

Transformaciones geométricas

1. Descripción

En esta tarea se programará una aplicación sencilla utilizando una clase que sigue el patrón de diseño denominado *functor*, para aplicar transformaciones geométricas de imágenes. La figura 1 muestra un ejemplo de transformación utilizando un mapeo bilineal

$$w = \frac{az + b}{cz + d} \quad (1)$$

$$a = 2,1 + j2,1$$

$$b = 0$$

$$c = 0,003$$

$$d = 1 + j$$



Figura 1: Imagen 'lenna' y su transformación con un mapeo bilineal

2. Objetivos

Al finalizar la tarea el estudiante podrá:

1. describir el concepto de patrón de diseño, en particular el functor
2. transformar una imagen en sus características geométricas
3. utilizar interpoladores para generar información entre píxeles.
4. utilizar polimorfismo estático, via programación genérica,

3. Metodología

Para realizar esta tarea usted debe

1. Busque qué es un patrón de diseño en el contexto de programación orientada a objetos.
2. Investigue qué características tiene el patrón de diseño denominado “functor”.
3. Utilice el código brindado en C++ para completar la clase que implementa el mapeo complejo bilineal. Dicho ejemplo ya está listo para ser completado únicamente con la transformación bilineal.
4. Complete los métodos de la clase en el archivo `ltiBilinearMapping_template.h` marcadas con `TODO`. Estos métodos incluyen el mapeo bilineal y su inverso, así como un método `apply` que realiza la conversión de las imágenes.
5. Para la programación del método que realiza la transformación geométrica asegúrese de
 - a) encontrar primero el mapeo de las cuatro esquinas
 - b) determinar las coordenadas del rectángulo que enmarca el mapeo de las cuatro esquinas
 - c) dimensionar el canal de salida para que cubra el rectángulo determinado, pero verificando que dicho tamaño no exceda los 2048 píxeles en ninguna dimensión.
 - d) recorrer todos los píxeles del canal de salida, y determinar por medio de la función de mapeo inverso cuál punto corresponde en la imagen de entrada.
 - e) utilizar un interpolador para encontrar el valor del canal de entrada correspondiente al pixel. Utilice el interpolador de la clase madre, al que puede acceder con `this->interpolator_` desde el método `apply` correspondiente.
Note que el método `apply` está declarado como `const`, lo que quiere decir que solo puede emplear métodos `const` del interpolador.
6. Utilice el programa principal entregado para demostrar la transformación bilineal.
Note que dicho programa utiliza un archivo `bilinear.dat` para obtener los valores de los parámetros del mapeo, archivo que usted puede editar para comprobar el funcionamiento del programa.

4. Entregables

1. Los archivos fuente.
2. Fecha de entrega: Jueves 01 de Noviembre