

Programación Aplicada y Lab.

PRÁCTICA No. 1

Prof. Maestro Jorge Rodríguez

I. DESCRIPCIÓN DEL SISTEMA

SE desea un programa que realice procesamiento digital de imágenes en formato PGM (por sus siglas en inglés *Pixel Gray Map*). El programa deberá de constar de cuatro partes: 1. Negativo de una imagen PGM. 2. Ecualización lineal de una imagen PGM. 3. Reducción de una imagen PGM y 4. Ampliación de una imagen PGM.

La práctica deberá constar de un menú el cual pida al usuario el proceso que desea realizar sobre la imagen: obtener su negativo, ecualizarla, reducirla o agrandarla. El programa deberá preguntar el nombre del archivo a modificar (este archivo debe de existir y estar en formato PGM ASCII). Después de la modificación el programa deberá preguntar por el nombre del nuevo archivo en donde almacenará la imagen manipulada.

Las funciones para la escritura y lectura de archivos en formato PGM permitirán leer la imagen en un archivo con dicho formato y cargarla en una matriz donde cada elemento corresponde al valor de luminancia de cada pixel de la imagen. El valor de luminancia representa el nivel de gris, siendo cero el color negro y 255 el blanco, los niveles de grises se encuentran linealmente distribuidos en este rango. La matriz deberá ser dimensionada (por medio de *malloc* o *calloc*) para tener el mismo tamaño que la imagen a procesar.

Negativo de una imagen.

El negativo de una imagen se define como el inverso de la imagen original. En imágenes en tono de grises, el negativo de la imagen se calcula invirtiendo el tono de gris con respecto al máximo y mínimo de valores que los pixeles pueden tomar.

De esta forma un pixel con valor 0, tendrá el valor 255 en su negativo, y un pixel con valor 255 tendrá el valor 0 en su negativo. Todos los tonos de grises intermedios se invertirán de manera lineal:

$$y = x*(-1) + 255$$

Donde x es el pixel original y y es el pixel invertido.

Ecualización de una imagen.

La ecualización lineal consiste en normalizar las variaciones

de las muestras para que ocupen todo el rango posible. En una imagen oscura, el rango de valores que ocupan los pixeles está limitado a valores (tonos de grises) oscuros. De igual manera en una imagen con demasiada luz, el rango de valores será muy alto (sin ocupar valores oscuros).

El objetivo de la ecualización consiste en escalar de manera lineal cada valor para que se abarque todo el rango de valores posibles, en nuestro caso del 0 al 255. La magnitud de escalamiento dependerá de cada imagen.

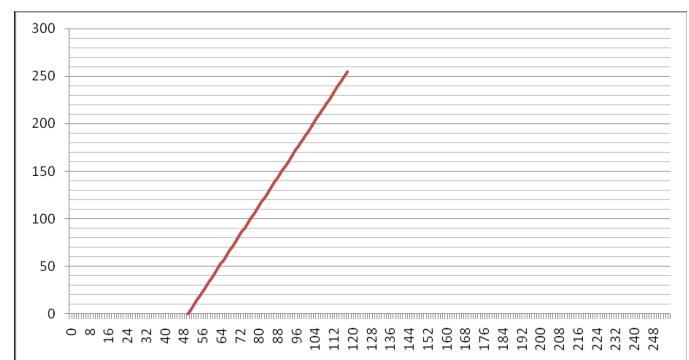


Figura 1. Eje x: pixeles de entrada (imagen original). Eje y: pixeles de salida (imagen ecualizada). Esta gráfica representa una línea recta puesto que el proceso de escalamiento es lineal. La ecuación de la recta se define como Función de Transferencia

Reducir una imagen

Para obtener una imagen reducida a la mitad, tanto horizontal como verticalmente, se deben quitar la mitad de las columnas y renglones de la imagen. De esta manera, al hacerlo de manera intercalada, se obtiene una imagen de la mitad de las dimensiones.

Ampliar una imagen

Para ampliar una imagen al doble, tanto horizontal como verticalmente, se deben insertar el doble de columnas y renglones. Para estas nuevas columnas y renglones, se debe generar la información (pixeles) que deban ocupar este espacio, para lo que se seguirá el método más sencillo que implica la duplicación de la información de las columnas y renglones anteriores a estos.

II. ALCANCES Y LIMITACIONES

El programa deberá de funcionar a través de un menú. El usuario deberá de introducir los datos por medio del teclado.

III. REQUISITOS FUNCIONALES

Al comenzar el programa debe aparecer en pantalla el nombre de la persona que lo elaboró. Se deberá esperar a que se presione un *enter* para continuar. Posteriormente debe desplegarse en pantalla un mensaje que explique brevemente el objetivo del sistema.

IV. RESTRICCIONES DE PROGRAMACIÓN

Se deben considerar los siguientes puntos:

- 1) Para el desarrollo del programa se deberá emplear lo visto en clase, o alguna otra de las funciones estándar de C que sirven para dar mejor presentación al programa, tal como la función `system("clear")`.
- 2) El código del programa debe estar debidamente comentado e indentado.
- 3) La documentación deberá estar de acuerdo al formato del IEEE.

V. FECHA DE ENTREGA

Consultar en el sistema.