

Procesamiento de imágenes

En este trabajo práctico se desarrollarán dos herramientas para procesar imágenes. Para independizarnos del formato, usaremos GraphicsMagick (<http://www.graphicsmagick.org/>) para convertir las imágenes a procesar en archivos rgb. Una imagen rgb es un archivo binario que contiene directamente los valores de los canales rojo, verde y azul de cada pixel usando tres bytes por pixel.

Primera etapa



Implementar una herramienta de línea de comandos para interpolar dos imágenes de iguales dimensiones. Esto es, que combine ambas imágenes usando un valor entre 0 y 1 para indicar en qué proporción afecta al resultado el color de cada imagen. Usaremos interpolación lineal entre el valor de cada canal de cada par de pixels de las dos imágenes.

La interpolación lineal entre dos valores, tiene un parámetro p , entre 0 y 1, que indica qué proporción de cada valor tendrá el resultado: $r = p \times v_1 + (1 - p) \times v_2$.

Notar que si $p = 0.5$ la interpolación lineal es igual al promedio.

1. **Programar en C** las siguientes funciones, para poder leer y escribir archivos rgb:

```
int leer_rgb(char *archivo, unsigned char *buffer, int filas, int columnas);  
int escribir_rgb(char *archivo, unsigned char *buffer, int filas, int columnas);
```

2. **Programar en lenguaje ensamblador**, usando instrucciones **SIMD**:

```
void interpolar(unsigned char *img1, unsigned char *img2, unsigned char *resultado,  
float p, int cantidad);
```

3. **Escriba un programa en C** que use las funciones anteriores para implementar la herramienta de combinación de imágenes, tomando los siguientes parámetros de la línea de comandos:

- img1.rgb
- img2.rgb
- filas
- columnas
- p
- resultado.rgb.

Incluya en el informe imágenes de prueba y el resultado de combinarlas con distintos valores de p .

Fecha de entrega: 8/11

Segunda etapa



Pantalla verde: Implementar una herramienta de línea de comandos que permita tomar una imagen de primer plano y un color que se considerará transparente. Esta imagen se combinará con una imagen de fondo que será visible a través de las partes de la primera imagen que contengan el color transparente.

1. **Implementar en lenguaje C** las siguientes funciones:

- `void separar_rgb(unsigned char *rgb, int cantidad, unsigned char *r, unsigned char *g, unsigned char *b);`
- `void combinar_rgb(unsigned char *r, unsigned char *g, unsigned char *b, int cantidad, unsigned char *rgb);`

`separar_rgb` debe tomar un vector de valores rgb y separarlo en tres vectores independientes, uno para cada canal.

`combinar_rgb` debe tomar los vectores de cada canal y combinarlos en un solo vector donde los valores r, g y b de cada pixel son consecutivos.

2. **Implementar en lenguaje ensamblador**, usando instrucciones **SIMD**, las siguientes funciones:

- `void gen_mascara(unsigned char *r, unsigned char *g, unsigned char *b, int cantidad, int rt, int gt, int bt, float t, unsigned char *mascara);`
- `void aplica_mascara(unsigned char *mascara, unsigned char *canal1, unsigned char *canal2, int cantidad, int unsigned char *resultado);`

La función `gen_mascara` recibirá tres vectores con las componentes rgb de la imagen figura, las componentes rgb del color transparente y la tolerancia t, y generará un vector del mismo tamaño conteniendo el byte 0FFh en las posiciones que correspondan a pixels transparentes y 0 en las que no.

Para determinar si un pixel es transparente o no, se calculará la distancia entre el color del pixel y el color transparente usando la siguiente fórmula: $d = \sqrt{(r_1 - r)^2 + (g_1 - g)^2 + (b_1 - b)^2}$.

r_1 , g_1 y b_1 son los componentes rojo, verde y azul de un pixel de la imagen 1, y r , g y b son los componentes del color transparente.

Si para un pixel cualquiera $d \leq t$, entonces el pixel se considerará transparente.

La función `aplica_mascara` será la encargada de combinar un canal de la imagen figura con un canal de la imagen fondo usando la máscara para seleccionar qué pixel va en el resultado. Esta función se llamará tres veces para generar los tres canales de la imagen resultado.

3. **Escriba un programa en C** que use las funciones anteriores para implementar la herramienta de "pantalla verde", tomando los siguientes parámetros de la línea de comandos:

- `img1.rgb`
- `img2.rgb`
- `filas`
- `columnas`
- `r`
- `g`
- `b`
- `t`
- `resultado.rgb`.

El ejecutable debe recibir como parámetros las dos imágenes, el alto y ancho, las componentes rgb del color transparente, el umbral del color y el nombre del resultado.

Incluya en el informe imágenes de prueba y el resultado de combinarlas con distintos valores de t y distinta selección de color transparente.

Fecha de entrega: 22/11

Entregas

Cada entrega debe ser por mail a la dirección choltman@ungs.edu.ar. Debe incluir:

- Nombre y Legajo de los dos integrantes del grupo.
- El código fuente completo.
- Un archivo bat o make para compilar los fuentes. Los ejecutables generados deben cumplir con la especificación del enunciado.
- **Un informe que describa el desarrollo del proyecto, fuentes y pruebas realizadas demostrando el funcionamiento del programa.**