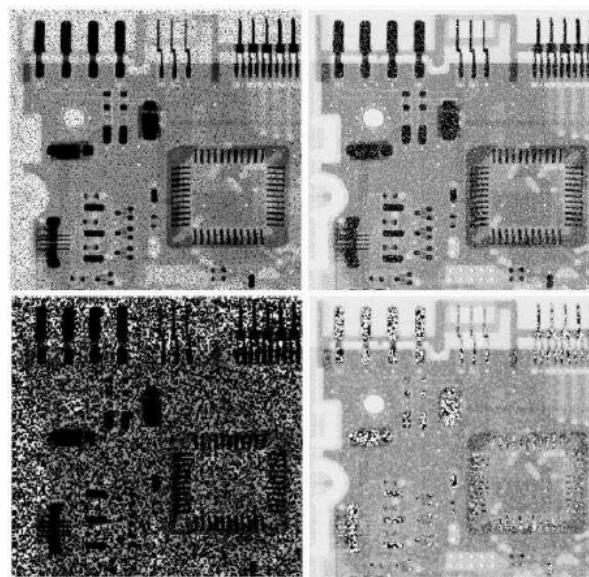


Figura 13: Mala elección de Q en filtro contra-armónico.

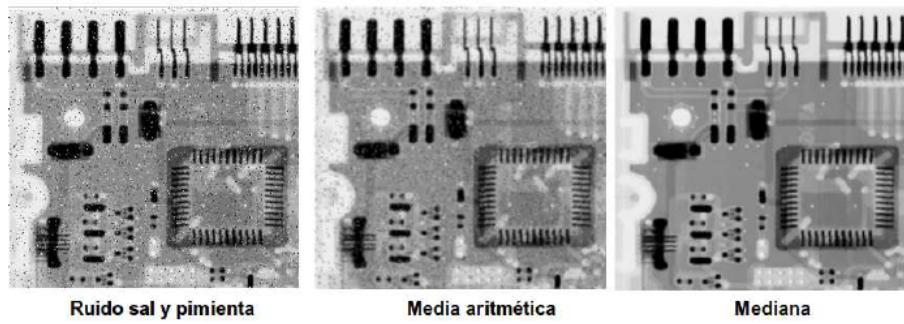


Figura 14: Filtro de Mediana.

2. Filtro de la Moda Elección del valor más frecuente en la vecindad. Aplicación de un umbral para el conteo. Útil en ruido impulsivo, malo para otros tipos. Ver figura (15)

$$\hat{f}(x, y) = moda_{(s,t)} \{g(s, t)\}$$



Figura 15: Ejemplo de Filtro de Moda.

3. Filtro de Máxima Elección del valor máximo en la vecindad. Útil para ruido tipo pimienta. Ver figura (16).

$$\hat{f}(x, y) = max_{(s,t)} \{g(s, t)\}$$

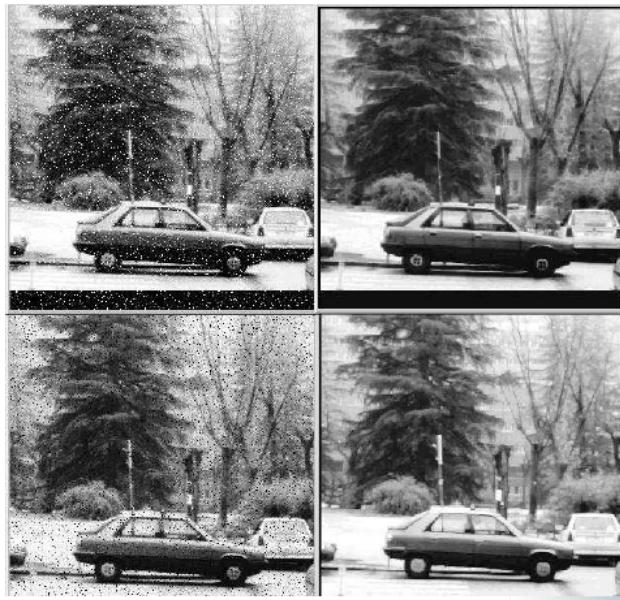


Figura 16: Ejemplo de filtro de mínima (línea superior) y máxima (línea inferior).

4. **Filtro del Punto Medio** Elección del promedio entre el máximo y mínimo de la vecindad. Útil para ruido tipo gaussiano o uniforme.

$$\hat{f}(x, y) = \frac{1}{2} [min_{(s,t)} \{g(s, t)\} + max_{(s,t)} \{g(s, t)\}]$$

5. **Filtro de la media alfa-recortado** Cálculo del promedio entre valores de la vecindad, con exclusión de d puntos: $d/2$ más bajos y $d/2$ más altos. Comportamiento situado entre la media aritmética y la mediana, dependiendo del valor de d . Útil para combinaciones de ruido gaussiano y sal y pimienta.

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t)} g_t(s, t)$$

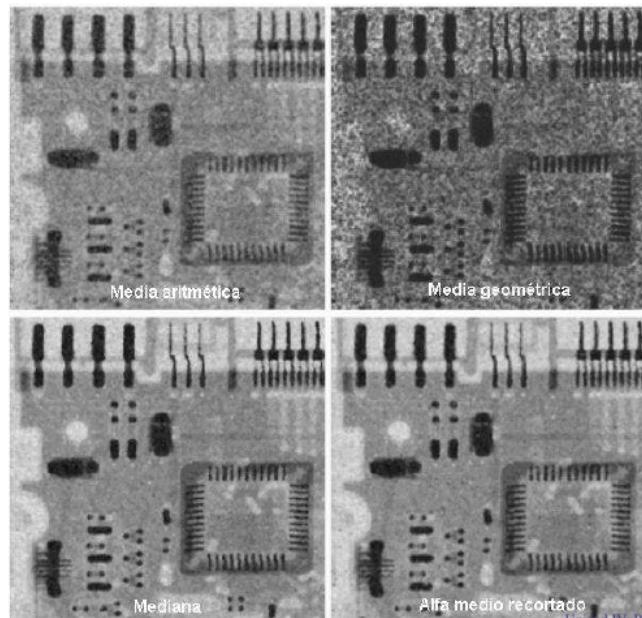


Figura 17: Comparación.

3.3. Filtros Adaptativos

Adaptación del comportamiento acorde a las características de la vecindad.

Hasta ahora: los filtros se aplicaban teniendo en cuenta características estadísticas globales de la imagen. Los filtros adaptativos cambian su comportamiento según la estadística de la imagen dentro de la ventana bajo análisis. El desempeño es superior a los filtros vistos anteriormente. Conllevan un aumento en la complejidad del filtro. Existen dos filtros básicos: Filtro adaptativo de reducción local del ruido y Filtro adaptativo de mediana.

- Filtro de reducción local del ruido:** Las medidas más simples de una variable aleatoria son la media y la varianza. Ante ruido nulo (caso trivial), el filtro debe retornar el valor de la imagen original. Ante variaciones locales (σ_L^2) cercanas a la varianza del ruido (σ_η^2), el filtro debe devolver la media (m_L). Ante variaciones locales mucho mayores que el ruido, el filtro debe devolver un valor cercano a la imagen original (varianza local alta asociada a bordes y detalles).

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2} [g(x, y) - m_L]$$

Se asume que $\sigma_\eta^2 \leq \sigma_L^2$. Debe conocerse la varianza del ruido. Algunas veces no se conoce σ_η^2 , por lo que hay que estimarla: Si la estimación es baja el algoritmo devuelve un resultado satisfactorio porque los cambios son pequeños. Si la estimación es alta se fija la relación a 1 y el algoritmo devuelve la media. Introducción de no linealidad en el algoritmo por la comprobación de la relación entre varianzas, que previene resultados sin sentido.

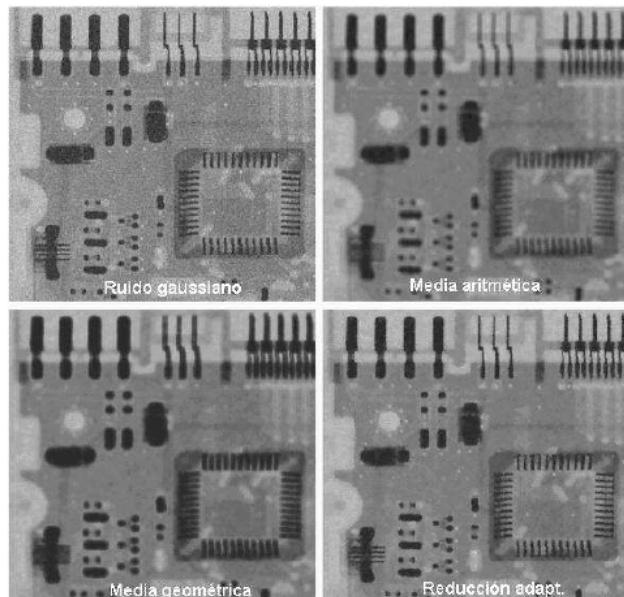


Figura 18: Comparación.

4. Ruido Periódico y Filtrado Frecuencial

4.1. Ruido Periódico

El **Ruido Periódico** está presente como interferencia eléctrica o electromecánica durante la adquisición de una imagen. tiene una gran dependencia con las características espaciales en la imagen y sus componentes son claramente visibles en el dominio frecuencial.

Para este tipo de ruido se realiza la estimación de parámetros estadísticos, por ejemplo realizando una Inspección del Espectro de Fourier, o una Inferencia Directa sobre la imagen (en casos muy simples), o aplicando Métodos Automáticos. Por otra parte, la reducción del ruido se hace mediante filtrado en el dominio frecuencial aplicando Filtros Pasa-Banda, Rechaza-Banda o Notch.

4.2. Restauración Mediante Filtrado Frecuencial

1. **Filtro rechaza-banda:** elimina o atenúa una banda de frecuencias alrededor del origen del espectro.

- **Ideal**

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) < D_0 - \frac{W}{2} \\ 0 & \text{si } D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1 & \text{si } D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

con $D(u, v) = \sqrt{u^2 + v^2}$, W : ancho del filtro y D_0 : centro radial.

- **Butterworth**

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)W}{D^2(u, v) - D_0^2} \right]^{2n}}$$

- **Gaussiano**

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[\frac{D^2(u, v) - D_0^2}{D(u, v)W} \right]^2}$$

2. **Filtro pasa-banda:** Acción opuesta al rechaza-banda. Útil para aislar las componentes de ruido y estudiar las propiedades de la imagen que está interfiriendo con la adquirida. Se obtiene a partir de la especificación de la función de transferencia del filtro rechaza-banda:

$$H_{PB}(u, v) = 1 - H_{RB}(u, v)$$

3. **Filtro Notch:** Rechaza frecuencias alrededor de una frecuencia central. Los picos aparecen de a pares conjugados (salvo en el origen). Diferentes configuraciones: número de pares, tamaño del área notch.

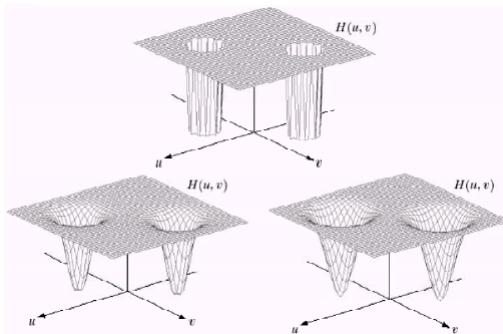


Figura 19: Filtros Notch.

- **Ideal:**

$$H(u, v) = \begin{cases} 0 & \text{si } D_1(u, v) \leq D_0 \text{ o } D_2(u, v) \leq D_0 \\ 1 & \text{en otro caso} \end{cases}$$

con $D_1(u, v) = D_2(u, v) = [(u - M/2 - u_0)^2 + (v - M/2 - v_0)^2]^{1/2}$

- **Butterworth:**

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u, v)D_2(u, v)} \right]^n}$$

- **Gaussiano:**

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[\frac{D_1(u, v)D_2(u, v)}{D_0^2} \right]}$$

- **Filtros Notch Pasantes:** opuestos a los anteriores, con

$$H_{PN}(u, v) = 1 - H_{RN}(u, v)$$

5. Estimación de la Degradación

Hasta aquí, se estudió el efecto del ruido con degradación nula. Ahora estudiaremos el efecto de la función de degradación con ruido nulo, es decir:

$$\begin{aligned} g(x, y) &= h(x, y) * f(x, y) \\ G(u, v) &= H(u, v)F(u, v) \end{aligned}$$

Comenzaremos estudiando degradaciones que pueden ser modeladas mediante procesos lineales e invariantes a la posición. Los filtros utilizados reciben el nombre de *filtros de deconvolución*. El proceso de restauración se denomina *filtrado inverso*. El problema se aborda realizando estimación de la función de degradación, habiendo tres maneras y planteo de algoritmos de filtrado.

5.1. Estimación de H

1. Estimación por observación de la imagen:

- Utilizado cuando se dispone solamente de imágenes adquiridas, no del sistema de adquisición.
- En áreas de alto contenido de señal podemos copiar grises y generar una subimagen limpia $\hat{f}_s(x, y)$ basada en las estructuras de la imagen.
- Inferencia de características de H mediante:

$$H_s(u, v) = \frac{G_s(u, v)}{\widehat{F}_s(u, v)}$$

- Generación de $H(u, v)$ mediante escalado (la invariancia al desplazamiento permite hacerlo).

2. Estimación por Aproximación:

- Utilizado cuando se dispone del sistema de adquisición.
- Obtención de la respuesta al impulso del sistema adquiriendo un patrón puntual de luz.
- Estimación de la función de degradación mediante:

$$H(u, v) = \frac{G(u, v)}{A}$$

3. Estimación por modelización:

- A menudo se emplea un modelo físico para obtener la respuesta al impulso del sistema degradante. Ej: modelo de turbulencia atmosférica (desenfoque aprox. gaussiano)

$$e^{-k(u^2+v^2)^{5/6}}$$

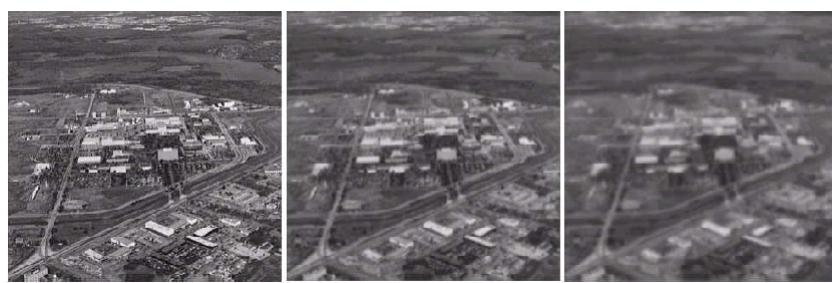


Figura 20: Estimación de H .

- En caso de conocer el motivo de la degradación, se puede plantear y ajustar un modelo matemático. Ej: desenfoque debido a movimiento uniforme en la imagen

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin [\pi(ua + vb)] e^{-j\pi(ua + vb)}$$

Figura 21: Estimación de H .

5.2. Filtrado Inverso

En la ausencia de ruido, y conocida (o estimada) la función H , la manera más simple de restaurar la imagen es:

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} = R(u, v)G(u, v)$$

donde el filtro inverso es $1/H(u, v)$.

En la presencia de ruido, la ecuación del sistema queda:

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

Analizando la ecuación, aunque se conozca H no se puede recuperar la imagen original. En valores pequeños de H , el ruido se amplifica desmesuradamente. En la práctica pueden ser difíciles de construir ya que son inestables.

1. **Primer Solución:** utilizar filtros pseudo-inversos:

$$R(u, v) = \begin{cases} \frac{1}{H(u, v)} & |H(u, v)| > \varepsilon \\ 0 & \text{en el resto} \end{cases}$$

2. **Segunda Solución:** suavizar la relación G/H con filtros pasa-bajos:

Figura 22: Estimación de H .

6. Filtrado de Mínimos Cuadrados

Aproximación que maneja la estadística del ruido y la función de degradación conjuntamente. El método considera a la imagen y al ruido como procesos aleatorios, y aplica estadística (correlación) entre la imagen y el ruido.

Filtro paramétrico de Wiener: función que minimiza la relación señal/ruido. La restauración en el dominio frecuencial queda dada por:

$$\widehat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \gamma \frac{S_\eta(u, v)}{S_f(u, v)}} \right] G(u, v)$$

γ : parámetro del filtro. $S_\eta(u, v) = |N(u, v)|^2$: espectro de potencia del ruido. $S_f(u, v) = |F(u, v)|^2$: espectro de potencia de la imagen original.

Casos particulares:

- $\gamma = 0$: filtro inverso.
- $\gamma = 1$: filtro de Wiener.
- $\gamma \neq 0, 1$: filtro paramétrico de Wiener.

La elección de $\gamma = 1$ conduce a una función que minimiza el error cuadrático medio (óptimo estadísticamente):

$$e^2 = E \{ (f - \widehat{f})^2 \}$$

La implementación requiere conocer los espectros de potencia de la imagen y del ruido, o bien la relación señal/ruido $\rho = S_\eta(u, v)/S_f(u, v)$:

$$\widehat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{\gamma}{\rho}} \right] G(u, v)$$

Notar que cuando $\eta \rightarrow 0$, el filtro paramétrico tiende al filtro inverso.

Cuando no se conocen (o no se pueden estimar) los espectros de potencia se emplea la siguiente aproximación:

$$\widehat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

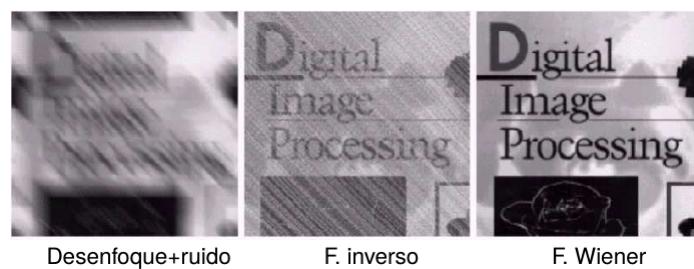
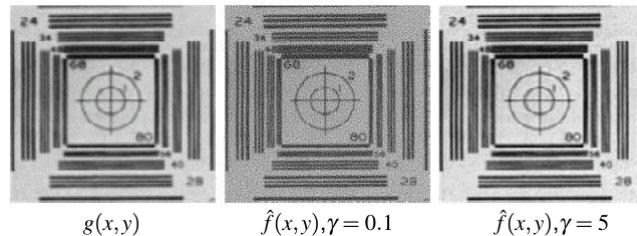


Figura 23: Comparación de Filtros.

Efecto del parámetro γ en el filtro paramétrico:

Valores pequeños de γ : mejor remoción de la degradación H (ej: desenfoque), y pobre filtrado del ruido (ej: ruido gaussiano). Valores altos de γ : pobre restauración de la degradación H y mejor filtrado del ruido.

Figura 24: Efectos de γ .

7. Filtro de la Media Geométrica

Generalización del filtro de Wiener:

$$\widehat{F}(u, v) = \left[\frac{H^*(u, v)}{|H(u, v)|^2} \right]^\alpha \left[\frac{H^*(u, v)}{|H(u, v)|^2 + \beta \frac{S_n(u, v)}{S_f(u, v)}} \right]^{1-\alpha} G(u, v)$$

con $\alpha, \beta \in \mathbb{R} \geq 0$.

Familia de filtros según los parámetros:

- $\alpha = 1$: filtro inverso.
- $\alpha = 0$: filtro paramétrico de Wiener. $\beta = 1$: filtro de Wiener.
- $\alpha = 1/2$: media geométrica entre el *f.* inverso y el *f.* de Wiener.
- $\alpha = 1/2$ y $\beta = 1$: filtro de ecualización del espectro de potencia.
- Combinaciones.

2. UNIDAD 5 - Segmentación

La segmentación se trata de dividir una imagen en sus regiones constituyentes u objetos de interés. Es un proceso el cual condiciona el resultado en etapas posteriores de análisis.

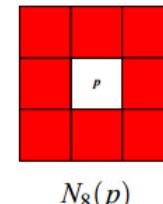
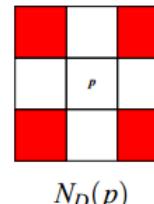
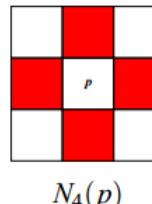
Un concepto ligado a segmentación es el de **vecindad de un píxel**. Dado un píxel $p = (x, y)$ se definen 3 vecindades relativas a p :

- 4-Vecinos: $N_4(p) = (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$
- Vecinos diagonales: $N_D(p) = (x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$
- 8-Vecinos: $N_8(p) = N_4(p) \cup N_D(p)$

Se debe tener en cuenta que si p se encuentra en algún borde de la imagen algunos de sus vecinos pueden quedar fuera de esta.

i-1,j-1	i-1,j	i-1,j+1
i,j-1	p	i,j+1
i+1,j-1	i+1,j	i+1,j+1

(a) Estructura de Vecindad



(b) Tipos de Vecindades

1. Detección de Discontinuidades

Se define discontinuidad a un cambio abrupto en la intensidad de gis. Existen 3 tipos básicos de discontinuidades a detectar: **Puntos, Líneas y Bordes**.

Metodología de detección: Primero se realiza un filtrado espacial con una máscara apropiada. Aplicando una máscara $W_{3 \times 3}$:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_n z_n = \sum_{i=1}^n w_i z_i$$

donde w_i son los coeficientes de la máscara y z_i son los valores de intensidad de los píxeles de la imagen. Luego se realiza una umbralización para binarizar el resultado obtenido.

1.1. Detección de Puntos

El objetivo principal de esta detección es identificar puntos aislados en zonas homogéneas. Para lograr esta detección 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, como mencionado anteriormente, se umbraliza con un determinado valor de umbral T .

1.2. Detección de Líneas

El objetivo principal de esta detección es identificar líneas de grosor de un píxel con distintas orientaciones. Para lograr esta detección se aplica una máscara direccional de suma cero (ver figura 25).

Luego se umbraliza con un determinado valor de umbral T . Se debe tener en cuenta que se puede aplicar una única máscara para identificar un tipo de linea, o muchas máscaras y luego sumar el resultado.

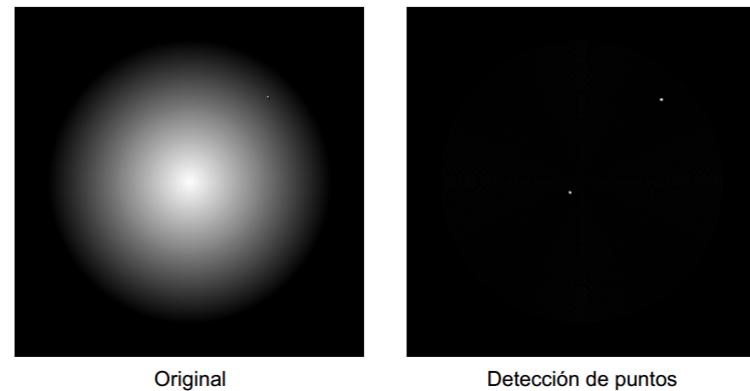
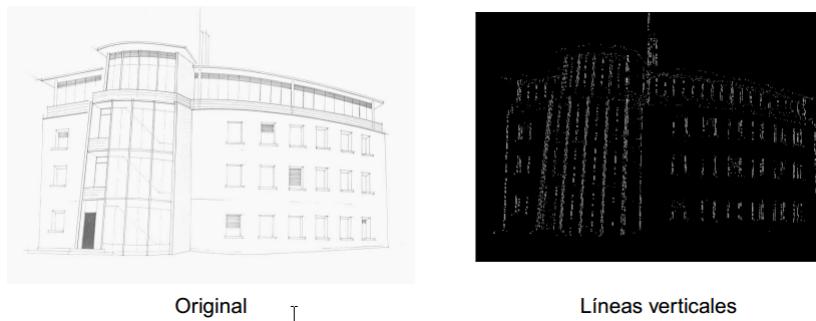


Figura 25: Detección de líneas. Máscaras direccionales de suma cero.



1.3. Detección de Bordes

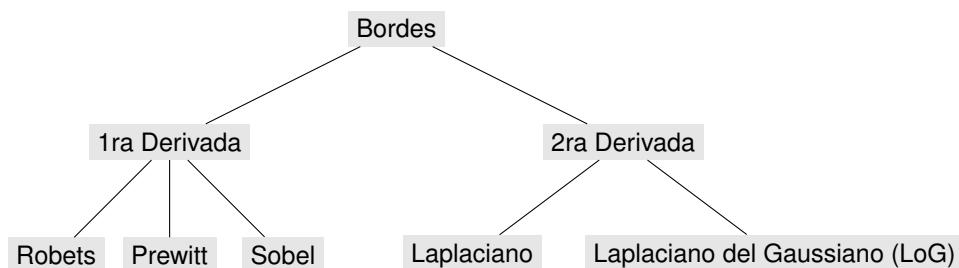
Existen dos tipos de bordes a detectar, por un lado el **borde ideal** identificado como un escalón y por otro lado tenemos el **borde real** identificado como una rampa. Esta identificación se realiza en relación a las intensidades de grises en los píxeles.



Figura 26: a) Borde Ideal - b) Borde Real

En los bordes reales el borde esta formado por todos los píxeles que contiene la rampa. Además el grosor de dicho borde depende de la pendiente de la rampa.

Métodos de detección de bordes:



- Métodos de la primera derivada:** En este método se tiene en cuenta la magnitud de la primera derivada para detectar la presencia de un borde. Dicha magnitud se utiliza en conjunto con una umbralización donde aquellos puntos que tengan magnitud mayor al valor determinado por el umbral se considera borde. El operador a tener en cuenta en estos

métodos es el *Gradiente*, donde se define el gradiente de una imagen $f(x, y)$ en el punto (x, y) como el vector siguiente:

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

La magnitud y dirección del vector gradiante \mathbf{f} se definen como:

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

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \text{ (Perpendicular a la dirección de un borde en un punto)}$$

Como muestra el árbol superior existen tres tipos de métodos dentro de los de primera derivada: **Robots**, **Prewitt** y **Sobel**. Cada uno de estos se define a partir de una vecindad del tipo:

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

- **Robots:** En este método las componentes del gradientes se calculan como:

$$G_x = z_9 - z_5$$

$$G_y = z_8 - z_6$$

Las máscaras que lo implementan son:

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

- **Prewitt:** En este método las componentes del gradientes se calculan como:

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

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

Las máscaras que lo implementan son:

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

Este método tiene una respuesta nula en zona de grises constantes.

- **Sobel:** En este método las componentes del gradientes se calculan como:

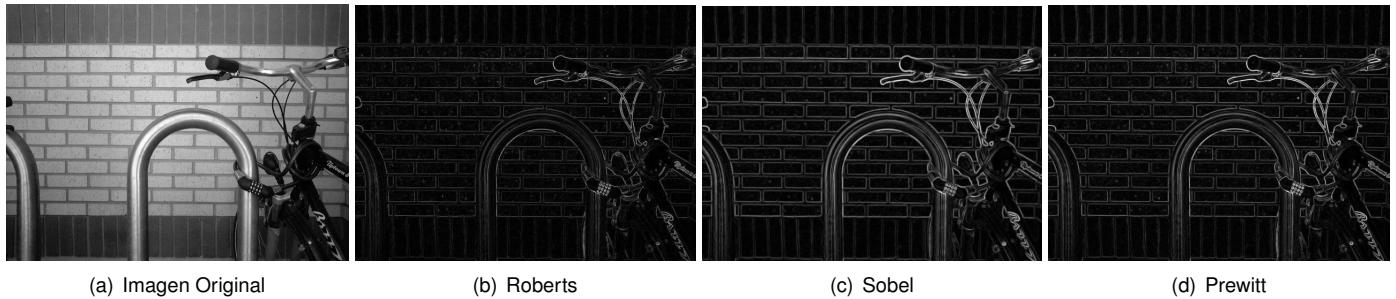
$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

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

Las máscaras que lo implementan son:

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

Este método tiene una respuesta nula en zona de grises constantes. Enfatiza los píxeles más cercanos al centro, consiguiendo una mejor respuesta en presencia de ruido tipo gaussiano.



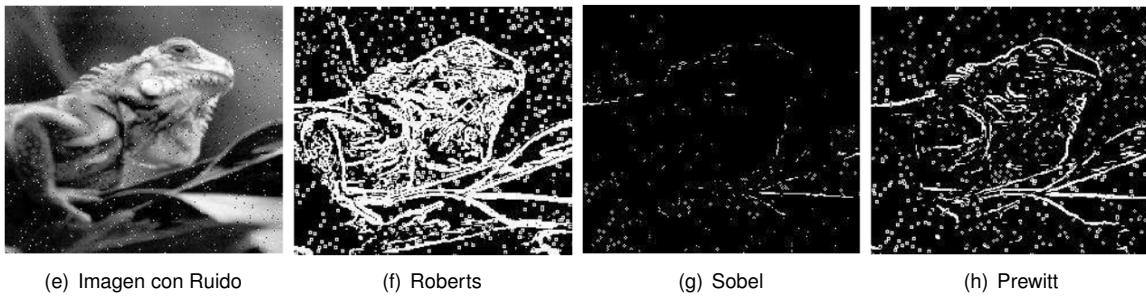
(a) Imagen Original

(b) Roberts

(c) Sobel

(d) Prewitt

- **COMPORTAMIENTO ANTE EL RUIDO:** Estos métodos presentan una alta sensibilidad a la existencia de ruido. Poseen una relación de compromiso entre sensibilidad al ruido y exactitud en la detección de bordes. Los parámetros de control para estas relaciones son el tamaño del filtro y el nivel del umbral, donde a mayor tamaño de filtro o del nivel del umbral se consigue una mayor remoción de ruido, pero se pierden bordes. Lo más utilizado y aconsejado es la eliminación del ruido en etapas previas de procesamiento.



(e) Imagen con Ruido

(f) Roberts

(g) Sobel

(h) Prewitt

2. Métodos de la segunda derivada:

- **Laplaciano:** Se define el laplaciano de una función 2D como: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

Considerando una aproximación con N_4 obtenemos una máscara del tipo:

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

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

Considerando una aproximación con N_8 obtenemos una máscara del tipo:

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

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

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.

- **Laplaciano del Gaussiano (LoG):** En este método se utiliza una suavización mediante un filtro gaussiano seguida de la aplicación del laplaciano:

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

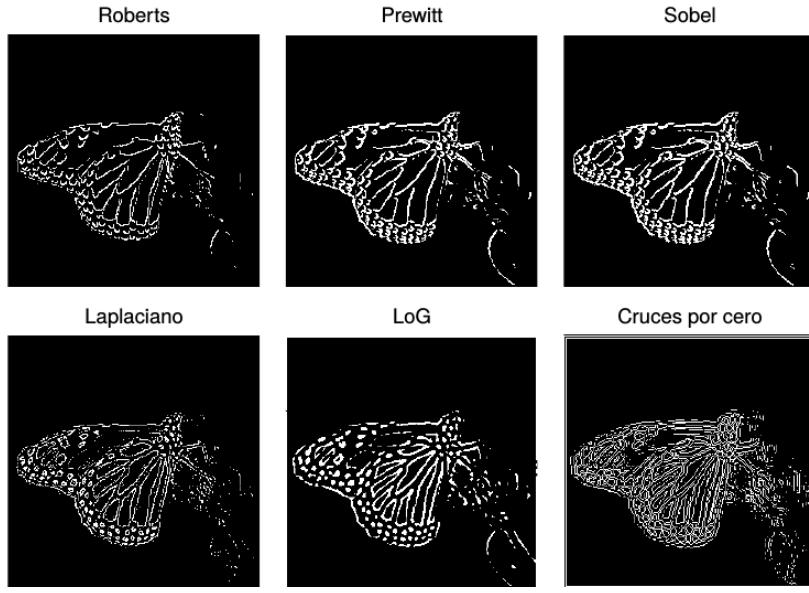
Entonces, al aplicar el laplaciano se tiene que:

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

Un ejemplo de máscara en este método es:

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

3. Comparación de todos los métodos:



2. Conexión de bordes: Proceso local

Una necesidad que se plantea con los puntos y segmentos encontrados con los métodos anteriores es unirlos y así poder detectar los bordes de un objeto. Se puede ver que cada punto o segmento detectado pertenece a un borde de algún objeto. De todas formas se debe tener en cuenta que dado por ejemplo el ruido o segmentos discontinuos se da la posibilidad de la detección de puntos y bordes falsos o espúreos.

Metodología: Recorrer la imagen de magnitud del gradiente, analizando pequeños entornos (ej. 3×3). 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.

$$\left| \nabla f(x, y) - \nabla f(x_0, y_0) \right| \leq T_1 \quad \left| \alpha(x, y) - \alpha(x_0, y_0) \right| < T_2$$

con T_1 y T_2 dos umbrales no negativos.

Búsqueda de líneas rectas por Transformada de Hough: El caso más simple para la transformada de Hough es la transformación lineal para detectar líneas rectas. En el espacio de la imagen, la recta se puede representar con la ecuación $y = m \cdot x + n$ y se puede graficar para cada par (x, y) de la imagen. En la transformada de Hough, la idea principal es considerar las características de una recta en término de sus parámetros (m, n) , y no como puntos de la imagen $(x_1, y_1), \dots, (x_n, y_n)$. Basándose en lo anterior, la recta $y = m \cdot x + n$ se puede representar como un punto (m, n) en el espacio de parámetros. Sin embargo, cuando se tienen rectas verticales, los parámetros de la recta (m, n) se indefinen. Por esta razón es mejor usar los parámetros que describen una recta en coordenadas polares, denotados (ρ, θ) .

El parámetro ρ representa la distancia entre el origen de coordenadas y el punto (x, y) , mientras que θ es el ángulo del vector director de la recta perpendicular a la recta original y que pasa por el origen de coordenadas.

Usando esta parametrización, la ecuación de una recta se puede escribir de la siguiente forma: $y = \left(-\frac{\cos \theta}{\sin \theta} \right) \cdot x + \left(\frac{\rho}{\sin \theta} \right)$ que se puede reescribir como $\rho = x \cdot \cos \theta + y \cdot \sin \theta$.

Entonces, es posible asociar a cada recta un par (ρ, θ) que es único si $\theta \in [0, \pi)$ y $\rho \in \mathbf{R}$ o $\theta \in [0, 2 * \pi)$ y $\rho \geq 0$. El espacio (ρ, θ) se denomina espacio de Hough para el conjunto de rectas en dos dimensiones.

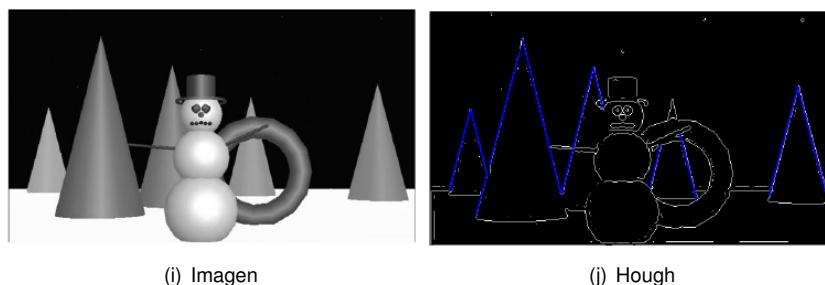
Para un punto arbitrario en la imagen con coordenadas (x_0, y_0) , las rectas que pasan por ese punto son los pares (ρ, θ) con $r = x \cdot \cos \theta + y \cdot \sin \theta$ donde ρ (la distancia entre la línea y el origen) está determinado por θ . Esto corresponde a una curva sinusoidal en el espacio (ρ, θ) , que es única para ese punto. Si las curvas correspondientes a dos puntos se intersecan, el punto de intersección en el espacio de Hough corresponde a una línea en el espacio de la imagen que pasa por estos dos puntos. Generalizando, un conjunto de puntos que forman una recta, producirán senosoides que se intersecan en los parámetros de esa línea. Por tanto, el problema de detectar puntos colineales se puede convertir en un problema de buscar curvas concurrentes.

Pseudocódigo del método:

```

1: cargar imagen
2: detectar los bordes en la imagen
3: por cada punto en la imagen:
4:     si el punto  $(x, y)$  esta en un borde:
5:         por todos los posibles ángulos  $\theta$ :
6:             calcular  $\rho$  para el punto  $(x, y)$  con un ángulo  $\theta$ 
7:             incrementar la posición  $(\rho, \theta)$  en el acumulador
8: buscar las posiciones con los mayores valores en el acumulador
9: devolver las rectas cuyos valores son los mayores en el acumulador.

```



3. Segmentación basada en regiones

Se define **región** al conjunto de píxeles vecinos que cumplen una condición de semejanza de grises. Entonces sea R la región correspondiente a una imagen completa 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:

- $\bigcup_{i=1}^n R_i = R$
- R_i es una región conectada \forall_i .
- $R_i \cap R_j = \emptyset, \forall i \neq j$.
- R_i cumple una cierta propiedad $P(R_i), \forall_i$.
- Dos regiones R_i y R_j verifican P , pero no la unión de ambas.

Algoritmo de Segmentación basada en Regiones:

- Agrupar píxeles o subregiones en regiones más grandes, basándose en propiedades que tengan en común como una 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.).



4. Criterios de homogeneidad

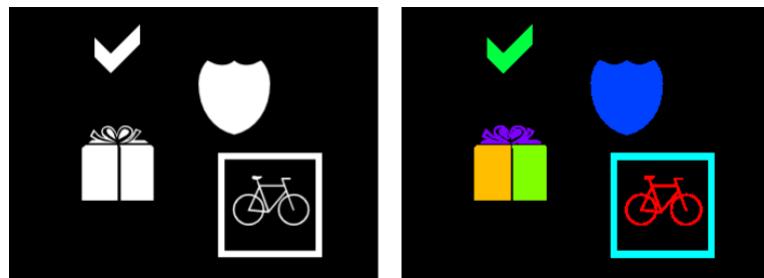
Existen diversos criterios de homogeneidad dentro de las imágenes. Estos dependen de la aplicación y de los conocimientos previos que se tengan. Algunos ejemplos de estos criterios son:

- 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.

5. Etiquetado de componentes conectadas

Se define *Componentes Conectadas* al conjuntos de píxeles conectados por medio de una vecindad o algún tipo de semejanza. Entonces etiquetar estos componentes consiste en asignar un mismo valor de gris o color, los cuales actúan como etiqueta.

En la imagen siguiente se puede observar que cada componente o elemento definido posee un color distinto al resto.



El algoritmo para lograr estos etiquetados consiste en 3 etapas:

1. 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.

2. 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.

Datos a tener en cuenta: 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.

3. UNIDAD 6 - Morfología

1. Usos de la Morfología Matemática

- realce de imágenes
- segmentación de imágenes
- restauración de imágenes
- detección de ejes
- análisis de texturas
- análisis de partículas
- generación de características
- obtención de esqueletos
- análisis de formas
- compresión de imágenes
- análisis de componentes
- espesamiento de curvas
- adelgazamiento general
- detección de características
- reducción de ruido
- filtrado espacio-tiempo

2. Contenido

1. Conceptos preliminares

- Operaciones matemáticas, lógicas y relacionales con imágenes
- Definiciones, propiedades y operaciones con conjuntos
- Imágenes como conjuntos

2. Morfología matemática binaria

- Elemento estructurante
- Operaciones básicas: dilatación, erosión, apertura, cierre y Hit-or-Miss

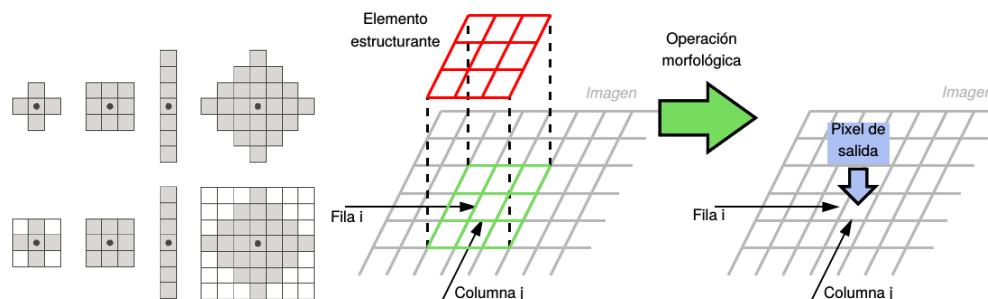
3. Algoritmos y aplicaciones

- Extracción de contornos
- Relleno de agujeros
- Extracción de componentes conectadas
- Envoltura convexa
- Adelgazamiento
- ...

3. Morfología matemática en imágenes

Elemento estructurante (EE):

- Son pequeños conjuntos o sub-imágenes
- Se utilizan para probar propiedades de la imagen que se estudia
- Debe tener especificado un origen
- Se deben definir las condiciones de borde

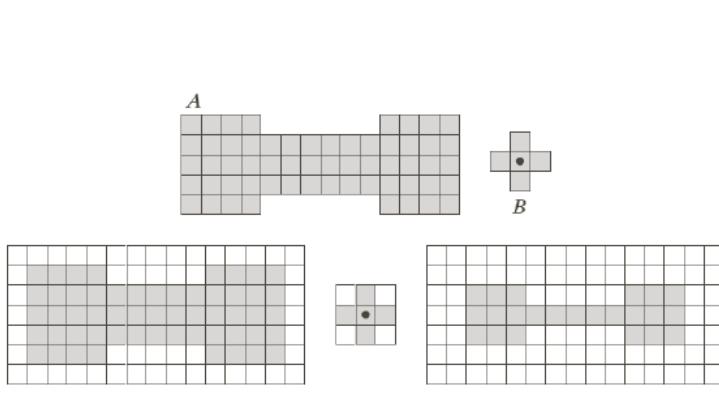


4. Erosión Binaria

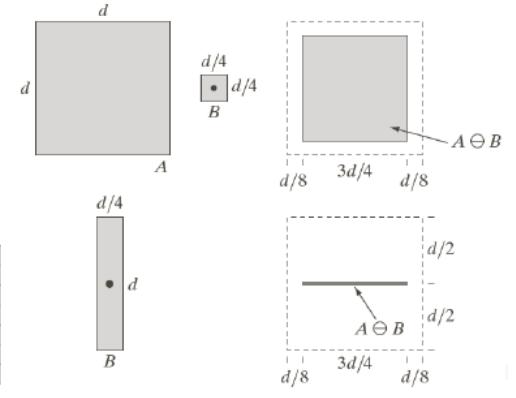
Se comprueba si: ¿Está el EE completamente contenido en el conjunto (imagen)? Considerando A y B como conjuntos de \mathbb{Z}^2 , la erosión se define como:

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad \text{ó}$$

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$



(m) Ejemplo de Erosión Binaria.



(n) Ejemplo de Erosión Binaria.

La erosión binaria *Encoge* y/o *Adelgaza* objetos en una imagen binaria. Puede considerarse una operación de filtrado morfológico.

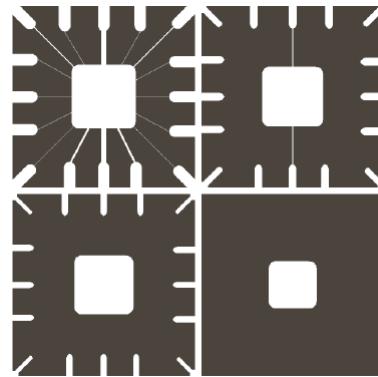


Figura 27: Ejemplo de Erosión Binaria - Encogido.

5. Dilatación binaria

Se comprueba si: ¿El EE reflejado en su origen y la imagen coinciden en, al menos, un elemento?. Considerando A y B como conjuntos de \mathbb{Z}^2 , la dilatación se define como:

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$$

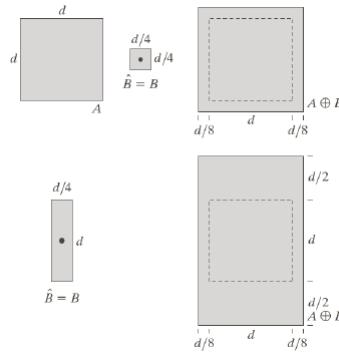


Figura 28: Ejemplo de Dilatación Binaria.

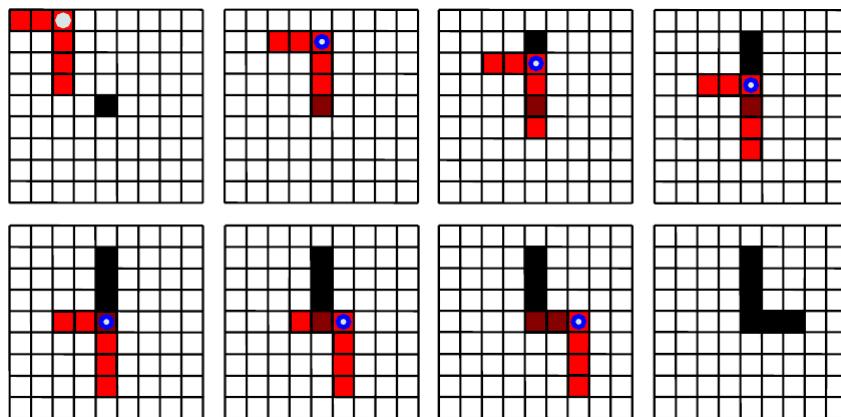


Figura 29: Ejemplo de Dilatación Binaria - EE reflejado.

La dilatación binaria hace *Crecer* y/o *Ensancha* objetos en una imagen binaria. La manera específica y el grado de ensanchamiento está controlado por el EE.

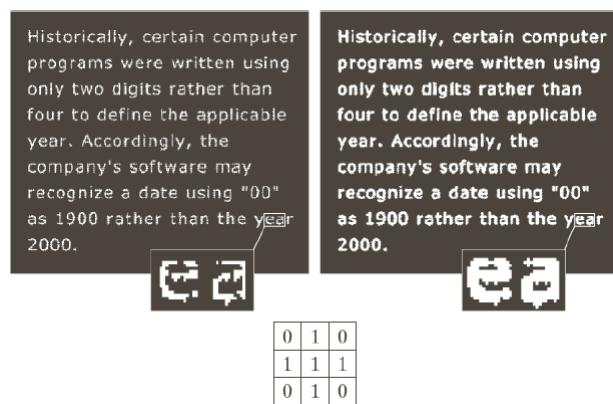
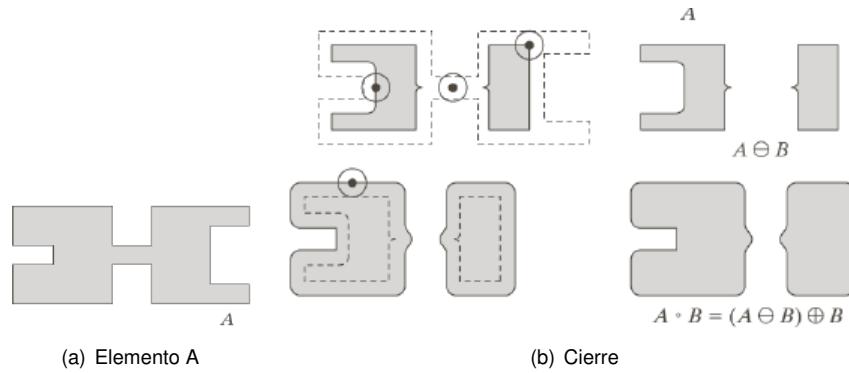


Figura 30: Ejemplo de Dilatación Binaria - Ensanchamiento.

6. Cierre

Suaviza el contorno de un objeto, rompe los istmos (cuello o franja estrecha) y elimina salientes delgadas. La apertura de un conjunto A y B se define como:

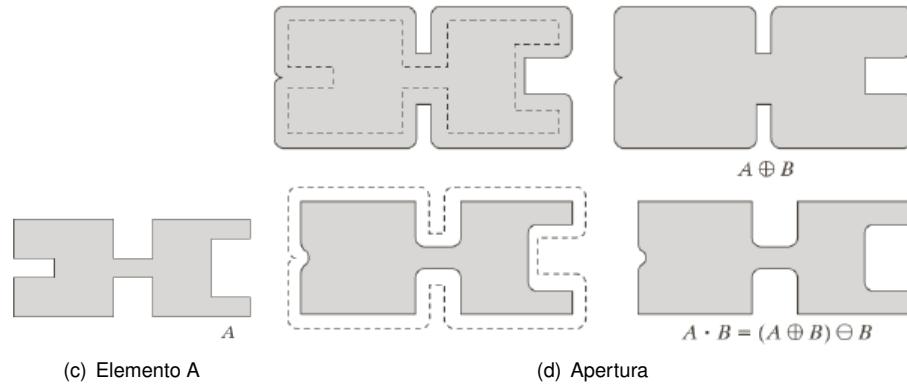
$$A \circ B = (A \ominus B) \oplus B$$



7. Apertura

Suaviza el contorno de un objeto, elimina agujeros pequeños, fusiona discontinuidades estrechas y gollos largos y finos, y rellena lagunas en el contorno. El cierre de un conjunto A y B se define como:

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



8. Propiedades

$$(A \circ B)^c = (A^c \bullet \widehat{B}) \quad \text{y} \quad (A \bullet B)^c = (A^c \circ \widehat{B})$$

- **Apertura**

- Si $C \subseteq D$, entonces $(C \circ B) \subseteq (D \circ B)$
- $A \circ B \subseteq A$
- $A \circ B = (A \circ B) \circ B$

- **Cierre**

- Si $C \subseteq D$, entonces $(C \bullet B) \subseteq (D \bullet B)$
- $A \subseteq A \bullet B$
- $A \bullet B = (A \bullet B) \bullet B$

Se desprende de los ítems c) de ambas operaciones que la aplicación de múltiples aperturas o cierres de una imagen no tienen efecto una vez que el operador fue aplicado una vez.



Figura 31: Ejemplo Apertura y Cierre.

9. Transformación de localización (Hit-or-Miss)

$$\begin{aligned} A \circledast B &= (A \ominus D) \cap [A^c \ominus (W - D)] \\ A \circledast B &= (A \ominus D) - (A \oplus \widehat{E}) \quad \text{con} \quad E = (W - D) \end{aligned}$$

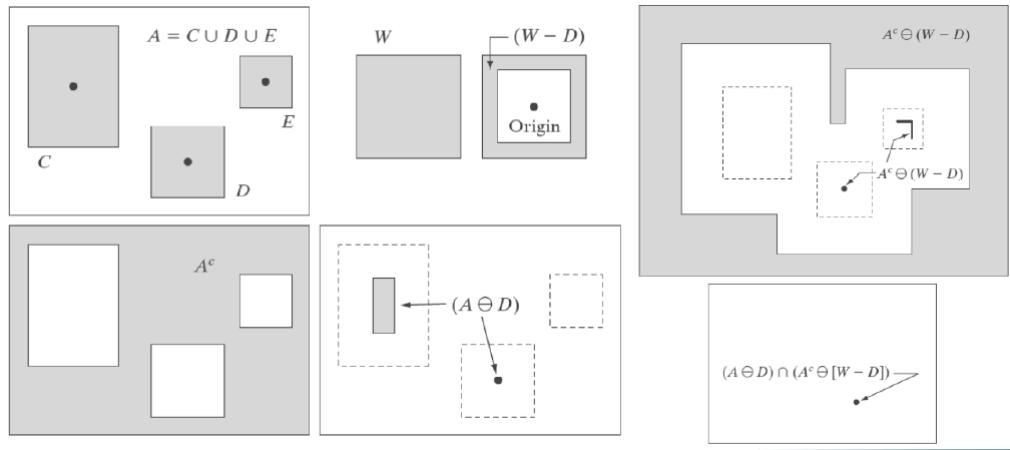


Figura 32: Hit-or-Miss.

10. Algoritmos

10.1. Extracción de contornos (gradiente morfológico)

$$\beta_E(A) = A - (A \ominus B)$$

$$\beta_D(A) = (A \oplus B) - A$$

$$\beta_{DE}(A) = (A \oplus B) - (A \ominus B)$$

Figura 33: EE de 3×3 .Figura 34: EE de 5×5 .

10.2. Relleno de agujeros semi-automático (dilatación condicionada)

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

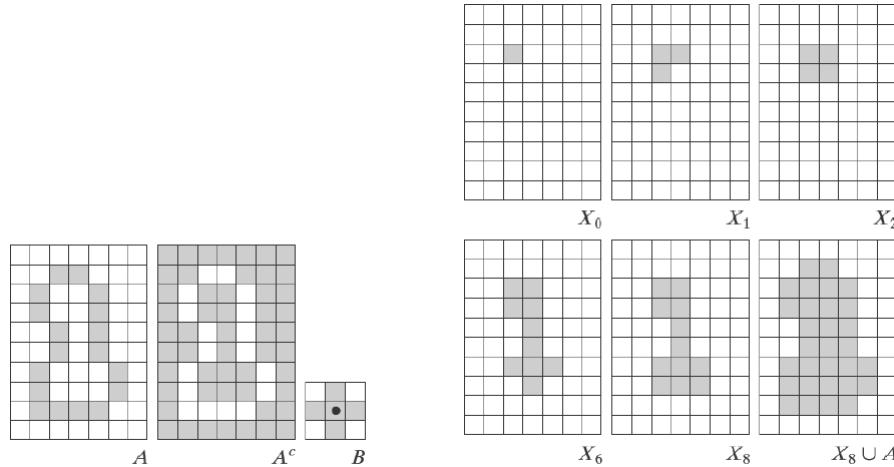


Figura 35: Relleno de agujeros semi-automático.

10.3. Extracción de componentes conectadas

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots$$

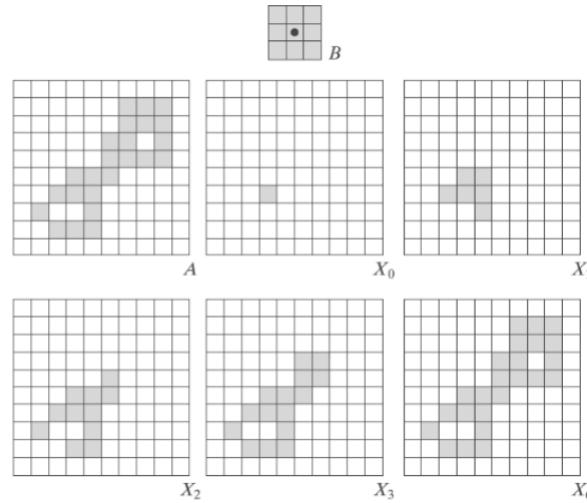


Figura 36: Extracción de componentes conectadas.

10.4. Envoltura convexa (Convex Hull)

El conjunto convexo $C(A)$ que contiene a A se obtiene mediante

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \quad y \quad k = 1, 2, 3, \dots$$

- $X_0^i = A$
- Converge cuando $X_k^i = X_{k-1}^i$
- $C(A) = \bigcup_{i=1}^4 X_k^i$

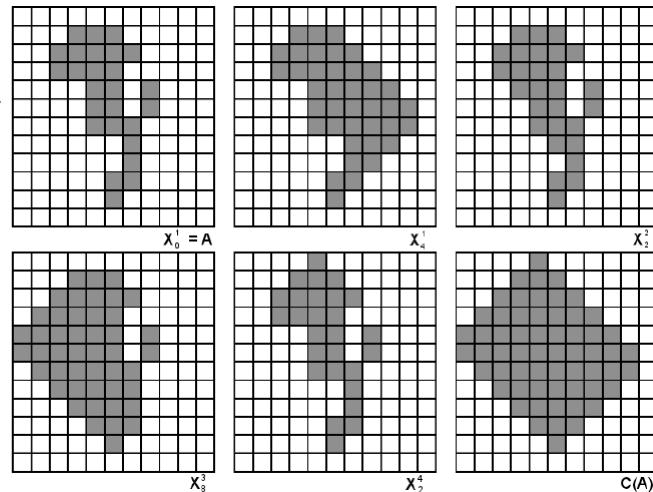
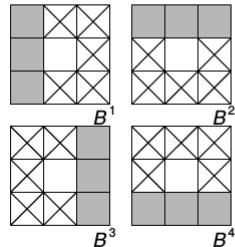


Figura 37: Convex Hull.

El conjunto crece más allá de lo mínimo necesario para garantizar convexidad (entonces es necesario limitar)

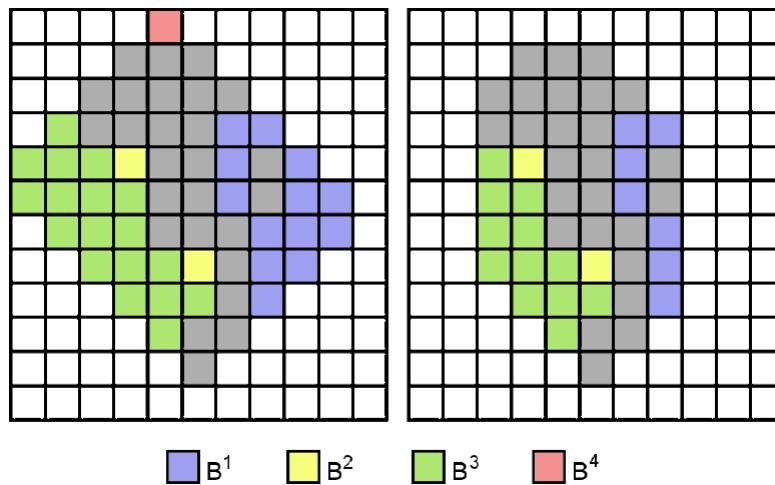


Figura 38: Convex Hull.

10.5. Adelgazamiento (Thinning)

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

El adelgazamiento simétrico de A , se puede definir de forma más útil basada en una secuencia de EE:

$$\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$$

B^i es una versión rotada de B^{i-1}

$$A \otimes \{B\} = ((\dots((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$$

10.6. Espesamiento (Thickening)

$$A \odot B = A \cup (A \circledast B)$$

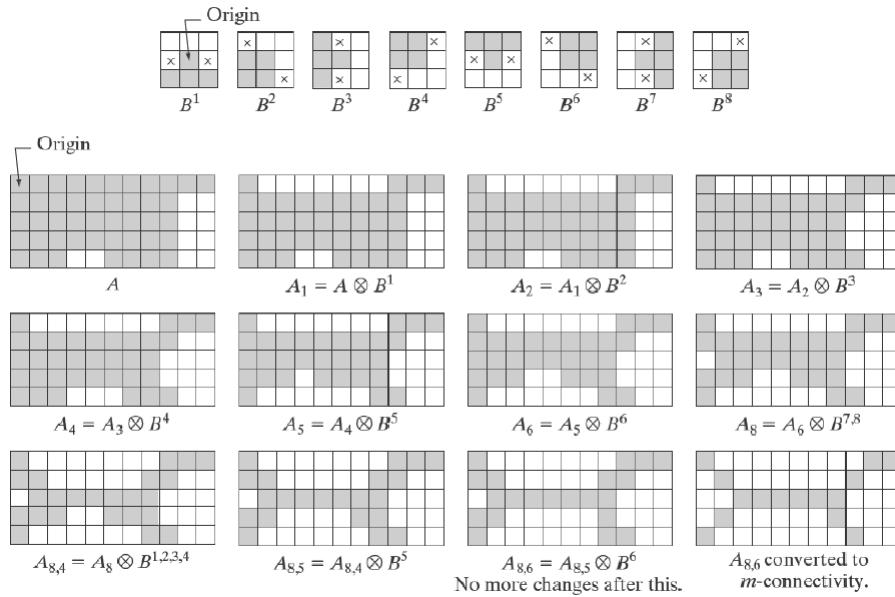


Figura 39: Thinning.

Similar al adelgazamiento simétrico de A , el espesamiento se puede definir de forma más útil basada en una secuencia de EE (complementos de los previos):

$$A \odot \{B\} = ((\dots((A \odot B^1) \odot B^2) \dots) \odot B^n)$$

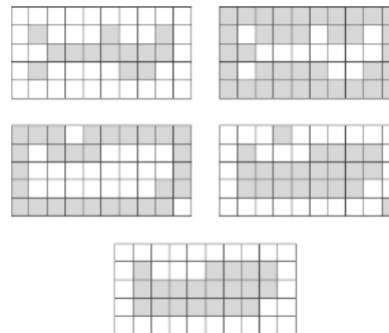


Figura 40: Thickening.

10.7. Esqueletos (Skeletons)

El esqueleto $S(A)$ de un conjunto A , puede deducirse según:

- Si $z \in S(A)$, se define $(D)_z \in A$ al mayor disco posible centrado en z .
- El disco máximo $(D)_z$ toca el borde de A en al menos 2 puntos.
- Serra [1982] definió el esqueleto en términos de erosión y apertura.

10.8. Reconstrucción morfológica

1. Dilatación geodésica (tamaño 1):

$$D_G^{(1)}(F) = (F \oplus G) \cap G, \quad F \subseteq G$$

2. Dilatación geodésica (tamaño n):

$$D_G^{(n)}(F) = D_G^{(1)} \left[D_G^{(n-1)}(F) \right], \quad D_G^{(0)}(F) = F$$

$$S(A) = \bigcup_{k=0}^K S_k(A) \quad \text{con} \quad S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

$$K = \max\{k \mid (A \ominus kB) \neq \emptyset\}$$

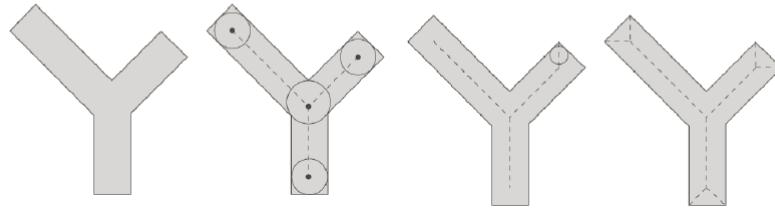


Figura 41: Skeletons.

$k \backslash$	$A \ominus kB$	$(A \ominus kB) \circ B$	$S_k(A)$	$\bigcup_{k=0}^K S_k(A)$	$S_k(A) \oplus kB$	$\bigcup_{k=0}^K S_k(A) \oplus kB$
0						
1						
2						

Figura 42: Skeletons.

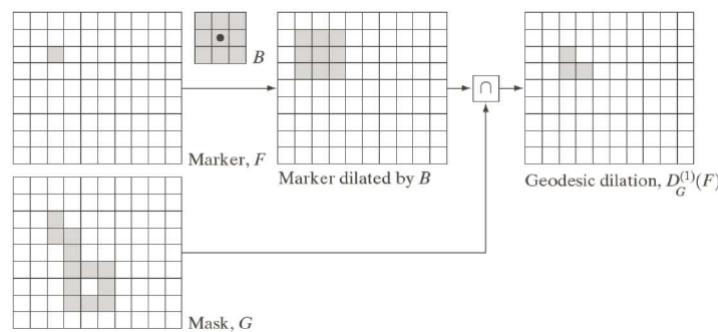


Figura 43: Reconstrucción Morfológica.

3. Erosión geodésica (tamaño 1):

$$E_G^{(1)}(F) = (F \ominus G) \cup G, \quad F \subseteq G$$

4. Erosión geodésica (tamaño n):

$$E_G^{(n)}(F) = E_G^{(1)} [E_G^{(n-1)}(F)], \quad E_G^{(0)}(F) = F$$

5. Reconstrucción por dilatación:

$$R_G^D(F) = D_G^{(k)}(F), \quad \text{hasta que} \quad D_G^{(k)}(F) = D_G^{(k+1)}(F)$$

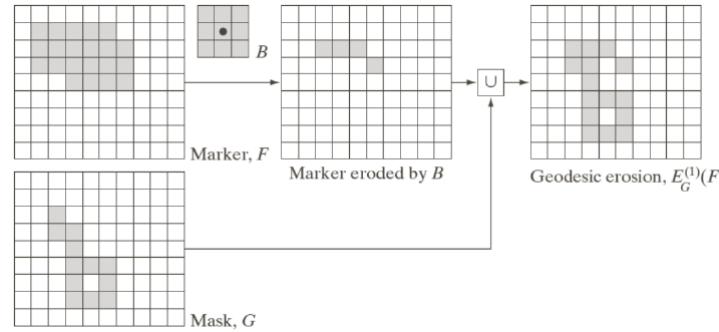


Figura 44: Reconstrucción Morfológica.

6. Reconstrucción por erosión:

$$R_G^E(F) = E_G^{(k)}(F), \quad \text{hasta que} \quad E_G^{(k)}(F) = E_G^{(k+1)}(F)$$

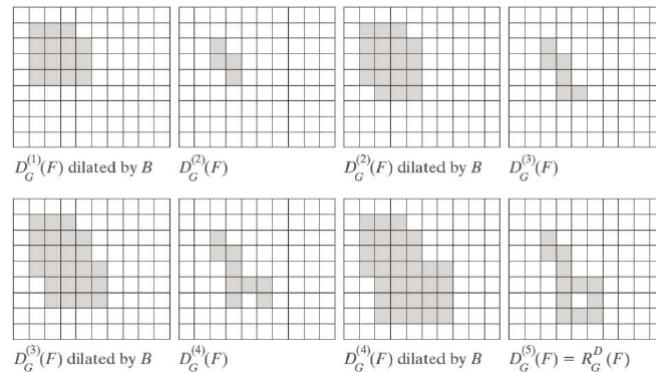


Figura 45: Reconstrucción Morfológica.

10.9. Aplicaciones de Reconstrucción Morfológica

1. Apertura por reconstrucción:

$$O_R^{(n)}(F) = R_F^D [(F \ominus nB)], \text{ para } n \text{ erosiones de } F \text{ y } B$$

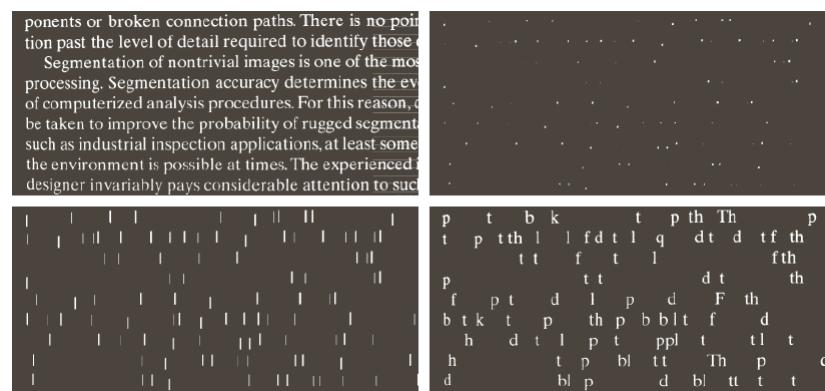


Figura 46: Aplicaciones de Reconstrucción Morfológica.

2. Relleno de agujeros automático:

$$F(x, y) = \begin{cases} 1 - I(x, y) & \text{si } (x, y) \text{ pertenecen al borde de } I \\ 0 & \text{en otro caso} \end{cases}$$

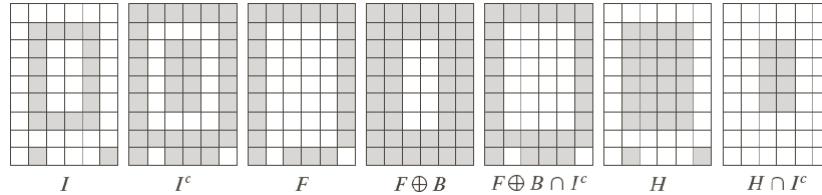


Figura 47: Aplicaciones de Reconstrucción Morfológica.

ponents or broken connection paths. There is no position past the level of detail required to identify those.

Segmentation of nontrivial images is one of the most processing. Segmentation accuracy determines the ev of computerized analysis procedures. For this reason, c be taken to improve the probability of rugged segment such as industrial inspection applications, at least some the environment is possible at times. The experienced i designer invariably pays considerable attention to suc

ponents or broken connection paths. There is no position past the level of detail required to identify those.

Segmentation of nontrivial images is one of the most processing. Segmentation accuracy determines the ev of computerized analysis procedures. For this reason, c be taken to improve the probability of rugged segment such as industrial inspection applications, at least some the environment is possible at times. The experienced i designer invariably pays considerable attention to suc



ponents or broken connection paths. There is no position past the level of detail required to identify those.

Segmentation of nontrivial images is one of the most processing. Segmentation accuracy determines the ev of computerized analysis procedures. For this reason, c be taken to improve the probability of rugged segment such as industrial inspection applications, at least some the environment is possible at times. The experienced i designer invariably pays considerable attention to suc

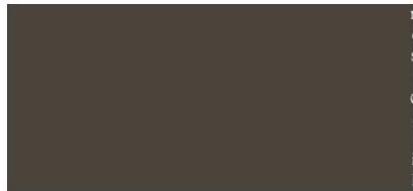
Figura 48: Aplicaciones de Reconstrucción Morfológica.

La imagen, similar a I , con los agujeros rellenos se obtiene por: $H = [R_{I^c}^D(F)]^c$

3. Limpieza de objetos en el borde:

$$F(x, y) = \begin{cases} I(x, y) & \text{si pertenecen al borde } I \\ 0 & \text{en otro caso} \end{cases}$$

La imagen, similar a I , sin objetos que tocan el borde: $X = I - R_I^D(F)$



ponents or broken connection paths. There is no position past the level of detail required to identify those.

Segmentation of nontrivial images is one of the most processing. Segmentation accuracy determines the ev of computerized analysis procedures. For this reason, c be taken to improve the probability of rugged segment such as industrial inspection applications, at least some the environment is possible at times. The experienced i designer invariably pays considerable attention to suc

Figura 49: Aplicaciones de Reconstrucción Morfológica.

Restauración mediante FILTRADO ESPACIAL		
METODO	VENTAJA	DESVENTAJA
tipo: Filtros de medias		
FILTRO DE MEDIA ARITMÉTICA	Suaviza variaciones locales de la imagen reduciendo el ruido a partir del desenfoque.	Produce desenfoque en la imagen.
FILTRO DE MEDIA GEOMÉTRICA	Suaviza la imagen con menor pérdida de detalle. Bueno para ruido Gausiano.	Falla en ruido impulsivo.
FILTRO DE MEDIA ARMÓNICA	Preserva detalles de la imagen. Bueno para ruido gausiano y tipo sal.	Falla en ruido pimienta.
FILTRO DE MEDIA CONTRA-ARMÓNICA	Bueno para ruido sal y pimienta. $Q>0$: Elimina pimienta. $Q<0$: Elimina sal. $Q=0$: Filtro media aritmética. $Q=-1$: Filtro media armónica.	No se pueden eliminar ambos tipos de ruido al mismo tiempo.
tipo: Filtros de orden		
FILTRO DE MEDIANA (media de la vecindad)	Buena reducción de ruido impulsivo. No produce desenfoque.	Filtro grande produce degradación.
FILTRO DE MODA (más frecuente de vecindad)	Buena reducción de ruido impulsivo.	Malo en el resto de ruidos.
FILTRO DE MÁXIMA	Util para ruido tipo pimienta.	Malo para tipo sal.
FILTRO DE MÍNIMA	Util para ruido tipo sal.	Malo para tipo pimienta.
FILTRO DEL PUNTO MEDIO	Util para ruido gaussiano o uniforme.	-
FILTRO DE LA MEDIA ALFA-RECORTADO	Util para combinaciones de tipo gausiano y sal y pimienta.	-
tipo: Filtros adaptativos		
FILTRO DE REDUCCIÓN LOCAL DEL RUIDO	Mejores resultados que los anteriores.	Mayor complejidad. Se necesita conocer la varianza del ruido o estimarla de forma precisa.