

Informática Gráfica
Ingeniería en Informática
Curso 13-14. Práctica 4.1

Carácter: obligatorio.

Fecha de entrega:

- ✓ Parte obligatoria: martes 18 de febrero a las 23:55 (la revisión será el miércoles 19 en el laboratorio).
- ✓ Parte opcional: martes 25 de febrero a las 23:55 (la revisión será el miércoles 26 en el laboratorio).

Objetivo: procesamiento de imágenes rasterizadas.

Descripción: se trata de programar algunas de las operaciones vistas en clase para transformar matrices de píxeles (*pixmaps*). Las imágenes de partida se podrán conseguir de dos formas:

- Leyendo un fichero en formato BMP (puedes usar el cargador disponible en el campus, programar el tuyo propio, o buscar alguno en internet),
- Renderizando el árbol de Pitágoras que diseñaste en la Práctica 1, y leyendo la matriz de píxeles desde el frame buffer.

Tras la inicialización de la imagen, ésta se visualizará en el puerto de vista, y pasará a considerarse el pixmap *activo*. A continuación el usuario podrá solicitar alguna de las siguientes operaciones para modificar este pixmap, y visualizar después el resultado obtenido:

1. Rotar la imagen un determinado ángulo, usando como centro de rotación el píxel que elija el usuario con el ratón. El ángulo se leerá por el teclado y podrá no ser múltiplo de 90°. Para implementar esta transformación geométrica usarás interpolación bilineal como se explicó en clase.
2. Aplicar sobre la imagen algunas de las operaciones aritméticas vistas en clase. Siguiendo nuestra terminología, se piden la media ponderada y la diferencia. Observa que estas operaciones requieren, además del pixmap activo, una segunda imagen con la que operar. Dicha imagen secundaria debe cargarse desde el BMP que elija el usuario.
3. Aplicar sobre el pixmap activo un suavizado Gaussiano (los parámetros σ y m se dejan a tu elección).

En cuanto al modelo de color de tus pixmaps, usarás RGB en la inicialización (desde BMP o desde el árbol de Pitágoras). Todas las operaciones que se piden trabajarán en este modelo, salvo la diferencia de imágenes que usa escala de grises. Por ello, para esta operación debes comenzar transformando las imágenes desde el modelo RGB a la banda Y del modelo YIQ, y trabajar en esa única banda después. Para visualizar una imagen en escala de grises en el frame buffer, puedes: (1) enviar los valores de gris al buffer a través de `glDrawPixels`, pero usando el parámetro `GL_LUMINANCE` en lugar de `GL_RGB`; o (2) convertir la escala de grises en un pixmap RGB sin más que replicar el valor de gris en cada una de las tres bandas.

Estructura de la información

La clase principal en esta práctica es la clase Pixmap que debe guardar la matriz de píxeles, y otros datos de interés para la imagen como por ejemplo su tamaño o el número de bandas que tiene. Además debe ofrecer la siguiente funcionalidad:

- ✓ Inicializar desde BMP, o desde el frame buffer.
- ✓ Visualizar la matriz de píxeles en la ventana.
- ✓ Rotar el pixmap. Interesa implementar la interpolación lineal con un método independiente.
- ✓ Computar la media ponderada, a partir de un factor k y otro pixmap.
- ✓ Construir la diferencia, usando como referencia otro pixmap.
- ✓ Aplicar el suavizado Gaussiano. Interesa implementar un método que aplique una máscara dada sobre la imagen.

Partes opcionales

1. [+] Permite que tu práctica salve el pixmap activo en un fichero BMP, para poder recuperarlo posteriormente.
2. [+] Aplica sobre el pixmap activo los operadores de Sobel para la extracción de bordes. Recuerda que esta operación trabaja sobre escala de grises.
3. [+] Permite que la matriz binaria obtenida en el apartado anterior se visualice sobre la imagen original. El objetivo es exagerar en la imagen original los bordes encontrados, por ejemplo mostrando los píxeles correspondientes en color blanco. Para conseguirlo, puedes usar el canal *alpha* de la matriz de bordes, y aplicar el *blending* de OpenGL sobre la imagen original.