Sergi Bons Fuses
NIU:1571359

Objetivos:

El algoritmo filtra una imagen (para pruebas de optimización, esta se trata de una pseudo imagen generada con con una seed) con dos filtros distintos, primero un filtro de media, que a función práctica emborrona la imagen, y después con un filtro detector de bordes.

Esta concatenación de filtros permite una detección de bordes, que ignora en cierta medida el ruido o textura de la imagen, es decir, solo encuentra los bordes más marcados.

El programa, primero inicializa la imagen pseudo aleatoria, (objetivo específico 1)
posteriormente filtra la imagen con el primer filtro, (objetivo específico 2)
después filtra el resultado con el segundo filtro. (objetivo específico 3)

Finalmente muestra por pantalla una selección de puntos para comprobar que estos sean correctos.

Sergi Bons Fuses
NIU:1571359

Algoritmo:

El algoritmo consta de una imagen original, que es un conjunto de tres matrices de dimensiones ampliables, así como de dos filtros, que son matrices de 3x3 predefinidas.

Para ejecutar los filtros, el algoritmo usa 4 bucles anidados(figura 1),

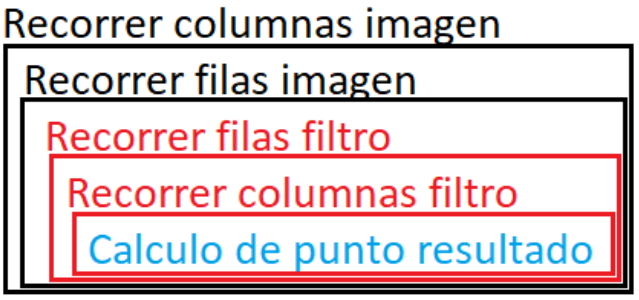

Figura 1

de manera que cada punto de la matriz resultado, se calcula usando los puntos vecinos de la matriz imagen, multiplicados por los puntos correspondientes del filtro(figura 2). efectuando una convolución.

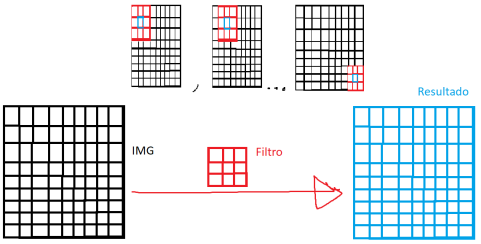

Figura 2

Cabe mencionar que en los bordes de la imagen, el cálculo se efectúa teniendo en cuenta los puntos más alejados al centro del cálculo, como se ve en la figura 3.

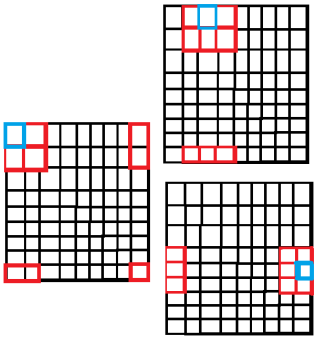

Figura 3

Sergi Bons Fuses
NIU:1571359

Funcionamiento computacional:

Si tenemos que:
W = Ancho imagen original
H = Alto imagen original
Fsz = Lado del filtro.

La complejidad de nuestro problema es de:
W*H + (W*H*(Fsz^2) *2)
Como sabemos los tamaños del filtro, tenemos que:
W*H + W*H*18 = 19WH
Y si asumimos que la imagen es cuadrada, y con dimensión de lado N, tenemos una complejidad algorítmica:

19N^2

Respecto a la complejidad espacial, tenemos que usamos dos matrices de H*W y dos filtros de 3x3, todo ints, así que: (H*W*2 + 16)*4

A lo que localidad espacial y localidad temporal se refiere, tenemos una alta localidad espacial, puesto a que accedemos a elementos contiguos de la matriz, así como al filtro, de manera muy reincidente. La localidad temporal es menor, ya que el único elemento al que se acude de manera reincidente es al filtro, y este también cambia a mitad de la ejecución. El cálculo de cada punto de la imagen sigue un patrón de cómputo REDUCE, ya que usa matrices de 3x3 para calcular cada punto, aunque, el cálculo final de la matriz, sigue un patrón MAP, ya que cada punto no depende de los otros puntos resultado.

Tamaños a evaluar y valores de entrada:

Para este problema se evaluarán tamaños de matrices imagen de 1Kx1K, 10Kx10K y 10Mx10M, y los datos de entrada son generados en el mismo programa, como simulación de la lectura de una imagen.