

Algoritmos paralelos para la reducción de ruido mixto gaussiano-impulsivo en imágenes médicas

Filtrado de imágenes en tiempo real acelerado con C++, OpenMP y MPI

Ignacio Encinas Rubio
Tutor: Josep Arnal García

Escuela Politécnica
Universidad de Alicante

13 de junio de 2022

Presentación disponible en
ls.ecomaikgolf.com/tfg/ignacio/slides.pdf



1. Introducción

- 1.1. Contexto
- 1.2. Punto de partida
- 1.3. Ruidos a filtrar

2. Marco Teórico

- 2.1. Algoritmo

3. Implementación

- 3.1. Versión secuencial
- 3.2. Versión en memoria compartida
- 3.3. Versión distribuida

4. Rendimiento

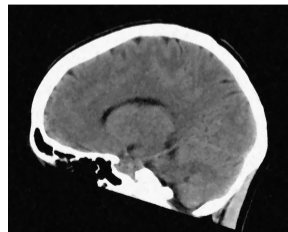
- 4.1. Equipos de pruebas
- 4.2. Versión en memoria compartida
- 4.3. Versión distribuida
 - Esquema de iteraciones locales

5. Calidad de filtrado

- 5.1. Métricas numéricas
- 5.2. Calidad visual

Las imágenes CT son una herramienta fundamental para diagnosticar numerosas patologías.

- Contaminadas en el proceso de adquisición, transmisión y almacenamiento
- Reducción en la radiación nociva para el paciente causa un aumento en el ruido
- Métodos de filtrado efectivos costosos computacionalmente

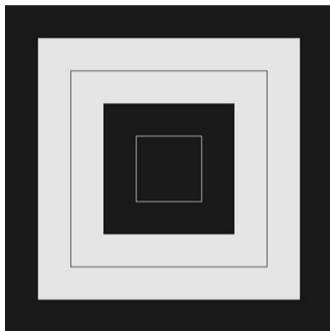




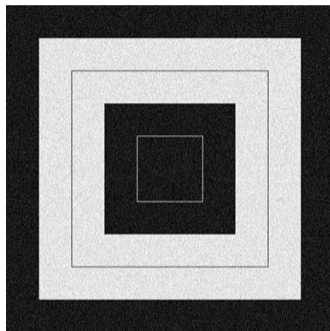
Implementación del filtro basado en la lógica difusa propuesto en **Camarena, J.-G., Gregori, V., Morillas, S., y Sapena, A. (2013). A simple fuzzy method to remove mixed gaussian-impulsive noise from color images. IEEE Transactions on Fuzzy Systems, 21(5), 971-978.**

Ejecutado de manera iterativa. Evaluación de un esquema de iteraciones locales en la implementación distribuida.

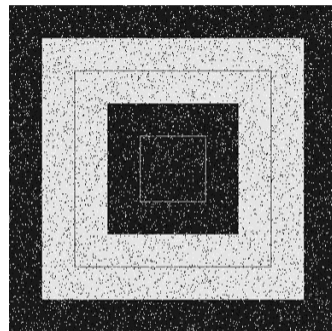
- Versión secuencial
- Versión en memoria compartida
- Versión distribuida



(a) Original



(b) Ruido gaussiano



(c) Ruido impulsivo

Figura: Ruidos a filtrar y sus efectos visuales

Primero, un poco de notación:

- x_i se refiere al pixel que está siendo filtrado
- x^j se refiere a cualquier pixel dentro de la ventana de filtrado

Trabajaremos con los q vecinos más similares a x_i

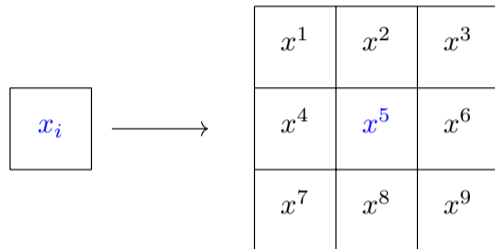
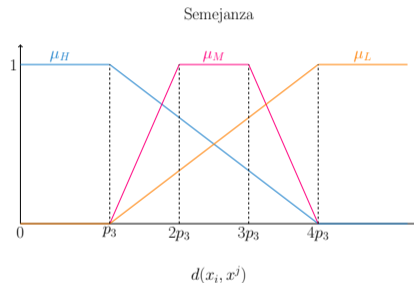
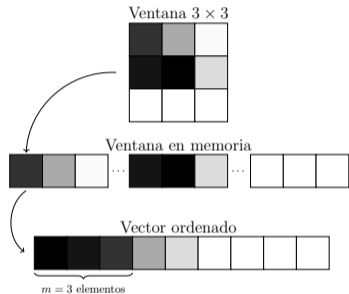


Figura: Ventana de filtrado. Ejemplo de dimensiones 3x3

Caracterización de los ruidos gaussianos e impulsivos.



Grado de impulsividad determinado por la

$$\text{métrica ROAD}_m = \sum_{j=1}^m d(x_i, x^j)$$

Tres grados de semejanza: alta, media y baja. Dependiente de la distancia entre píxeles $d(x_i, x^j)$.

- Reglas difusas
 1. ...
 2. **SI** (x^j no es impulsivo **Y** x_i es impulsivo **Y** la semejanza entre x^j y x_i es moderada) **ENTONCES** ω_j es un peso moderado.
 3. ...
- Defuzzificación mediante el centro de gravedad

Algoritmo 1: Filtro difuso secuencial

Datos: Imagen ruidosa I , parámetros

 $n, q, m, p_1, p_2, p_3, p_4$
Resultado: Imagen filtrada I'

 Imagen $I_0 = I$
para Iteración $it = 1, \dots$ **hacer**

 Imagen $I_{it} = I_{it-1}$
para x_i pixel $\in I_{it}$ **hacer**

 Tomar la ventana W $n \times n$ centrada en x_i
Cálculo del grado de impulsividad

 Calcular $\mu(x_i)$
Cálculo del grado de semejanza

 Ordenar los píxeles $x^j \in W$ según $d(x_i, x^j)$

 Seleccionar los q píxeles mas cercanos

 x^1, \dots, x^q
para $j = 1, \dots, q$ **hacer**

Calcular

 $\mu_H(x_i, x^j), \mu_L(x_i, x^j), \mu_H(x_i, x^j),$
fin
Cálculo de los pesos mediante defuzzificación
para $j = 1, \dots, q$ **hacer**

 Calcular las reglas difusas para $\{x_i, x^j\}$

 Calcular el peso ω_j correspondiente a x^j

mediante COG

fin
Cálculo del nuevo valor para x_i

$$\hat{x}_i = \frac{\sum_{j=1}^q \omega_j \cdot x^j}{\sum_{j=1}^q \omega_j}$$

fin
fin

Cálculo del centro de gravedad mediante métodos geométricos.

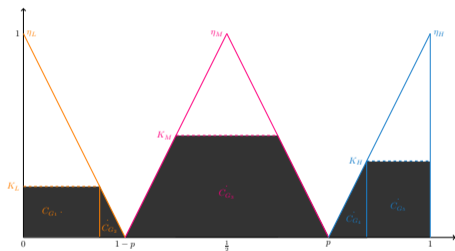


Figura: Centro de gravedad a obtener por cada $x_i, x_j \in W$

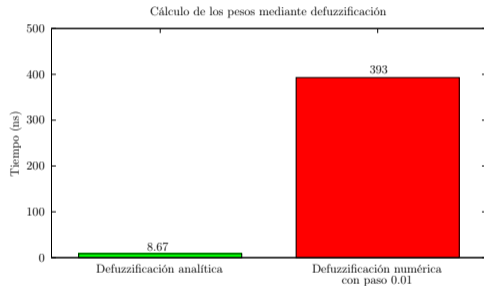
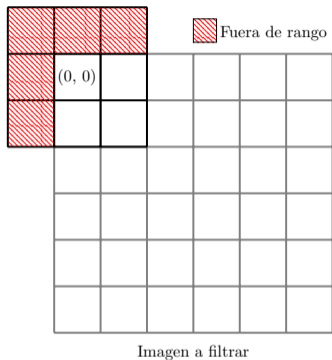
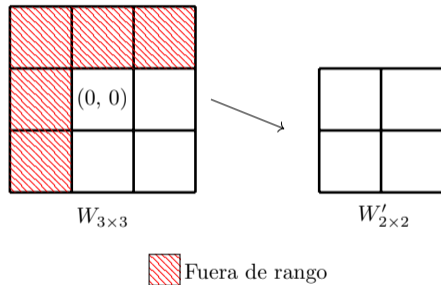


Figura: Aceleración obtenida gracias al cálculo geométrico



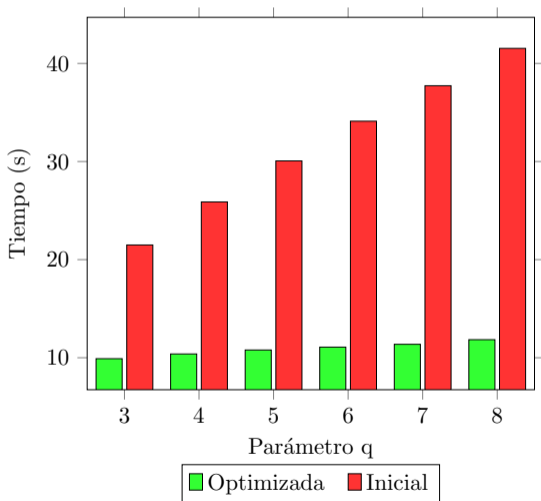
(a) Ventanas que sobrepasan los límites de la imagen a filtrar



(b) Ajuste automático. $W \rightarrow W'$

Figura: Gestión de las ventanas de filtrado transparente al desarrollador

La implementación inicial ingenua podía llegar a calcular el grado de impulsividad de cada pixel en numerosas ocasiones. Por ello introdujimos un «almacén» de grados de impulsividad para evitar este problema.



Paralelización del núcleo de la aplicación mediante directivas de **OpenMP**.

- Paralelización del tratamiento de la imagen.
- Almacenamiento local para cada hilo (TLS)
- Sincronización sin cerrojos

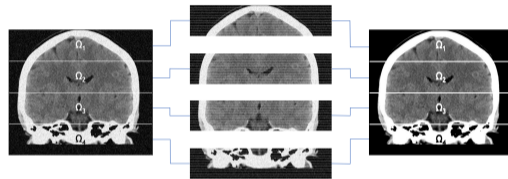


Figura: La paralelización actúa a nivel de subdominio Ω_i

Desarrollada utilizando MPI.

Tamaño de ventana $n = 2\omega + 1$

- Ω_i : Región a **filtrar** por el nodo i
- Ω_i^ω : Región a **necesaria** para el nodo i
- Solapamiento \rightarrow **Comunicación**

Propuesta

Esquema de iteraciones locales. En lugar de comunicar cada iteración, hacerlo cada x iteraciones.

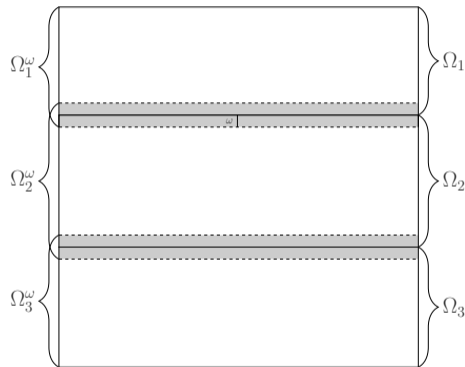


Figura: Descomposición del trabajo para 3 nodos

Equipo 1. Ordenador personal

- (x1) AMD Ryzen 2700X (8C / 16T), 16GB RAM DDR4 3200MT/s
- ArchLinux. Kernel 5.16.12
- GCC 11.2.0

Equipo 2. Cluster del IUII

- (2x) CPU Intel Xeon X 5660 (6C/12T), 48GB RAM DDR3 1333MT/s
- CentOS 7. Kernel 3.10.0
- GCC 7.5.0
- OpenMPI 4.0.2

Rendimiento **óptimo** en el Equipo 1.

Rendimiento **subóptimo** en el Equipo 2.

- **Nodos dual-socket**
- **Compilador más anticuado**

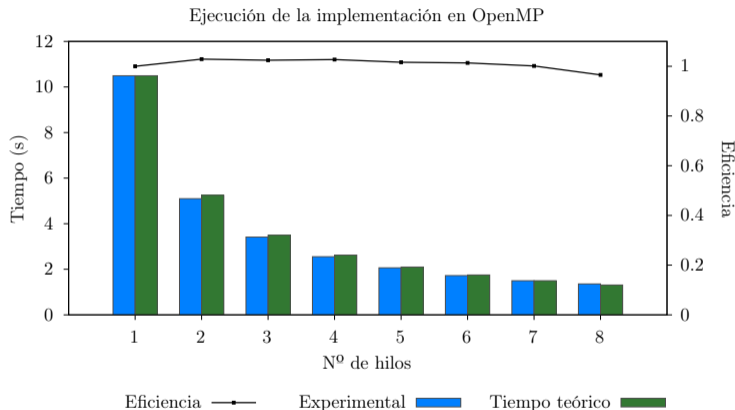
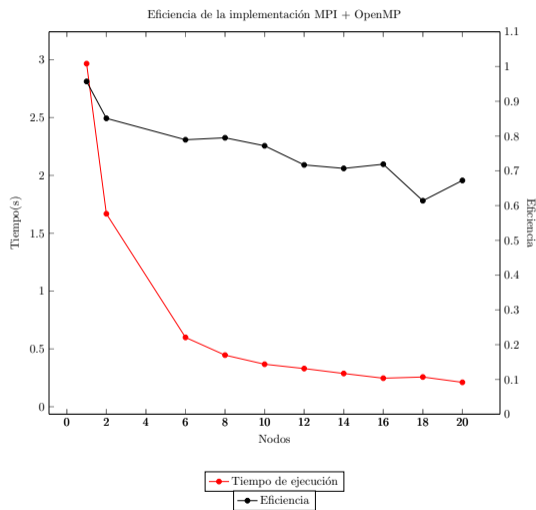
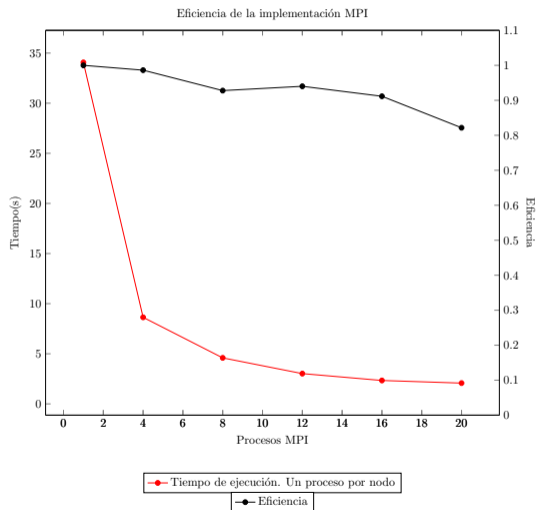
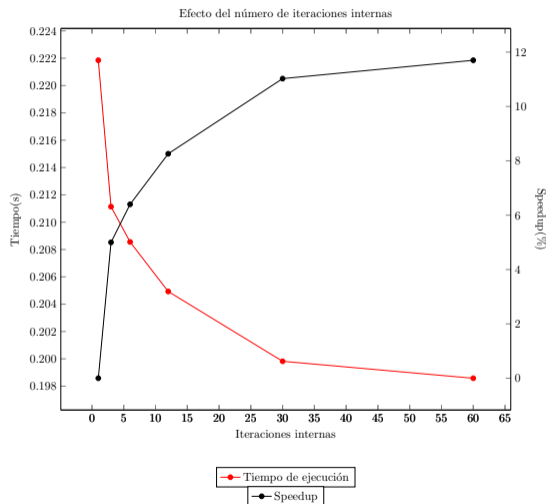


Figura: Benchmark medido en el Equipo 1



- La eficiencia combinada se ve lastrada por el rendimiento paralelo en los equipos del clúster.
- El esquema de iteraciones locales supone una aceleración cercana al **12%** sin pérdida de calidad de filtrado.

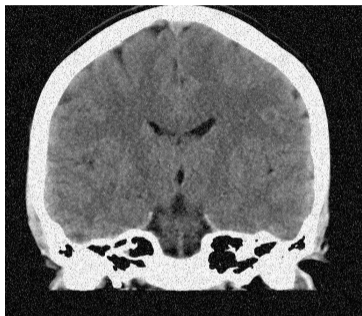


Ruido	MAE					
	Vista axial		Vista sagital		Vista coronal	
	Ruidosa	Filtrada	Ruidosa	Filtrada	Ruidosa	Filtrada
$\sigma = 5, p = 0,05$	13.19	4.32	8.54	2.60	9.20	2.98
$\sigma = 10, p = 0,1$	21.29	5.97	14.28	3.61	15.37	4.07
$\sigma = 20, p = 0,2$	35.65	8.26	24.87	5.91	26.64	6.47
$\sigma = 30, p = 0,3$	50.03	13.02	34.84	9.08	37.29	9.81

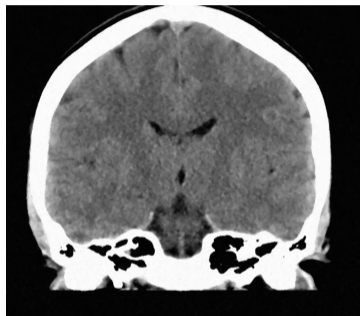
Tabla: Valores mínimos de MAE en las imágenes contaminadas con los distintos ruidos gaussiano e impulsivos

Ruido	PSNR					
	Vista axial		Vista sagital		Vista coronal	
	Ruidosa	Filtrada	Ruidosa	Filtrada	Ruidosa	Filtrada
$\sigma = 5, p = 0,05$	19.03	34.57	18.85	34.81	19.21	34.32
$\sigma = 10, p = 0,1$	16.06	32.22	15.86	32.53	16.13	32.03
$\sigma = 20, p = 0,2$	12.98	29.53	12.82	28.80	13.07	28.46
$\sigma = 30, p = 0,3$	11.17	25.80	11.02	25.24	11.25	25.06

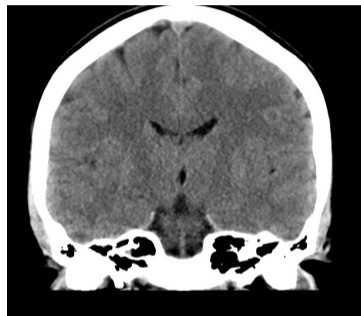
Tabla: Valores máximos de PSNR en las imágenes contaminadas con los distintos ruidos gaussiano e impulsivos



(a) Contaminada $\sigma = 10, \rho = 0,1$



(b) Filtrada



(c) Original

Figura: Vista coronal contaminada, filtrada y original

Resultados:

- Alta eficiencia de la implementación paralela
 - > 85 % Equipo 1
 - > 96 % Equipo 2
- Alta eficiencia de la implementación distribuida. > 82 % con 20 nodos.
- Calidad de filtrado notable.

Líneas futuras:

- Implementación con primitivas MPI no bloqueantes.
- Desarrollar una versión acelerada por GPU.

Gracias por su atención.



Escuela Politécnica Superior

Algoritmos paralelos para la reducción de ruido mixto gaussiano-impulsivo en imágenes médicas

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Ignacio Encinas Rubio

Tutor:
Josep Arnal Garcia

Universitat d'Alacant
Universidad de Alicante

- Valores lógicos $\{0, 1\} \rightarrow [0, 1]$
- Pertenencia a conjuntos definida por funciones
- Operadores lógicos
 - $\{A \wedge B\} \rightarrow \{A \cdot B\}$
 - $\{A \vee B\} \rightarrow \{A + B - A \cdot B\}$
- Reglas difusas
- Defuzzificación

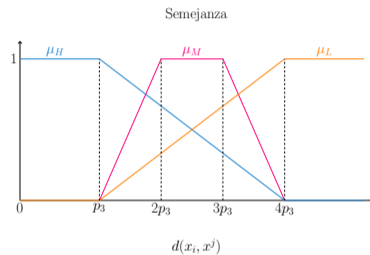


Figura: Grado de semejanza para un pixel x^j en función de $d(x_i, x^j)$

- Caracterización de la impulsividad y la semejanza entre píxeles dentro de la ventana de filtrado.
- Reglas difusas
 1. ...
 2. **SI** (x^j no es impulsivo **Y** x_i es impulsivo **Y** la semejanza entre x^j y x_i es moderada)
ENTONCES ω_j es un peso **moderado**.
 3. ...
- Defuzzificación mediante el centro de gravedad

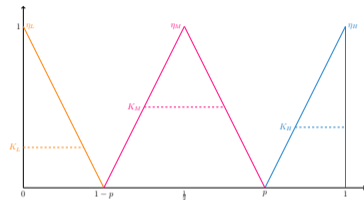


Figura: Funciones de membresía $\eta_H(\omega_j)$, $\eta_M(\omega_j)$, y $\eta_L(\omega_j)$ junto a K_L, K_M, K_H correspondientes a las reglas difusas {1, 2, 3} respectivamente para determinados píxeles x_i y x^j