# Filtros en el dominio espacial

## Lección 05.1

Dr. Pablo Alvarado Moya

MP6123 Procesamiento Digital de Imágenes
Programa de Maestría en Electrónica
Énfasis en Procesamiento Digital de Señales
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica

II Cuatrimestre 2012

# Contenido

# Operaciones espaciales

1. Operaciones geométricas
2. Operaciones sobre valores de píxeles
3. Transformaciones de dominio
4. Operaciones sobre los valores de vecindades

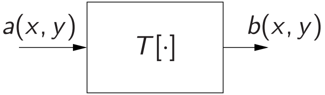
# Operaciones sobre los valores de vecindades

Para imagen de entrada $\mathcal{A}$, la imagen de salida $\mathcal{B}$ es

$$\mathcal{B} = \{\mathbf{b} \mid \mathbf{b} = \langle \underline{\mathbf{x}}, \underline{\mathbf{d}} \rangle \,, \quad \underline{\mathbf{d}} = T(\mathrm{val}(N(\mathbf{a}))), \quad \mathbf{a} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle \in \mathcal{A}\}$$

# Teoría de sistemas

## Tres tareas básicas

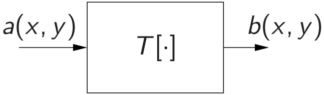



1. **Simulación** de sistemas: $T$ y $a$ conocidos, determinar $b$
2. **Reconstrucción** de señales: $T$ y $b$ conocidos, determinar $a$
3. **Identificación** de sistemas: $a$ y $b$ conocidos, determinar $T$

# Diferencias de PDI con PDS tradicional

1. la señales se definen en el espacio y no en el tiempo
2. dos variables independientes espaciales en vez de una temporal.

# Linealidad



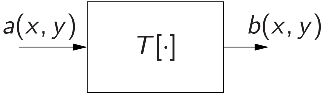

Si $b_i(x, y) = T[a_i(x, y)]$ entonces

$$T[k_1 a_1(x, y) + k_2 a_2(x, y)] = k_1 T[a_1(x, y)] + k_2 T[a_2(x, y)]$$

o en general

$$T\left[\sum_{k=1}^{N} k_i a_i(x, y)\right] = \sum_{k=1}^{N} k_i T[a_i(x, y)] = \sum_{k=1}^{N} k_i b_i(x, y)$$

# Invarianza a la traslación




Si $b(x,y) = T[a(x,y)]$ entonces:

$$b(x - x_0, y - y_0) = T[a(x - x_0, y - y_0)]$$

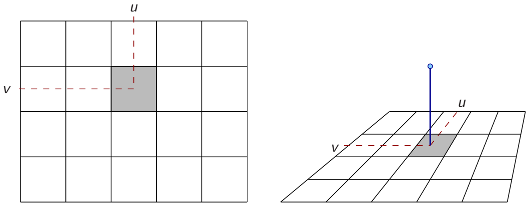
# Sistemas LTS

**LTS**: **L**inear and **S**hift **I**nvariant
( Equivalente de sistemas LTI en el espacio )

# Impulso unitario bidimensional

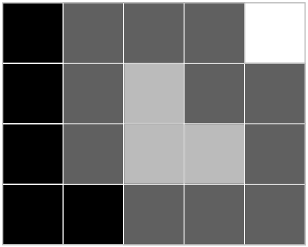
$$\delta(x,y) = \delta(x)\delta(y) = \begin{cases} 1 & \text{si } x = y = 0 \\ 0 & \text{en el resto} \end{cases}$$

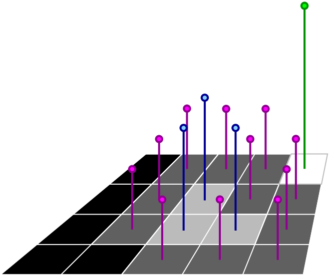
Propiedad de muestreo:




Píxel en posición $(u,v)$: $f(x,y)\delta(x-u,y-v)$

# Imagen como suma de píxeles





$$f(x,y) = \sum_{(u,v)} f(u,v)\delta(x-u,y-v) = f(x,y) * \delta(x,y)$$

# Convolución

Sea $T$ un sistema LSI

$$
\begin{aligned}
g(x,y) = T[f(x,y)] &= T\left[\sum_{(u,v)} f(u,v)\delta(x-u,y-v)\right] \\
&= \sum_{(u,v)} T\left[f(u,v)\delta(x-u,y-v)\right] \\
&= \sum_{(u,v)} f(u,v)\,T\left[\delta(x-u,y-v)\right] \\
&= \sum_{(u,v)} f(u,v)h(x-u,y-v) \\
&= f(x,y) * h(x,y)
\end{aligned}
$$

con respuesta al impulso $h(x,y) = T[\delta(x,y)]$

# Función de dispersión puntual

La respuesta al impulso $h(x, y) = T[\delta(x, y)]$ llamada en PDI también **función de dispersión puntual** o PSF por *point spread function*

# Pasos de la convolución

$$f(x,y) * h(x,y) = \sum_{(u,v)} f(u,v)h(x-u,y-v)$$

Cuatro pasos:

1. Invertir $h$ tanto en $x$ como en $y$ (rotación en 180°)
2. Trasladar origen de $h$ invertida a píxel de interés en $(x,y)$
3. Multiplicar punto a punto $h$ invertida y trasladada con $x$
4. Sumar todos los productos anteriores y asignarlos al resultado $g(x,y)$.

# Otra interpretación

## Convolución

$$f(x,y) * h(x,y) = \sum_{(u,v)} f(u,v)h(x-u,y-v)$$

superposición de la PSF desplazada a cada $(u,v)$ y amplificada por el respectivo $f(u,v)$ (sistema LSI).

# Propiedades
## Convolución

- Conmutatividad:

$$a(x, y) * b(x, y) = b(x, y) * a(x, y)$$

- Asociatividad:

$$\begin{aligned} a(x, y) * b(x, y) * c(x, y) &= [a(x, y) * b(x, y)] * c(x, y) \\ &= a(x, y) * [b(x, y) * c(x, y)] \end{aligned}$$

- Distributividad:

$$a(x, y) * (b(x, y) + c(x, y)) = a(x, y) * b(x, y) + a(x, y) * c(x, y)$$

# Costo de la convolución

- Imagen $f(x, y)$ tiene $R$ filas y $C$ columnas
- $h(x, y)$ (kernel, núcleo, máscara, PSF) tiene $N$ filas y $M$ columnas con $x \in \{x_i, \ldots, x_f\}$, $y \in \{y_i, \ldots, y_f\}$, $N = y_f - y_i + 1$, $M = x_f - x_i + 1$
- La convolución se puede reescribir

$$f(x, y) * h(x, y) = \sum_{v=y-y_f}^{y-y_i} \sum_{u=x-x_f}^{x-x_i} f(u, v)h(x-u, y-v)$$

  para recorrer la máscara.
- Tamaño máscara: $K = NM$, tamaño imagen: $S = RC$
- Cada píxel: $K$ productos, $K - 1$ sumas
- Total operaciones: $SK$ productos, $S(K-1)$ sumas

# Filtro separable

Kernel $h(x,y)$ separable si

$$h(x,y) = h_h(x)h_v(y)$$

# Convolución con filtro separable

$$f(x,y) * h(x,y) = \sum_{v=y-y_f}^{y-y_i} \sum_{u=x-x_f}^{x-x_i} f(u,v) h_h(x-u) h_v(y-v)$$
$$= \sum_{v=y-y_f}^{y-y_i} h_v(y-v) \sum_{u=x-x_f}^{x-x_i} f(u,v) h_h(x-u)$$

La suma interna $g_x(x,v) = \sum_{u=x-x_f}^{x-x_i} f(u,v) h_h(x-u)$ es un filtrado unidimensional para cada fila $v$ de $f(x,y)$, y así:

$$f(x,y) * h(x,y) = \sum_{v=y-y_f}^{y-y_i} h(y-v) g_x(x,v)$$

# Costo de convolución separable

- Píxel en una fila: $M$ productos, $M-1$ sumas (Producto punto)
- Fila: $MC$ productos y $C(M-1)$ sumas
- Todas las filas $MRC = MS$ productos, $RC(M-1) = S(M-1)$ sumas.
- Píxel en una columna: $N$ productos, $N-1$ sumas.
- Columna: $NR$ productos y $R(N-1)$ sumas.
- Todas las columnas $NRC = NS$ productos, $RC(N-1) = S(N-1)$ sumas.
- Gran total: $S(N+M)$ productos, $S(N+M-2)$ sumas

## Comparación de máscaras separables y no separables

- Asúmase $N = M$, $R = C$ (máscara e imagen cuadradas)
- Máscara no separable: $R^2 N^2$ productos, $R^2(N^2 - 1)$
- Máscara separable: $R^2 2N$ productos, $R^2(2N - 2)$ sumas
- $\mathcal{O}(N^2)$ contra $\mathcal{O}(N)$

# Correlación

- Correlación:

$$f(x,y) \circ h(x,y) = f(x,y) * h(-x,-y) = \sum_{u,v} f(u,v)h(x+u,y+v)$$

- La máscara **no** se rota, como la convolución
- Se utiliza para buscar patrones similares a $h$

# Resumen