



Visión por Computadora

Unidad III

“Procesamiento Digital de Imágenes”

Rogelio Ferreira Escutia



Contenido

- 1) Operaciones Individuales**
 - a) Transformaciones Punto a Punto**
 - b) Transformaciones de 2 Imágenes Punto a Punto**
- 2) Operaciones de Vecindad**
- 3) Transformaciones Lógicas**
- 4) Transformaciones Geométricas**

1) Operaciones Individuales

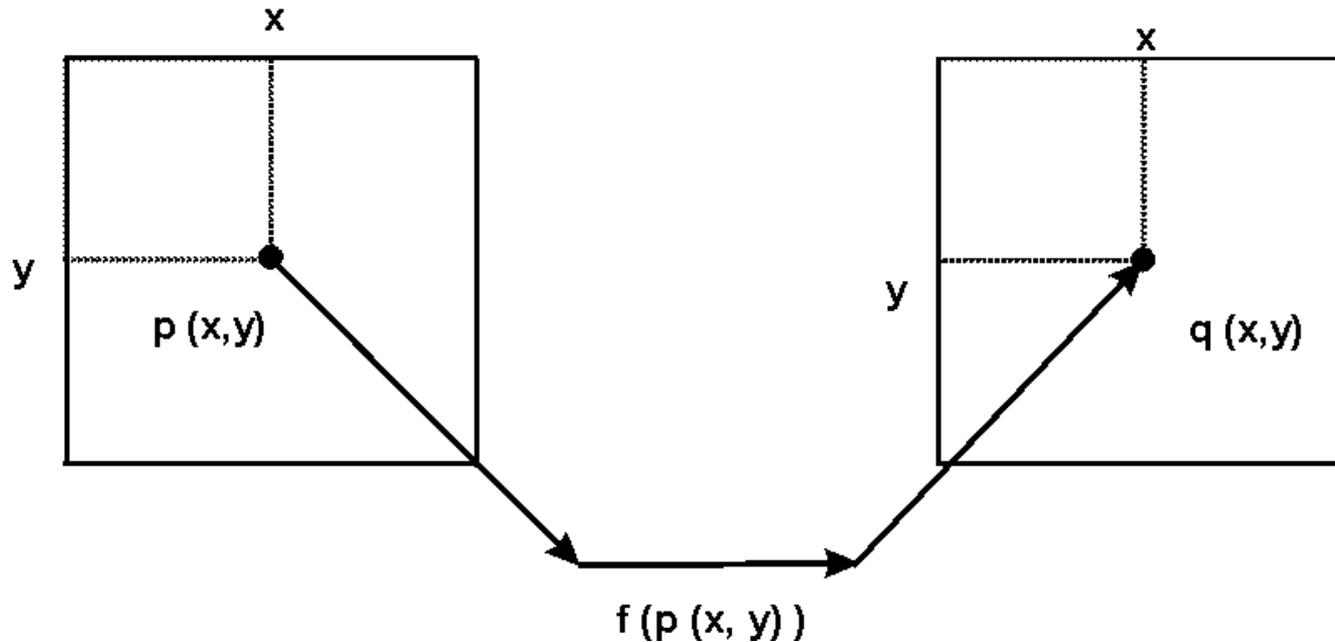
Operaciones Individuales

- **Las operaciones individuales implican la generación de una nueva imagen modificando el valor del pixel en una simple localización de la imagen original.**
- **El proceso consiste en obtener el valor del pixel de una localización dada en la imagen original.**
- **El proceso consiste en obtener el valor del pixel de una localización dada en la imagen, modificándolo por una operación lineal o no lineal y colocando el valor del nuevo pixel en la correspondiente localización de la nueva imagen.**
- **El proceso se repite para todas y cda una de las localizaciones de los pixeles en la imagen original.**

a) Transformaciones Punto a Punto

Operaciones Individuales

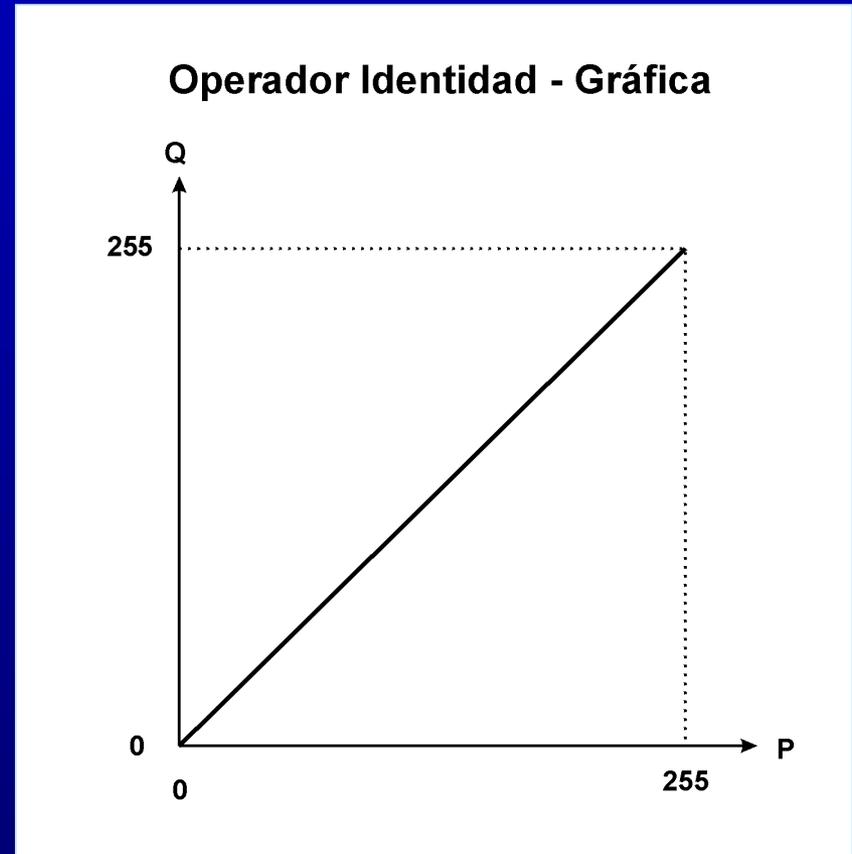
Transformación Punto a Punto



Operador Identidad

- Este operador crea una imagen de salida que es idéntica a la imagen de entrada.
- Función de transformación:

$$q = p$$



Operador Identidad



Imagen 1

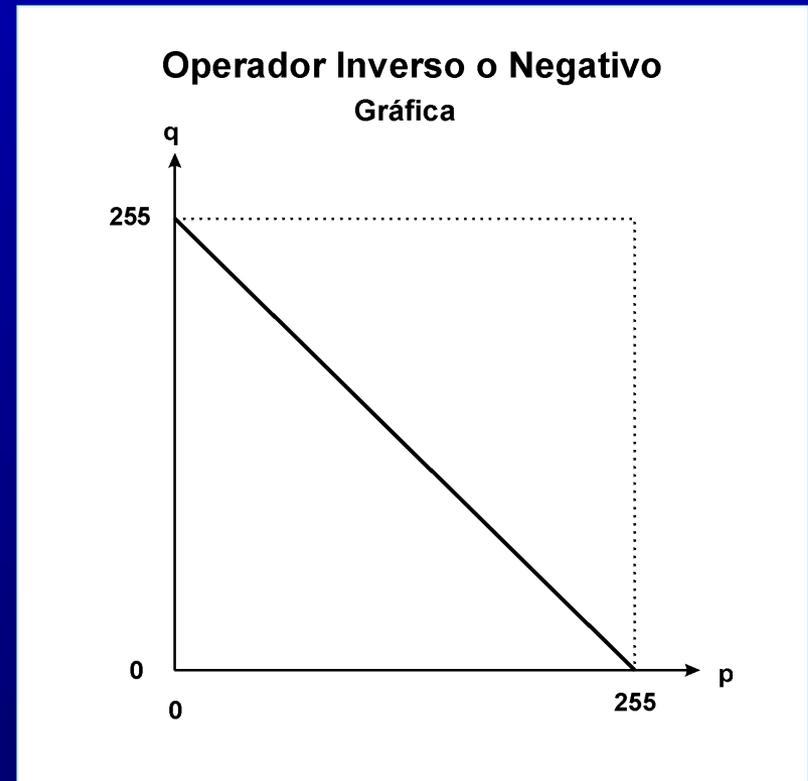


Transformación

Operador Inverso o Negativo

- Este operador crea una imagen de salida que es la inversa de la imagen de entrada.
- Este operador es útil en diversas aplicaciones tales como imágenes médicas.
- Para una imagen con valores de gris en el rango de 0 a 255 la función de transformación es:

$$q = \{ 255 - p$$



Operador Inverso o Negativo



Imagen 1

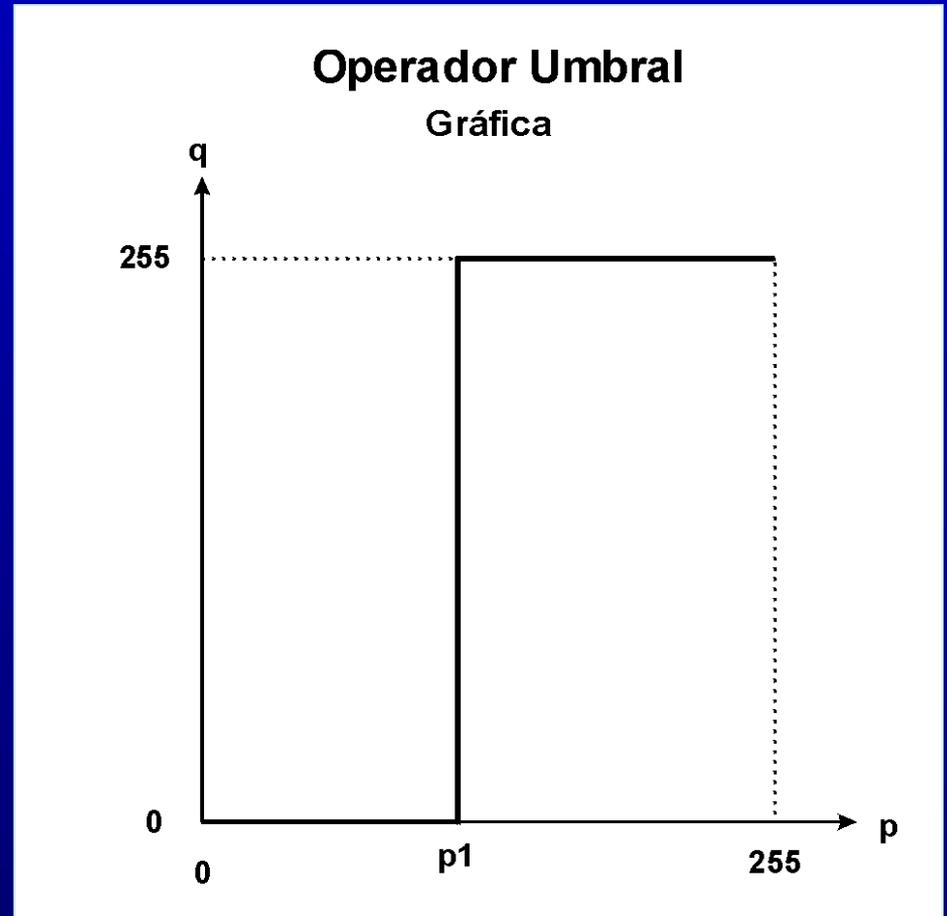


Transformación

Operador Umbral

- Se crea una imagen de salida binaria a partir de una imagen de grises, donde el nivel de transición está dado por el parámetro de entrada p_1 .
- Función de transformación:

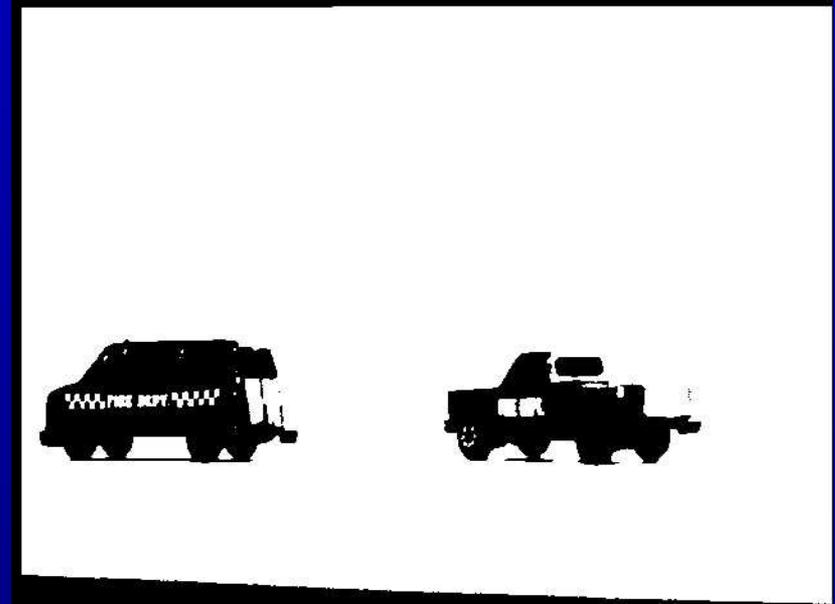
$$q = \begin{cases} 0 & \text{para } p \leq p_1 \\ 255 & \text{para } p > p_1 \end{cases}$$



Operador Umbral



Imagen 1



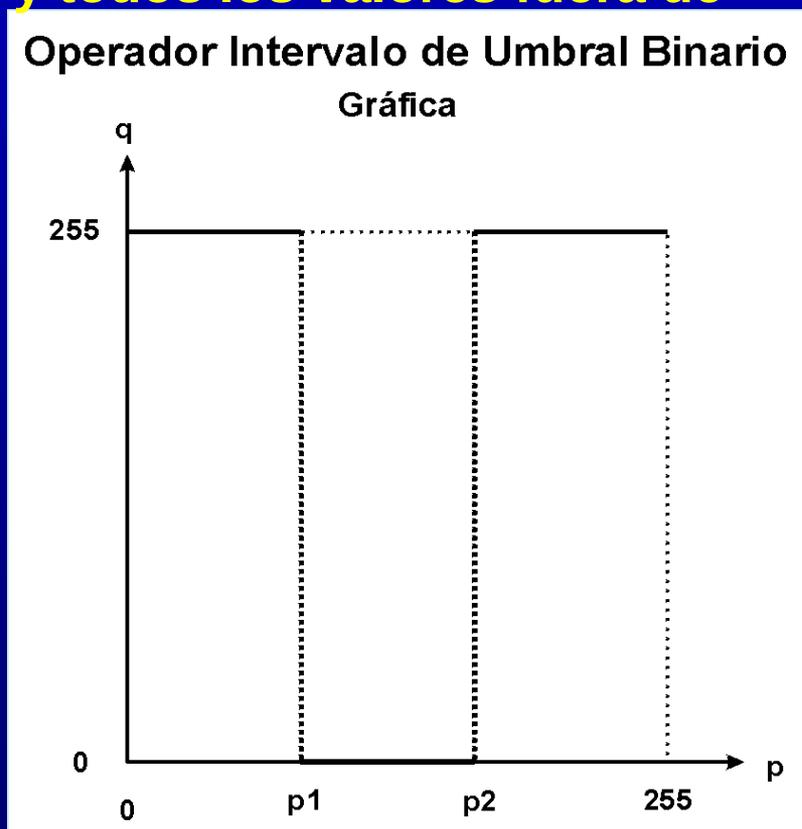
Transformación

$$p_1 = 150$$

Operador Intervalo de Umbral Binario

- Esta clase de transformación crea una imagen de salida binaria a partir de una imagen de grises, donde todos los valores de gris cuyo nivel está en el intervalo definido por p_1 y p_2 son transformados a 255, y todos los valores fuera de ese intervalo a 0.
- Función de transformación:

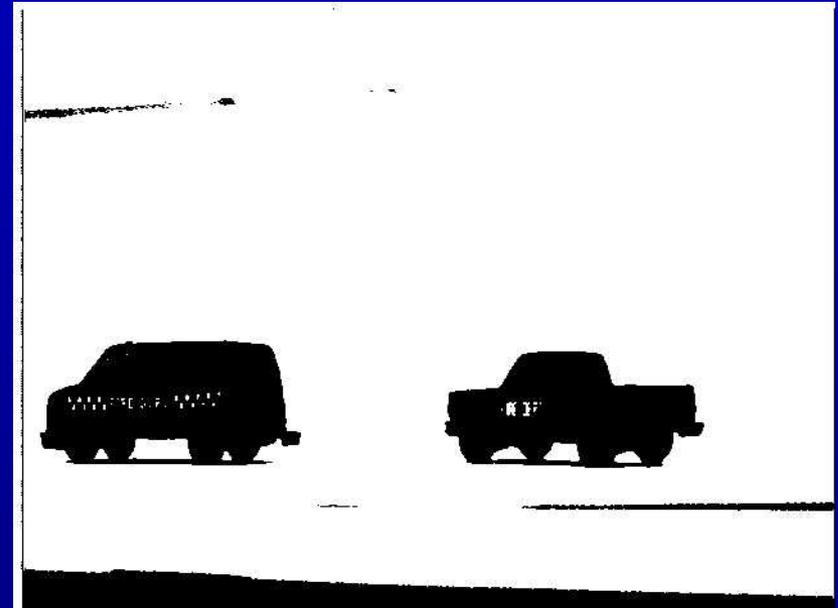
$$q = \begin{cases} 255 & \text{para } p \leq p_1 \text{ ó } p \geq p_2 \\ 0 & \text{para } p_1 < p < p_2 \end{cases}$$



Operador Intervalo de Umbral Binario



Imagen 1



Transformación

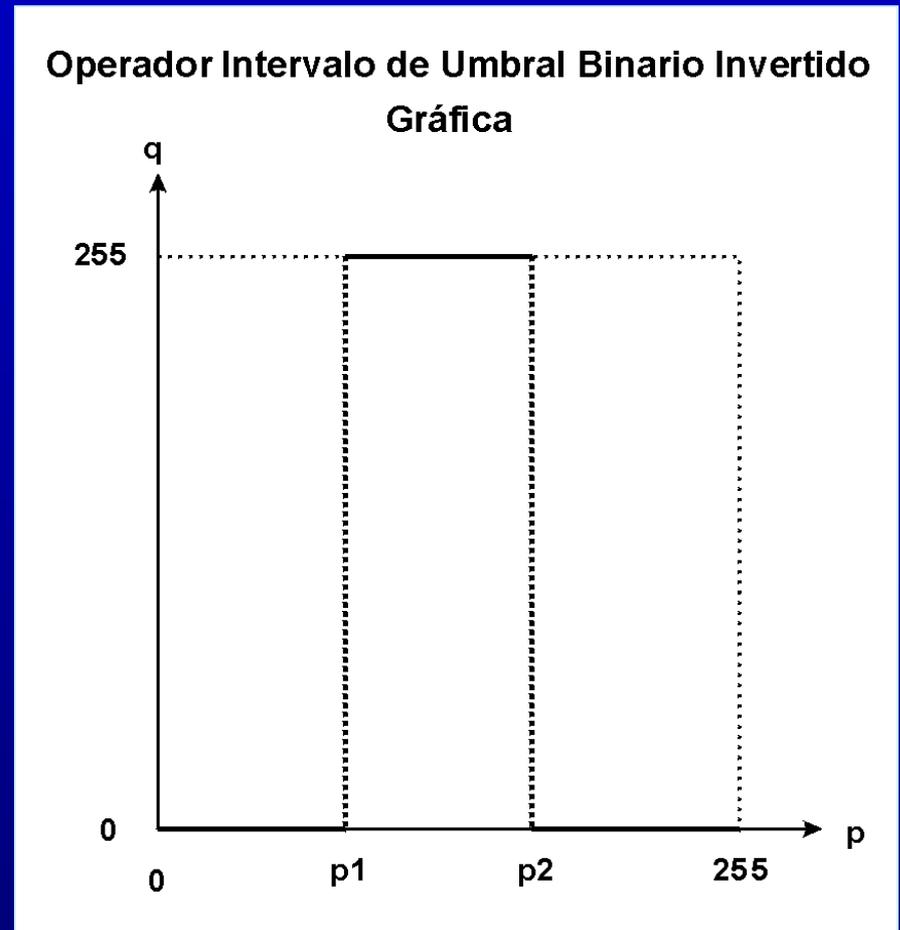
$$p_1 = 40 \quad p_2 = 190$$

Operador Intervalo de Umbral Binario Invertido

- Esta clase de transformación crea una imagen de salida binaria a partir de una imagen de grises, donde todos los valores de gris cuyos niveles están en el intervalo definido por p_1 y p_2 son transformados a 0, y todos los valores fuera de ese intervalo a 255.

- Función de transformación:

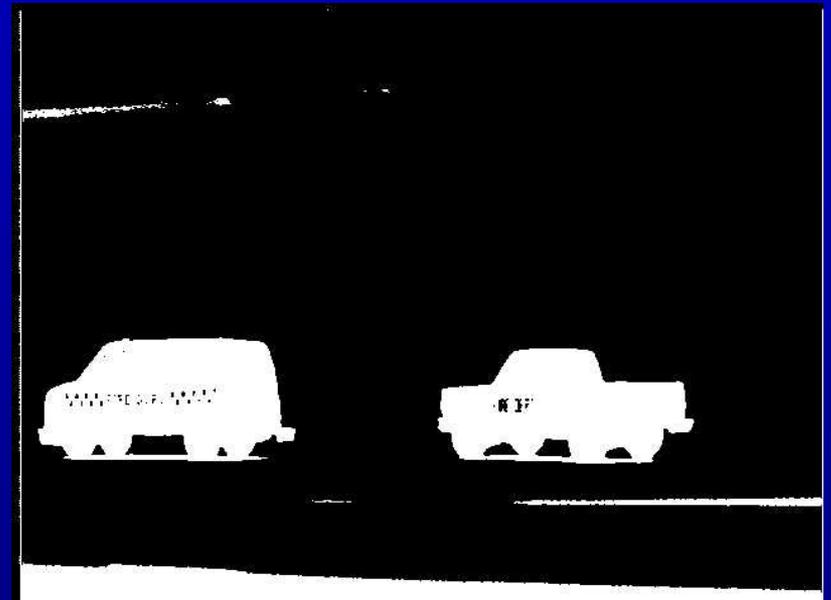
$$q = \begin{cases} 0 & \text{para } p \leq p_1 \text{ ó } p \geq p_2 \\ 255 & \text{para } p_1 < p < p_2 \end{cases}$$



Operador Intervalo de Umbral Binario Invertido



Imagen 1



Transformación

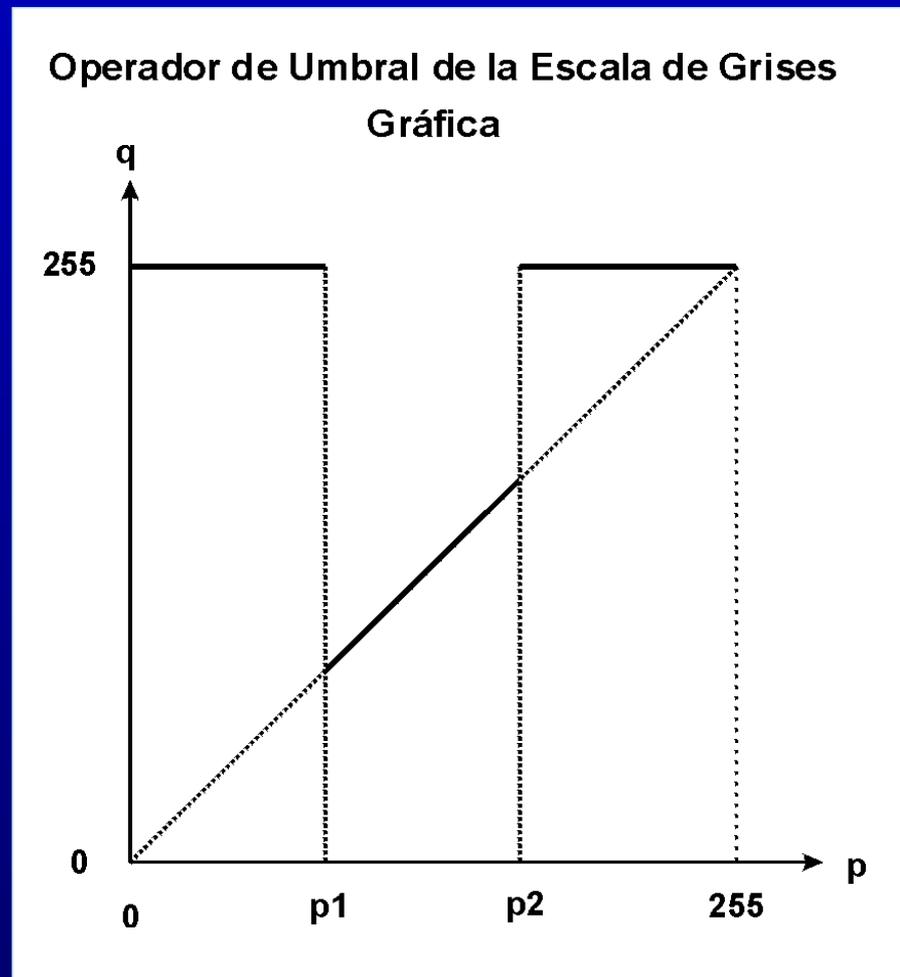
$$p_1 = 40 \quad p_2 = 190$$

Operador de Umbral de la Escala de Grises

- Se crea una imagen de salida con los únicos valores de nivel de gris comprendidos en el intervalo definido por p_1 y p_2 , y el resto a 255.

- Función de transformación:**

$$q = \begin{cases} 255 & \text{para } p \leq p_1 \text{ ó } p \geq p_2 \\ p & \text{para } p_1 < p < p_2 \end{cases}$$



Operador de Umbral de la Escala de Grises



Imagen 1



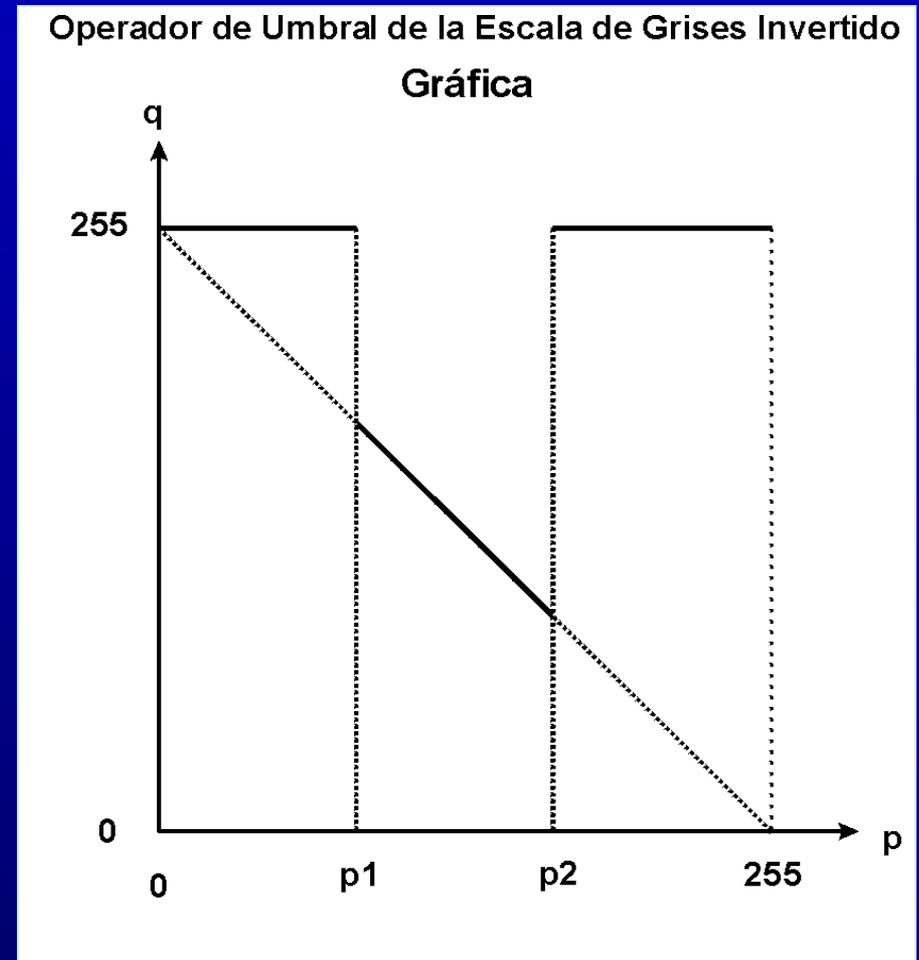
Transformación

$$p_1 = 40 \quad p_2 = 190$$

Operador de Umbral de la Escala de Grises Invertido

- Se crea una imagen de salida con los únicos valores de nivel de gris invertidos comprendidos en el intervalo definido por p_1 y p_2 , y el resto a 255.
- Función de transformación:

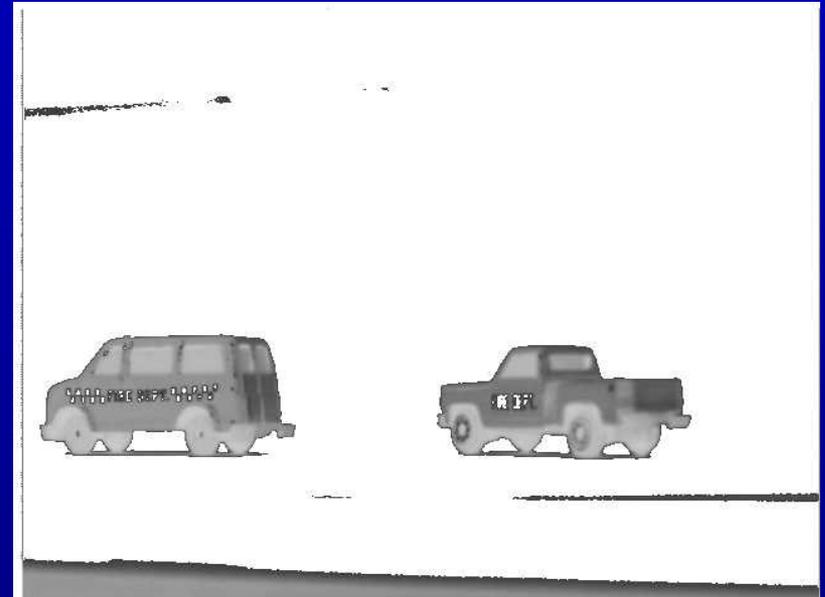
$$q = \begin{cases} 255 & \text{para } p \leq p_1 \text{ ó } p \geq p_2 \\ 255 - p & \text{para } p_1 < p < p_2 \end{cases}$$



Operador de Umbral de la Escala de Grises Invertido



Imagen 1



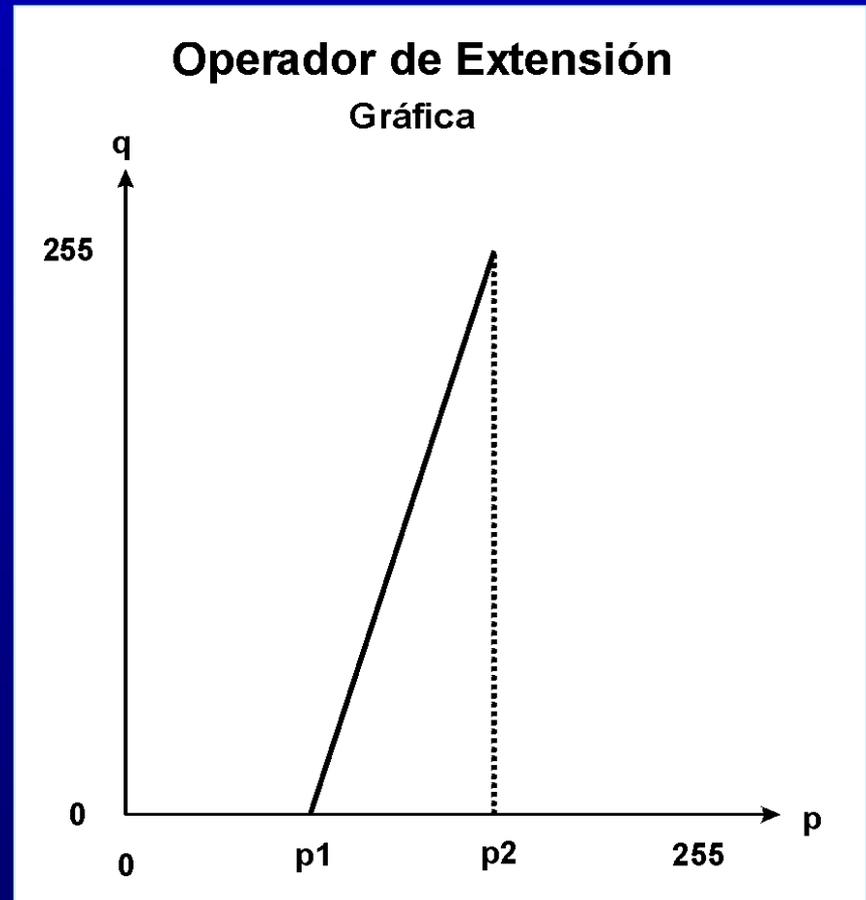
Transformación

$$p_1 = 40 \quad p_2 = 190$$

Operador de Extensión

- Proporciona una imagen de salida con la escala de grises completa correspondiente al intervalo de entrada definido por p_1 y p_2 , y suprime todos los valores fuera de este rango.
- Función de transformación:

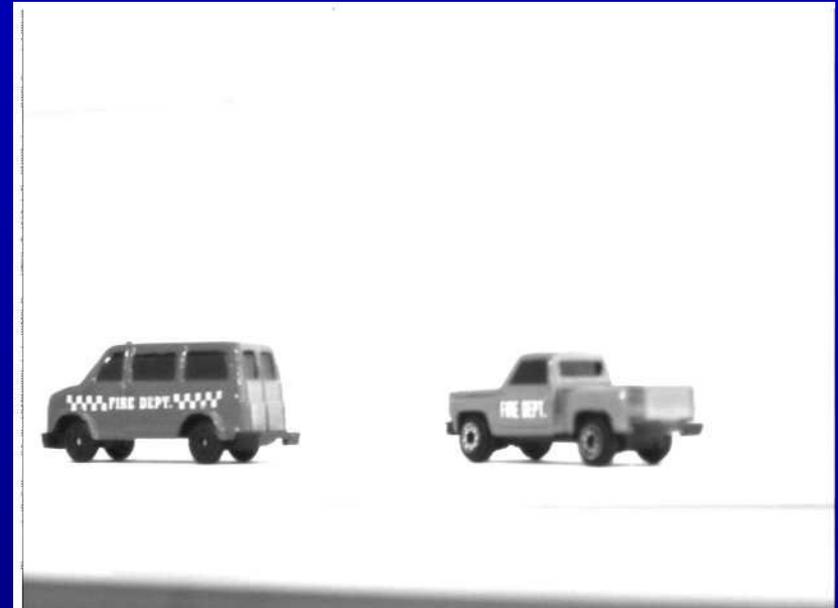
$$q = \begin{cases} 0 & \text{para } p \leq p_1 \text{ ó } p \geq p_2 \\ (p - p_1) \frac{255}{p_2 - p_1} & \text{para } p_1 < p < p_2 \end{cases}$$



Operador de Extensión



Imagen 1



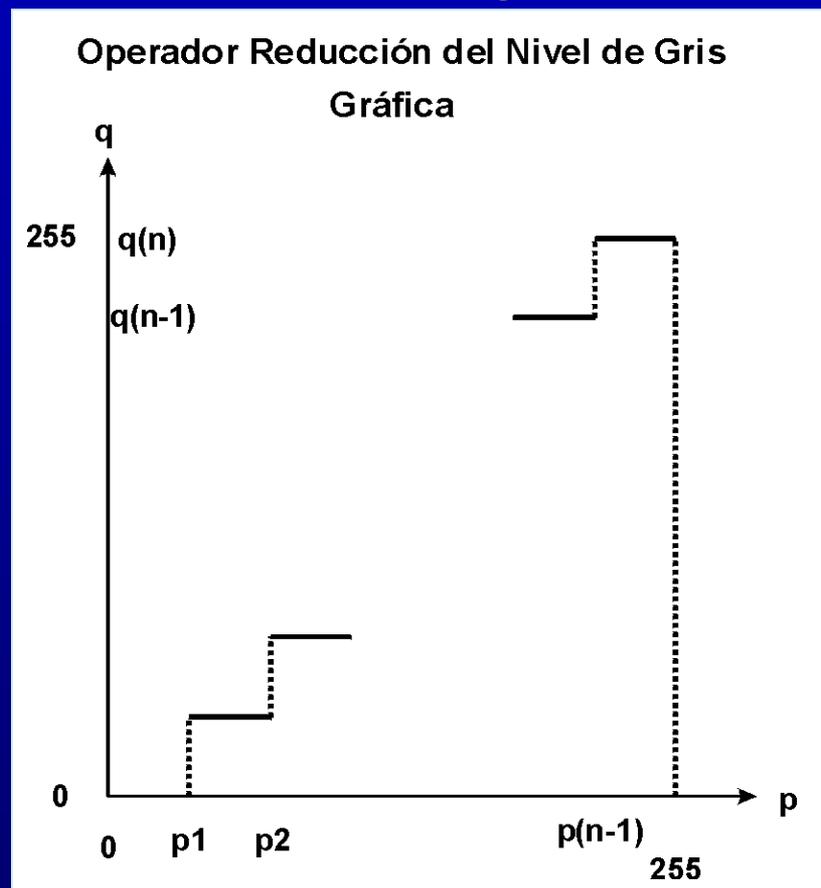
Transformación

$$p_1 = 40 \quad p_2 = 190$$

Operador Reducción del Nivel de Gris

- Proporciona una imagen de salida con un menor número de niveles de gris respecto de la imagen original de entrada, la imagen de entrada es reducida a $n+1$ niveles de gris.
- Función de transformación:**

$$q = \begin{cases} 0 & \text{para } p \leq p_1 \\ q_1 & \text{para } p_1 < p \leq p_2 \\ q_2 & \text{para } p_2 < p \leq p_3 \\ \dots & \\ q_n & \text{para } p_{n-1} < p \leq 255 \end{cases}$$



Operador Reducción del Nivel de Gris



Imagen 1



Transformación

$$\begin{aligned} n &= 10 & p_1 &= 25 & p_2 &= 50 & p_3 &= 75 & p_4 &= 100 \\ p_5 &= 125 & p_6 &= 150 & p_7 &= 175 & p_8 &= 200 \\ p_9 &= 225 & p_{10} &= 250 \end{aligned}$$

b) Transformaciones de 2 imágenes Punto a Punto

Transformaciones Punto a Punto

- Para esta transformación se utilizan 2 imágenes de entrada A y B, para crear una nueva imagen C.
- La dimensión de las imágenes es la misma.
- La función de la transformación f_D puede ser lineal o no.
- Esta función se aplica a todos los pares de píxeles en las imágenes de entrada.
- La función característica esta dada por la siguiente ecuación:

$$c_{x,y} = f_D(a_{x,y}, b_{x,y})$$

- La función f_D puede ser adición, sustracción, multiplicación, división, exponenciación, máximo, o cualquier otra función que se pueda definir.

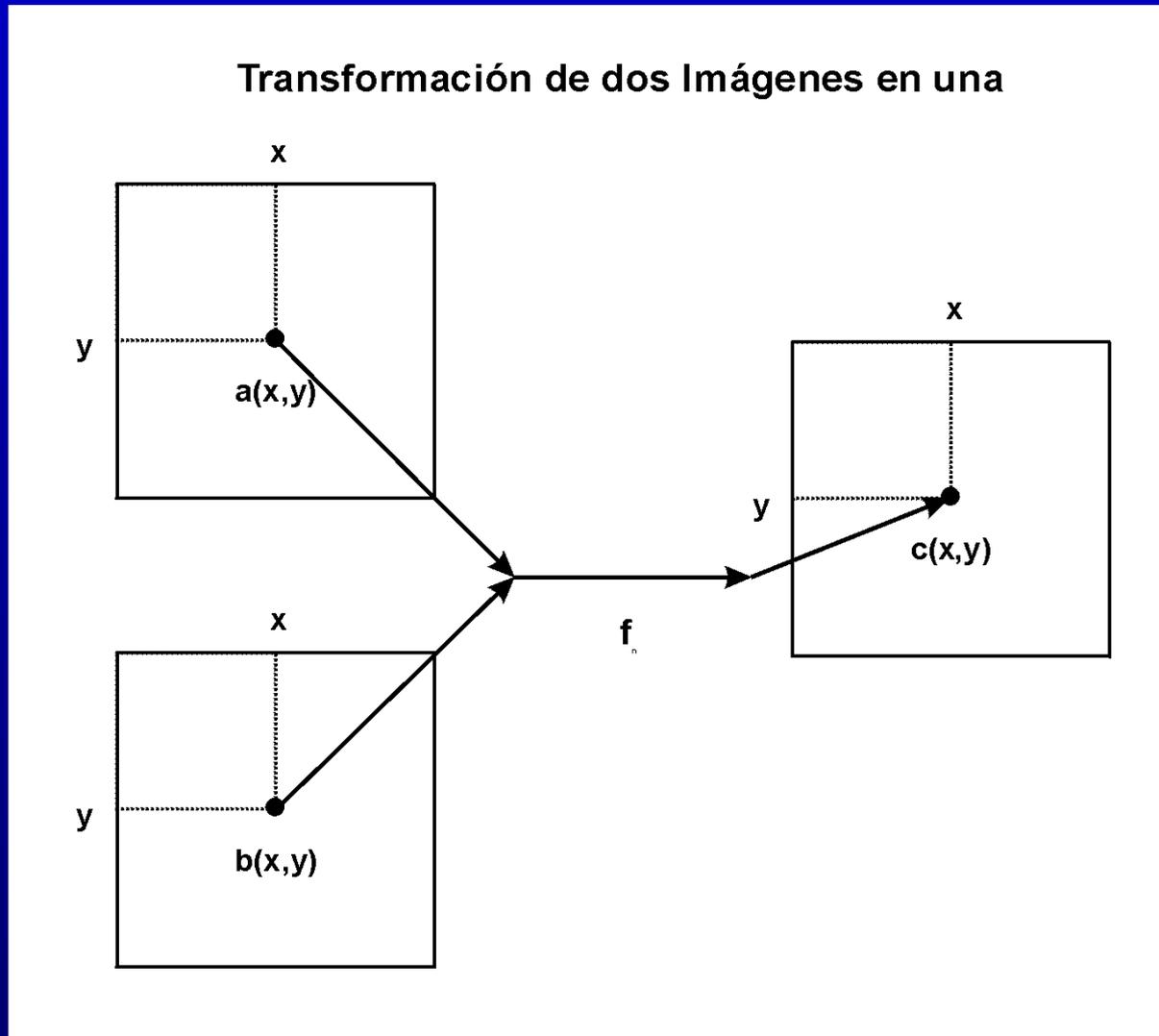
Transformaciones Punto a Punto (cont.)

- La función deberá tener un factor de escala apropiado k para mantener los valores de salida dentro de un rango adecuado, así como para evitar desbordamientos y valores negativos.
- La transformación implica dos variables asociadas con los pares de píxeles:

$$R_{x,y} = f_D(a(x,y), b(x,y))$$

- Donde a y b son las matrices de entrada, F_D es el operador funcional y R es la matriz resultante de salida.

Transformaciones Punto a Punto



Suma (Adición)

- La adición puede utilizarse para reducir los efectos del ruido en la imagen.
- El valor de salida es:

$$c_{x,y} = (a_{x,y} + b_{x,y}) / k$$

- Donde k es 2 para el caso de las 2 imágenes de entrada.
- Los valores de salida finales deben redondearse por defecto o por exceso.

Resta (Sustracción)

- La sustracción es técnica útil para detectar el cambio producido en dos imágenes que han sido captadas en 2 instantes de tiempo diferentes.
- Los datos de ambas imágenes también pueden representar pérdidas de calor o frío, o si la fuente de datos es el espectro infrarrojo.
- Puesto que en el procesamiento de imágenes se utilizan números positivos, es necesario definir la salida de alguna manera que haga todos los valores positivos, esto implica un reescalado donde el valor mas negativo se corresponde con el 0 y el máximo con el 255 para la escala de grises del 0 al 255.

$$c_{x,y} = k(a_{x,y} - b_{x,y})$$

- Donde k es una función no lineal de forma que el valor mínimo que toma $c_{x,y}$ es 0 y el máximo 255

Suma y Resta



Imagen 2



Imagen 3



Suma 2+3



Resta 2-3

Rodajas del Plano de Bits

- Lo que se busca es la contribución que hacen los diferentes bits a la imagen.
- Si cada pixel esta formado por 8 bits, supongamos que la imagen esta compuesta por 8 planos, cada uno con un bit significativo y el resto a cero.
- Desde el plano cero para el bit menos significativo al 7 para el bit mas significativo.
- Se trata de separar el contenido de la imagen entre los rangos $2^n \dots 2^{n+1}$ para $n = 0, 1 \dots 7$.

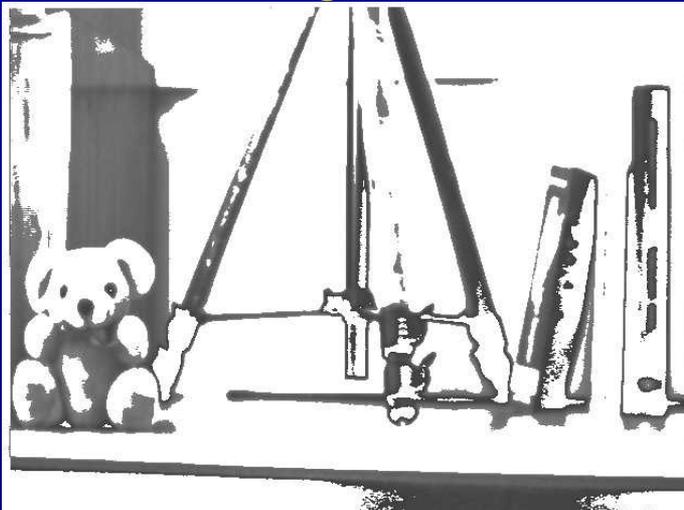
Rodajas del Plano de Bits



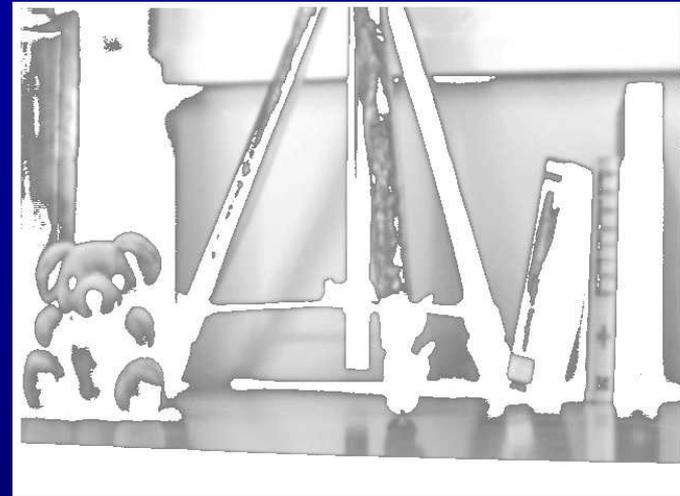
Imagen 2



n = 5



n = 6



n = 7

2) Operaciones de Vecindad

Operaciones de Vecindad

- En esta transformación, el nuevo valor del pixel en la imagen de salida, depende de la combinación de los valores de los pixeles en la vecindad del pixel de la imagen original que esta siendo transformada.
- Si se considera una vecindad, se realiza una suma ponderada con los valores de los 8 vecinos y el resultado de dicha suma es el valor del nuevo pixel q de la imagen de salida en la misma posición.
- Lo que resta definir son los valores de la ponderación, lo cual se hace generalmente definiendo una máscara con valores constantes, dicha máscara es un filtro.

Operaciones de Vecindad (cont.)

- Para la máscara siguiente:

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

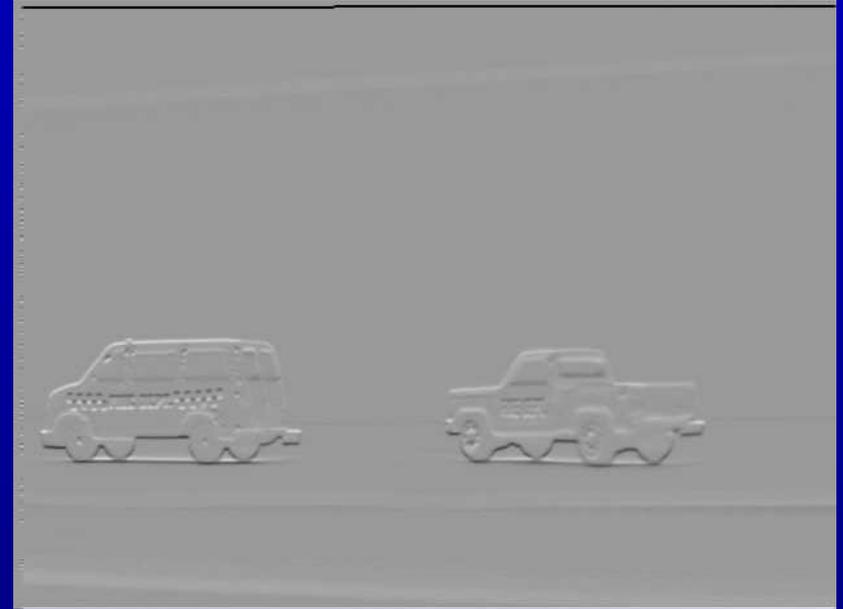
- El valor del pixel $q(x,y)$ esta dado por la suma ponderada, la cual nos da el efecto de Repujado en Relieve:

$$q(x, y) = 1 \cdot p(x-1, y-1) + 2 \cdot p(x, y-1) + 1 \cdot p(x+1, y-1) + \\ 0 \cdot p(x-1, y) + 0 \cdot p(x, y) + 0 \cdot p(x+1, y) \\ - 1 \cdot p(x-1, y+1) - 2 \cdot p(x, y+1) - 1 \cdot p(x+1, y+1)$$

Repujado en Relieve



Imagen 1



Transformación

Mayor Contraste

- **Para la máscara siguiente:**

$$\begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix}$$

- **Con esta máscara se obtiene un mayor contraste de la imagen original**

Mayor Contraste



Imagen 4



Transformación

3) Transformaciones Lógicas

Transformaciones Lógicas

- Dentro de las operaciones básicas se encuentra la binarización de una imagen, donde la imagen de salida sólo posee 2 niveles de gris, 0 y 255.
- Si se considera los números anteriores como el 0 lógico y el 1 lógico se obtiene una imagen binaria lógica.
- Con una imagen de estas características es posible realizar sobre ella todo tipo de operaciones lógicas, entre ellas negación, and, or, xor.
- También, dadas 2 imágenes, es posible realizar sobre ellas otro tipo de operaciones relacionales, tales como $<$, $>$, \leq , \geq , por ejemplo, dadas las imágenes A y B, una nueva imagen $C = (A \leq B)$, se obtiene realizando la comparación dada pixel a pixel y obteniendo el valor de 1 para C en aquellos pixeles donde se cumple la relación y 0 donde no se cumple.

Transformaciones Lógicas

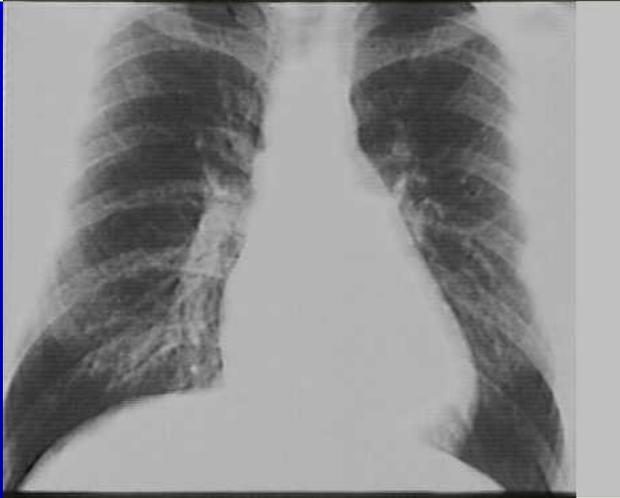


Imagen 5

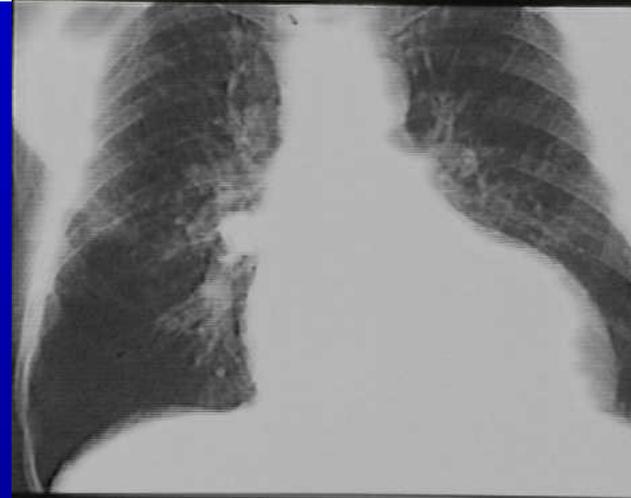
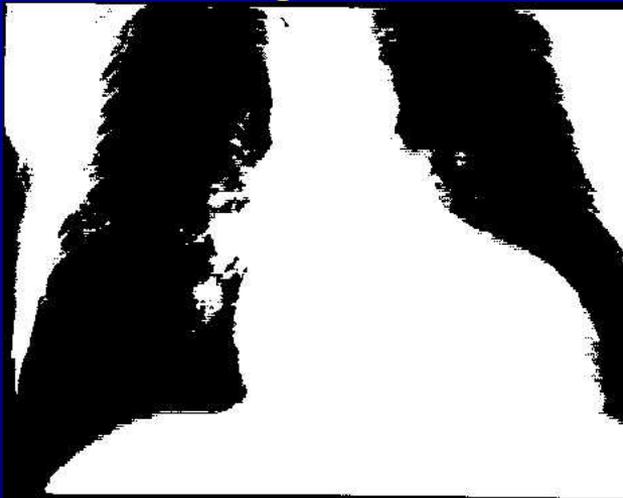
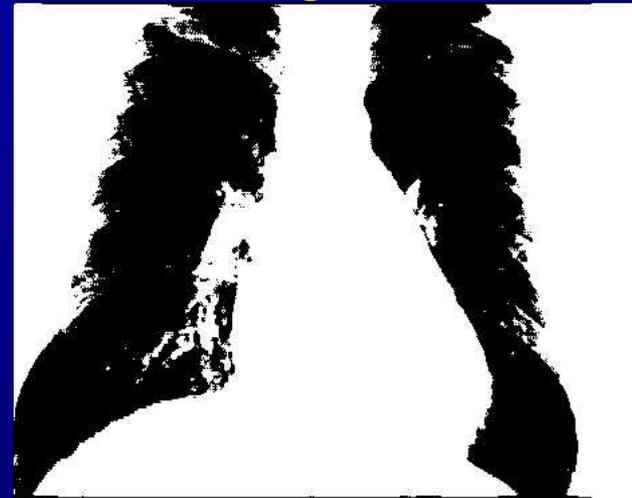


Imagen 6

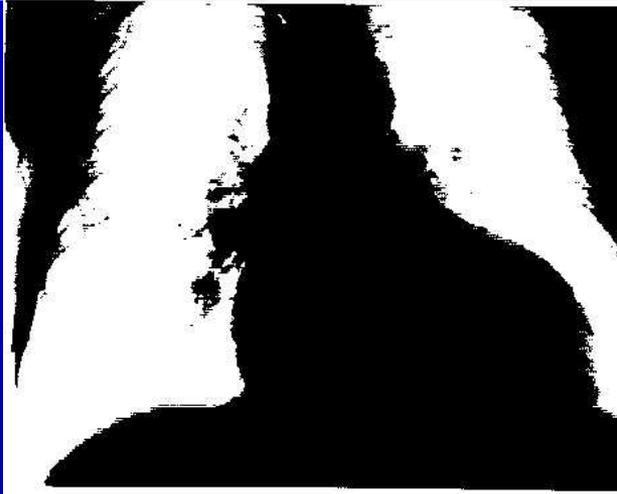


Binarización de 5 (umbral= 128)

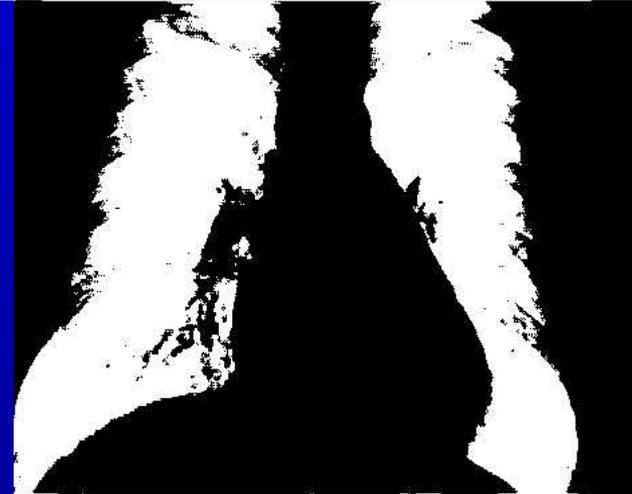


Binarización de 6 (umbral= 128)

Transformaciones Lógicas



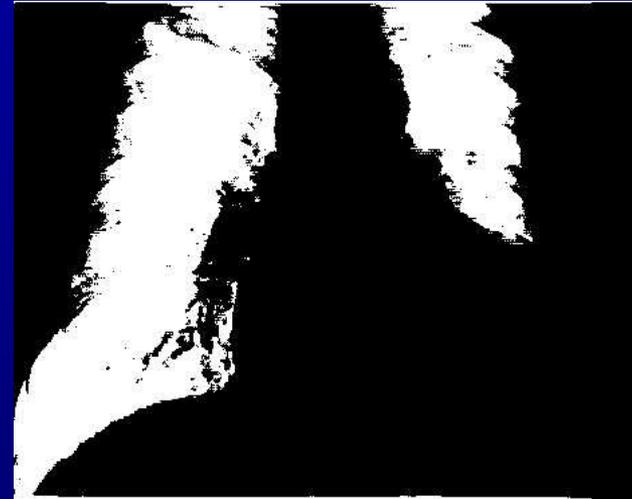
Negación de la Binarización de 5



Negación de la Binarización de 6

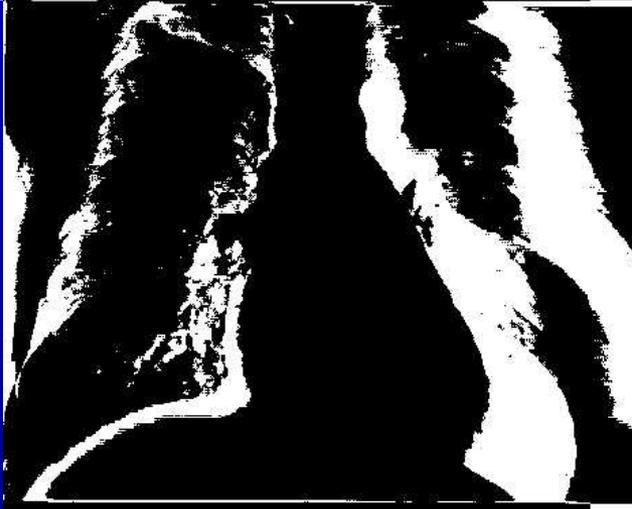


Operación OR entre 5 y 6



Operación AND entre 5 y 6

Transformaciones Lógicas



Operación XOR entre 5 y 6



Transformación $5 \leq 6$

4) Transformaciones Geométricas

Transformaciones Geométricas

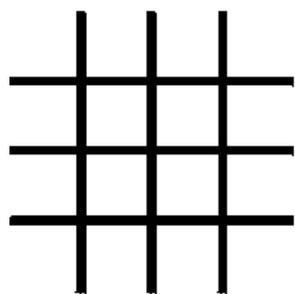
- **El objetivo fundamental de una operación geométrica es la transformación de los valores de una imagen tal y como podría observarse desde otro punto de vista.**
- **Las operaciones de ampliar o reducir una imagen, no es sino aproximar o alejar el punto de vista, rotarla equivale a girar el punto de observación, trasladarla es hacer lo propio con dicho punto.**

Desplazamiento de Pixeles

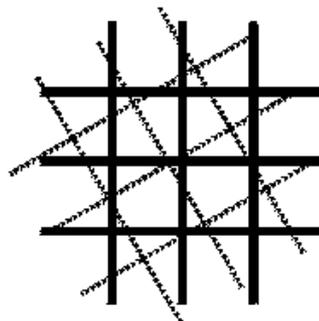
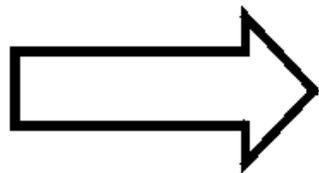
- En una imagen original, los pixeles se encuentran dentro de una estructura matricial (retícula) es decir, una rejilla.
- Al transformar esta rejilla según un desplazamiento, un giro o un acercamiento, los nuevos pixeles ya no tienen por qué quedar situados sobre tales intersecciones y caerán, por lo general, sobre puntos intermedios de ellos.
- Al tener que proyectar estos pixeles sobre los de la imagen final, es necesario calcular los valores de los pixeles finales en función de los transformados

Desplazamiento de Pixeles

Transformación Geométrica



Giro



los pixeles no concuerdan
al realizar el giro

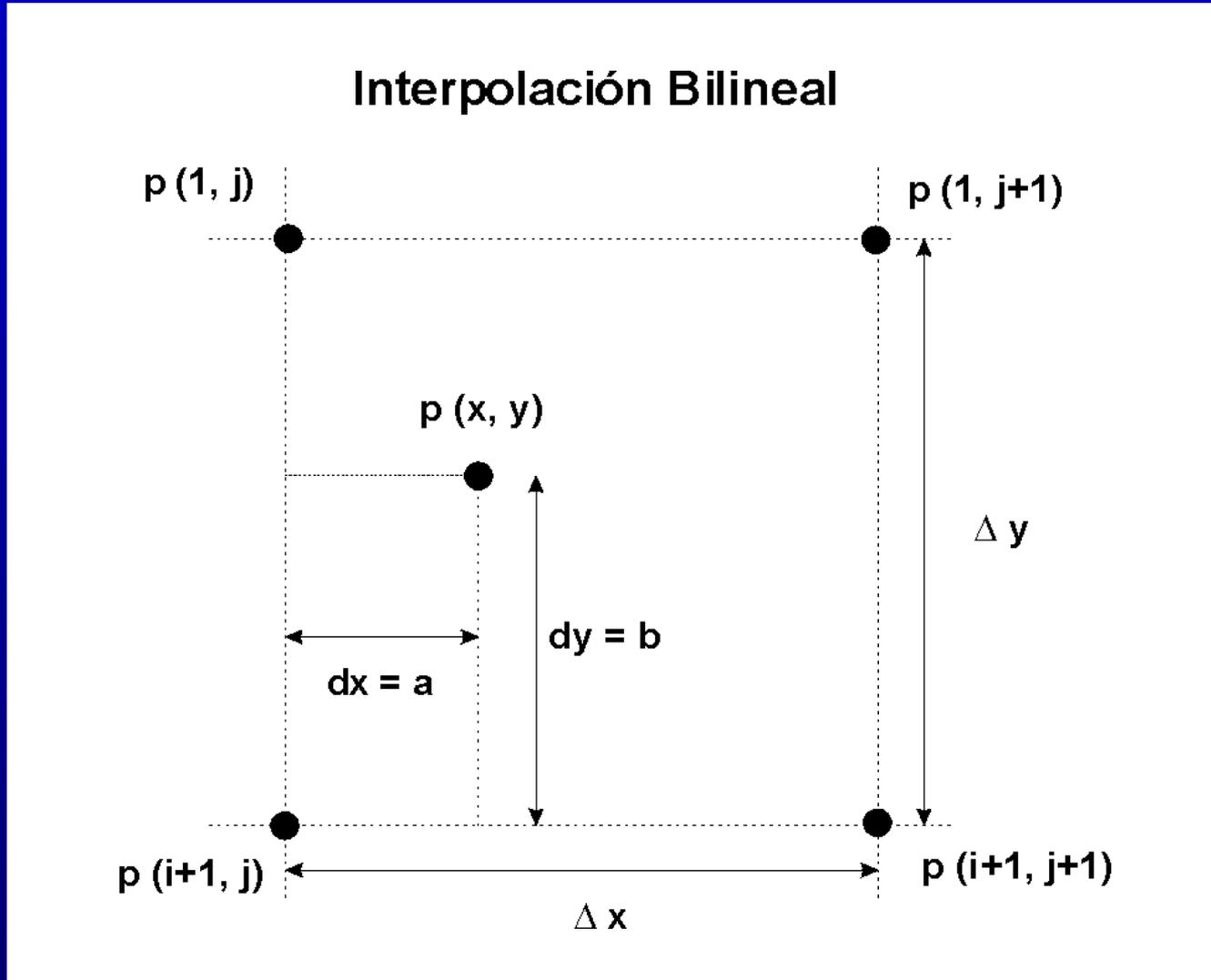
Interpolación

- **Para encontrar los nuevos puntos se requiere interpolar.**
- **La interpolación puede considerarse como el cálculo del valor de intensidad de un pixel, en una posición cualquiera, como una función de los pixeles que le rodean.**
- **Una forma de hacerlo es suponer que el pixel toma el mismo valor que el mas cercano de entre los cuatro que le rodean.**
- **Para decidir cual es el mas cercano se puede utilizar la distancia Euclídea.**

Interpolación Bilineal

- **Una forma de interpolar con mejores resultados pero con mayor costo computacional es la interpolación bilineal, la cual asigna al pixel en cuestión un valor medio ponderado de las intensidades de los 4 pixeles que le rodean.**
- **Los factores de ponderación vienen dados por la distancia entre el pixel y los del entorno.**

Interpolación Bilineal



Interpolación Bilineal

- Los factores de ponderación se calculan de la manera siguiente:

$$a_1 = \left(1 - \frac{dx}{\Delta x}\right) \left(1 - \frac{dy}{\Delta y}\right) \quad a_2 = \frac{dx}{\Delta x} \left(1 - \frac{dy}{\Delta y}\right) \quad a_3 = \left(1 - \frac{dx}{\Delta x}\right) \frac{dy}{\Delta y} \quad a_4 = \left(\frac{dx}{\Delta x}\right) \frac{dy}{\Delta y}$$

donde $0 \leq dx \leq 1, 0 \leq dy \leq 1, \Delta x = 1, \Delta y = 1$

por lo que se obtiene

$$a_1 = (1 - dx)(1 - dy); a_2 = dx(1 - dy); a_3 = (1 - dx)dy; a_4 = dxdy$$

finalmente, el valor del pixel interpolado, en función de los 4 de su entorno queda:

$$p(x, y) = a_1 p(i, j) + a_2 p(i, j + 1) + a_3 p(i + 1, j) + a_4 p(i + 1, j + 1)$$

Fin

Unidad III – Procesamiento Digital de Imágenes