

Proyecto Programación en C

Laboratorio de Paradigmas de Programación

5 de mayo de 2014

1. Implementar las siguientes funciones en el lenguaje C, aplicadas a imágenes PPM.

- a) Función **RGBtoGrises**: Esta función se encarga de convertir una imagen de color a su equivalente en escala de grises, ver la figura 1. Para obtener el valor en escala de grises, se debe obtener el promedio de los valores Red, Green y Blue. A cada píxel en la posición (x, y) aplicar:

$$MGrises[x][y] = \frac{MValRed[x][y] + MValGreen[x][y] + MValBlue[x][y]}{3}$$



Figura 1: Al aplicar la función **RGBtoGrises** a una imagen, se obtiene el valor promedio de los valores RGB y se almacena el valor resultante en la matriz de datos, el valor resultante se observa del lado derecho de la figura.

- b) Función **GrisesAimagenBinaria**: Esta función recibe la imagen en escala de grises y un valor T entre 0 y 255. Los valores por arriba de T se convierten en 0 y los valores menores a T se convierten en 255, ver la figura 2, es decir:

$$MBinaria[x][y] = \begin{cases} 0 & \text{si } MGrises[x][y] \geq T \\ 255 & \text{si } MGrises[x][y] < T \end{cases} \quad (1)$$

- c) Función **NOT**. Esta función lo que hace es invertir los valores de los píxeles cuyo valor sea 255 o 0, es decir, el píxel en la posición (x, y) que tenga un valor de 255 le es asignado el valor 0 en esa posición de la imagen; y cuando se observe en la posición del píxel (x, y) un valor de 0, se debe asignar el valor 255. La figura 3 nos muestra el resultado de aplicar la función **NOT** a una imagen.
- d) Función **AND**: Esta función recibe dos imágenes, una imagen es BINARIA y la otra imagen puede ser RGB, escala de grises o BINARIA. lo que hace la función es una operación AND lógico, es decir, para valores de 0 en el píxel cuya posición es (x, y) en la imagen binaria, se debe asignar un valor de 0 (cero)



Figura 2: Para binarizar una imagen de escala de grises se aplica un valor T, comúnmente conocido como UMBRAL, a cada píxel en la imagen, los valores mayores a T se asignan al tono de gris 0, mientras que valores inferiores a T son asignados al valor de tono de gris 255 (cero).

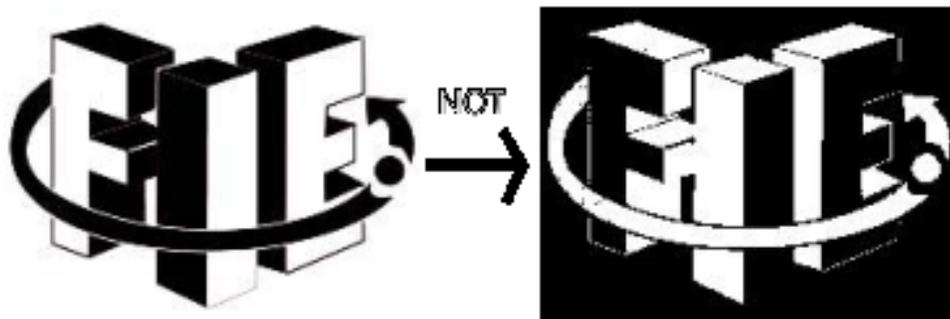


Figura 3: Ejemplo de aplicar la función NOT a una imagen. Del lado izquierdo la imagen de entrada y del lado derecho el resultado de aplicar la función NOT a la imagen original.

en el píxel cuya posición es (x, y) de la imagen resultante, mientras que para los píxeles con valor de 255 se debe asignar el valor o valores del píxel con posición (x, y) de la imagen que no es BINARIA, ver la figura 4.

$$ImagenAND[x][y] = \begin{matrix} 0 & \text{si } MBinaria[x][y] = 0 \\ IMAGEN2[x][y] & \text{si } MBinaria[x][y] = 255 \end{matrix} \quad (2)$$

e) Función **sumar**: Esta función realiza la suma aritmética de los valores existentes en cada píxel.

$$Msuma[x][y] = M1[x][y] + M2[x][y]$$

2. Realizar la siguiente secuencia de ordenes para generar la imagen que se observa en la Figura 5.

- a) Leer 2 IMÁGENES y determinar cual de ellas tiene las menores dimensiones (ancho y alto).
- b) Convertir a tono de grises la imagen de menor tamaño.
- c) Convertir a imagen BINARIA la imagen resultante en del paso 2b MB. El valor de T se debe pasar como parámetro al programa.
- d) Aplicar la función NOT a la imagen MB, MBNOT.
- e) Aplicar la función AND a las imágenes MBNOT y la imagen con las dimensiones mayores, resultando la imagen MBNOTandIMG1.

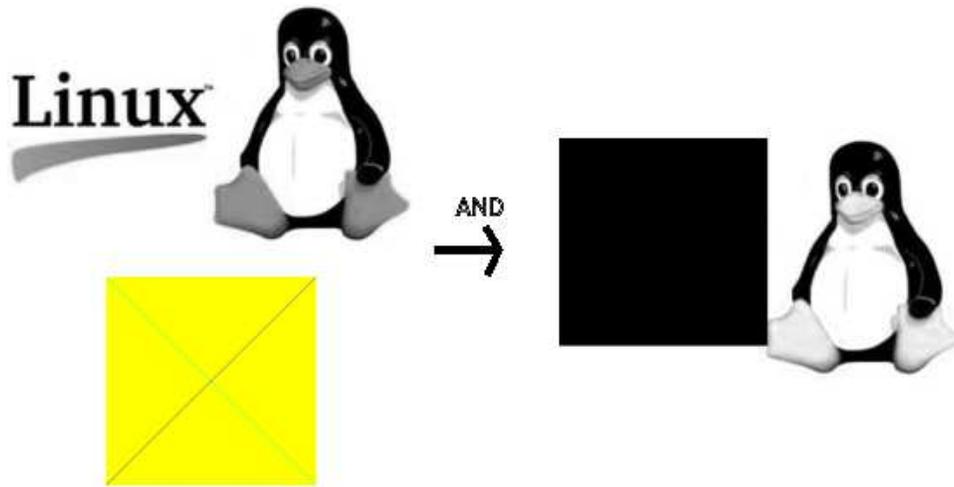


Figura 4: Ejemplo de aplicar la función AND a dos imágenes, de las cuales una es BINARIA.

- f) Aplicar la función AND a las imágenes MB y la imagen de menor tamaño, resulta la imagen MBandIMG2.
- g) Aplicar la función sumar a las imágenes MBandIMG2 y MBNOTandIMG1.
- h) Guardar la imagen a un archivo, el nombre del archivo se recibe como parámetro de entrada al programa.

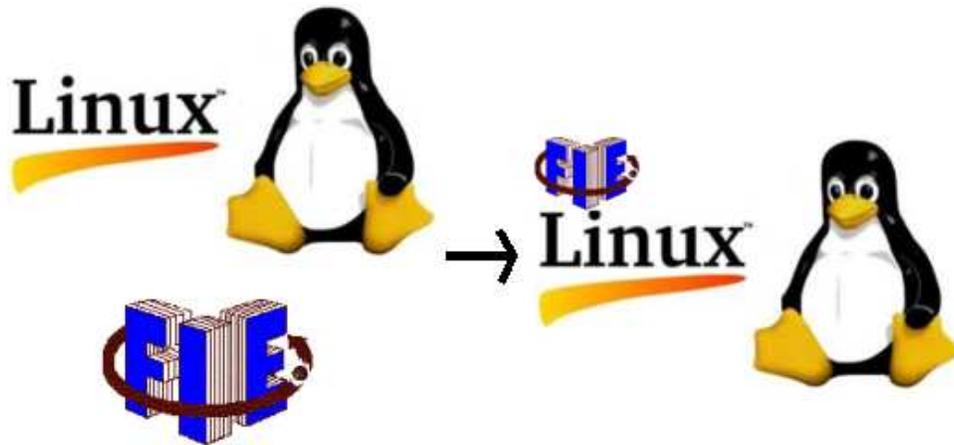


Figura 5: Imagen resultante de aplicar el procedimiento descrito anteriormente. La imágenes de entrada son las del lado izquierdo y la imagen resultante es la del lado derecho. Está es una técnica para sobrescribir regiones de una imagen que no son rectangulares dentro de otra imagen.