

Taller 3

Evaluación experimental de algoritmos

Ejercicios a desarrollar

Una operación frecuente en procesamiento de imágenes es la detección de bordes (edge-detection). Esta operación es intensiva computacionalmente, por lo que resulta interesante poder evaluar su eficiencia, particularmente en aplicaciones que requieren el análisis de muchas imágenes.

La biblioteca del texto guía incluye el ADT [Picture](#), el cual implementa funciones básicas para leer/escribir imágenes desde el disco y para leer/modificar valores de los puntos que componen la imagen.

a). Implementar el algoritmo `edgeDetection` como una función de biblioteca. `edgeDetection` toma como entrada una imagen (una instancia de `Picture`) y devuelve la imagen resultante de aplicar el algoritmo de detección de bordes con un determinado umbral:

```
Picture edgeDetection(Picture entrada, double umbral)
```

(Ver referencias al final sobre el algoritmo de detección de bordes).

b). Hacer una función de biblioteca que lee el nombre/URL de una imagen, lea la imagen, obtenga sus bordes y visualice la imagen resultante.

c). Para estimar experimentalmente la eficiencia del algoritmo `edgeDetection`, implementar una función de biblioteca:

```
void performanceTest(String[] imagenes)
```

la cual toma como argumento un arreglo de Strings con los nombres de archivo/URLs de las imágenes con las que se van a hacer las pruebas. Para cada imagen calcula sus bordes 5 veces y obtiene el tiempo promedio de los 5 ejecuciones (hacer uso de [Stopwatch](#)). Al terminar, imprime el nombre la imagen, su tamaño, el tiempo promedio para calcular sus bordes.

d) Los datos obtenidos de la evaluación del algoritmo se tabulan en una hoja de calculo: Tamaños de imagen y tiempos de ejecución.

e) Hacer la gráfica X-Y de los datos obtenidos y obtener la curva de mejor ajuste para el desempeño del algoritmo utilizando los métodos de interpolación disponibles en la hoja

de cálculo.

f) Obtener la expresión de la curva de mejor ajuste.

g) A partir del código del algoritmo, estimar su orden de crecimiento utilizando el método analítico. Comparar con el resultado experimental. Incluir el análisis como un texto en la misma hoja de cálculo.

Referencias

[Simple Edge Detection](#)

[Canny Edge Detector Implementation](#)

Entregables

Remitir el código fuente de la solución implementada y la hoja de cálculo con los resultados de las pruebas (se aceptan Word, Excel, OpenOffice, LibreOffice, PDF). Nombrar el archivo comprimido Taller3-<Nombre1>-<Nombre2>... (.zip .rar .7z o .tgz). Para estandarizar la forma de invocar el programa, ubicar el método main y las funciones de biblioteca solicitadas en la clase Taller3.

En caso de utilizar estructuras de las bibliotecas del texto (algs4.jar) **no** anexar la biblioteca.

Grupos máximo de 2 personas.