PRÁCTICA 1. LIBRERÍAS PARA MANEJAR IMÁGENES.

Descargar de la página web de la asignatura las librerias *ami_bmp.h* y *ami.h* .

Apartado 1.1. Implementar la función

void aan_unir_canales_unsigned_char(unsigned char *canal1, unsigned char *canal2, unsigned char **canal_output, int width, int height)

que toma los canales *canal1* y *canal2*, de tamaño (*width x height*) y genera el canal *canal_output* como unión de los dos canales en uno de tamaño *((width*2+4) x height)*, dejando una franja de 4 píxeles en negro entre ambos. Para comprobar el correcto funcionamiento de la función, aplicar el algoritmo a imágenes reales (en formato bmp), teniendo en cuenta que hay que aplicarlo a cada uno de los canales (rojo, verde y azul) que forman la imagen. Implementar la misma función para imágenes en precisión flotante, es decir:

void aan_unir_canales_float(float *canal1, float *canal2, float **canal_output, int width, int height)

Para leer y escribir las imágenes de/en disco, se utilizarán las funciones:

```
int ami_read_bmp(char name[200], unsigned char **red, unsigned
char **green, unsigned char **blue, int *width, int *height)

int ami_write_bmp(char name[200], unsigned char *red, unsigned
char *green, unsigned char *blue, int width, int height)
```

donde *name* es el nombre del fichero, *red*, *green* y *blue* son los 3 canales de la imagen, y *width* y *height* son las dimensiones.

Apartado 1.2. Implementar la función

void aan_normalizar_canal_unsigned_char(unsigned char *canal_input, unsigned char *canal_output, int width, int height)

que normaliza el vector canal_input entre los valores 0 y 255 y lo vuelca en el canal_output utilizando la fórmula

canal_output[i]=255*(canal_input[i]-min)/(max-min)

donde min y max son el máximo y el mínimo del vector canal_input[].

Utilizar esta función para implementar la función

void aan_normalizar_imagen_unsigned_char(unsigned char *red, unsigned char *green, unsigned char *blue, int width, int height)

que normaliza los valores de los 3 canales de una imagen.

Para comprobar el correcto funcionamiento de las funciones, aplicar el algoritmo de normalización a imágenes reales. Utilizar la función aan_unir_canales_unsigned_char() para crear una imagen que sea la original unida a la imagen después de normalizarla.

Implementar también las correspondientes versiones de estas funciones en precisión flotante.

Apartado 1.3. Extraer, a partir de los canales RGB de una imagen, los correspondientes canales HSV. Mostrar cada uno de esos canales junto con la imagen original. Para mostrar cada canal, al crear la imagen, utilizar el mismo canal repetido tres veces. Utilizar la siguiente conversión:

Sea max el máximo de (R,G,B) y sea min el mínimo de (R,G,B):

V=max

S=max-min

Si el máximo es R:
H=43*(G-B)/(max-min) módulo 256

Si el máximo es G:
H=43*(B-R)/(max-min) + 85

Si el máximo es B:
H=43*(R-G)/(max-min) + 170