

Algoritmos paralelos para la reducción de ruido speckle en imágenes médicas

Paralelización del método SRAD haciendo uso de OpenMP y MPI

Autor: Adrián Jiménez González¹

Tutor: Josep Arnal García¹

¹ Escuela Politécnica Superior, Universidad de Alicante

13 de junio de 2022



Tabla de Contenidos

1 Introducción

- Contexto
- Speckle
- Punto inicial

2 Desarrollo

- Algoritmo secuencial
- Versión MPI
- Versión OpenMP
- Versión híbrida

3 Experimentos

- Máquinas
- Imágenes
- Métricas

4 Resultados

5 Conclusiones

6 Líneas futuras

Paralelización
SRAD

Adrián
Jiménez

Introducción

- Contexto
- Speckle
- Punto inicial

Desarrollo

- Algoritmo secuencial
- Versión MPI
- Versión OpenMP
- Versión híbrida

Experimentos

- Máquinas
- Imágenes
- Métricas

Resultados

Conclusiones

Líneas futuras

Las imágenes de Ultra Sonidos (US) son un tipo de imágenes médicas que tienen ruido *speckle* implícito.

- Naturaleza multiplicativa.

$$i(s) = o(s)n(s)$$

- Deteriora la calidad de la imagen.
- Puede eliminar características importantes.

Ruido speckle



Figura: Ruido speckle

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto

Speckle

Punto inicial

Desarrollo

Algoritmo secuencial

Versión MPI

Versión OpenMP

Versión híbrida

Experimentos

Máquinas

Imágenes

Métricas

Resultados

Conclusiones

Líneas futuras

Punto inicial

Implementación del filtro Speckle Reduction Anisotropic Diffusion (SRAD) del artículo **Y. Yu and S. T. Acton, “Speckle reducing anisotropic diffusion,” IEEE Transactions on Image Processing, vol. 11, no. 11, pp. 1260–1270, 2002.**

Se trata de un filtro iterativo de alto coste computacional.

- Versión secuencial.
- Versión MPI.
- Versión OpenMP.
- Versión híbrida (MPI+OpenMP).

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto

Speckle

Punto inicial

Desarrollo

Algoritmo secuencial

Versión MPI

Versión OpenMP

Versión híbrida

Experimentos

Máquinas

Imágenes

Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial

Para filtrar la imagen I es necesario una serie de modificaciones en I . Para ello se añade un borde con valor 0 alrededor de I .

0	0	0	0	0	0	0	0
0	X	X	X	X	X	X	0
0	X	X	X	X	X	X	0
0	X	X	X	X	X	X	0
0	X	X	X	X	X	X	0
0	X	X	X	X	X	X	0
0	X	X	X	X	X	X	0
0	0	0	0	0	0	0	0

Figura: Acolchado de I .

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial

Dentro del proceso iterativo se calcula q_0 a partir de:

$$q_0(t) = \frac{\sqrt{\text{var}[z(t)]}}{z(t)}$$

Siendo z la ROI seleccionada.

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial



Discretizamos la Partial Differential Equation (PDE) de segundo grado utilizando una aproximación de tres pasos siguiendo el método iterativo de Jacobi.

$$\begin{cases} \partial I(x, y; t) / \partial t = \operatorname{div}[c(q) \nabla I(x, y; t)] \\ I(x, y; 0) = I_0(x, y), (\partial I(x, y; t) / \partial \vec{n})|_{\partial \Omega} = 0 \end{cases}$$

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial



En primer lugar, calculamos las aproximaciones de las derivadas y la aproximación del laplaciano como:

$$\nabla_R I_{i,j}^n = \left[\frac{I_{i+1,j}^n - I_{i,j}^n}{h}, \frac{I_{i,j+1}^n - I_{i,j}^n}{h} \right]$$

$$\nabla_L I_{i,j}^n = \left[\frac{I_{i,j}^n - I_{i-1,j}^n}{h}, \frac{I_{i,j}^n - I_{i,j-1}^n}{h} \right]$$

$$\nabla^2 = \frac{I_{i+1,j}^n + I_{i-1,j}^n + I_{i,j+1}^n + I_{i,j-1}^n - 4I_{i,j}^n}{h^2}$$

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial

El segundo consiste en calcular el coeficiente de difusión ($c(q)$) haciendo uso de

$$c_{i,j}^n = c \left[q \left(\frac{1}{I_{i,j}^n} \sqrt{|\nabla_R I_{i,j}^n|^2 + |\nabla_L I_{i,j}^n|^2}, \frac{1}{I_{i,j}^n \nabla^2} I_{i,j}^n \right) \right]$$

En el tercer paso se calcula la divergencia de $c(\cdot)\nabla I$.

$$d_{i,j}^n = \frac{1}{h^2} [c_{i+1,j}^n (I_{i+1,j}^n - I_{i,j}^n) + c_{i,j}^n (I_{i-1,j}^n - I_{i,j}^n) + c_{i,j+1}^n (I_{i,j+1}^n - I_{i,j}^n) + c_{i,j}^n (I_{i,j-1}^n - I_{i,j}^n)]$$

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Algoritmo secuencial



Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

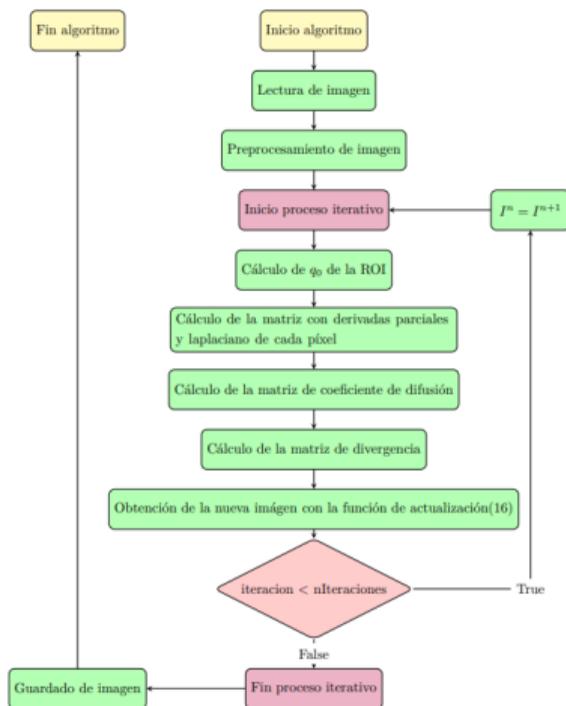
Líneas futuras

Por último, aproximando la diferencial temporal con diferenciación directa, la aproximación numérica de la ecuación diferencial viene dada por

$$I_{i,j}^{n+1} = I_{i,j}^n + \frac{\Delta t}{4} d_{i,j}^n.$$

También llamada función de actualización SRAD.

Algoritmo secuencial



Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Implementación haciendo uso de MPI.

- Ω_k es la región perteneciente al nodo.
- Ω_k^ξ es la región necesaria para el calculo del filtro por cada nodo.
- Es necesaria la comunicación de filas entre nodos.

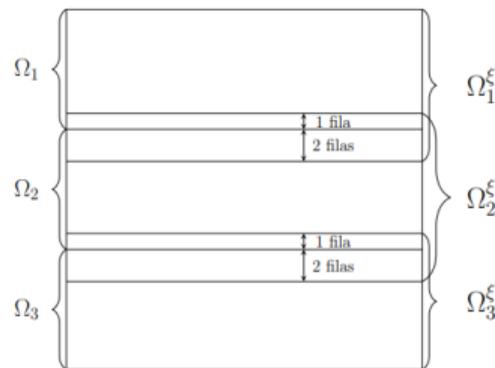


Figura: Subdominios solapados Ω_k^ξ , $k = 1, 2, 3$

Versión OpenMP



Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP

Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Hace uso de la librería OpenMP, que proporciona:

- El uso de varias unidades de procesamiento dotadas de memoria compartida.
- Paralelización mediante directivas de compilador, rutinas y variables de entorno.
- Modifica el comportamiento del programa en tiempo de ejecución.

Versión híbrida (MPI+OpenMP)

Combina MPI y OpenMP para explotar características de ambas librerías.

MPI para comunicación de datos entre nodos conectados en red.

OpenMP para paralelización en memoria compartida en cada nodo.

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Máquinas utilizadas (Hardware)

Máquina 1. Computador del Grupo de Investigación CAPP de la UA.

- CPU Intel Xeon E5320 (8 cores).
- 1.86 GHz.
- 8GB RAM.

Máquina 2. Cluster del Instituto Universitario de Investigación Informática.

- 26 Nodos.
- (2x) CPU Intel Xeon X5660 (12 cores/Nodo).
- 2.3 GHz.
- 48GB RAM.

Máquina 3. Supercomputador Trunks de UMH.

- (2x) CPU Intel Xeon Gold 6140 (36 cores).
- 2.8 GHz.
- 384GB RAM DDR3.

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Máquinas utilizadas (Software)

Máquina 1. Computador del Grupo de Investigación CAPP de la UA.

- Linux Ubuntu 18.04.4 LTS
- C/OpenMP: gcc 7.4.0
- MPI: Openmpi 4.0.3

Máquina 2. Cluster del Instituto Universitario de Investigación Informática.

- Linux Centos 7.9
- C/OpenMP: gcc 4.8.5
- MPI: mpicc 5.8.5

Máquina 3. Supercomputador Trunks de UMH.

- Linux CentOS 7.9.2009 (Core)
- GNU gcc 11.1.0
- openmpi 4.0.5

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

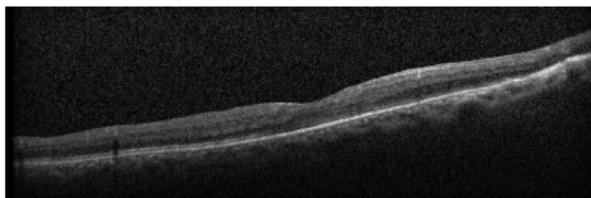
Máquinas
Imágenes
Métricas

Resultados

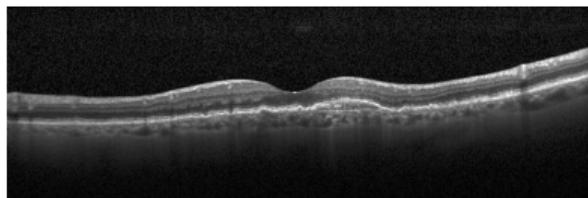
Conclusiones

Líneas futuras

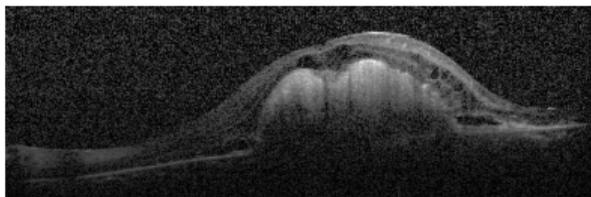
Imágenes utilizadas (I)



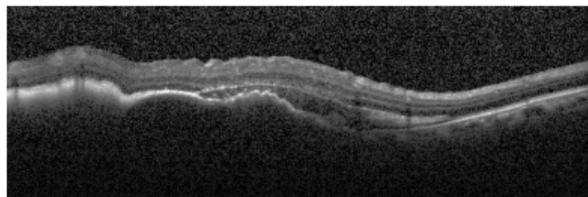
(a) OCT 1 (496 x 1536)



(b) OCT 2 (496 x 1536)



(c) OCT 3 (496 x 1536)



(d) OCT 4 (496 x 1536)

Figura: Imágenes utilizadas

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

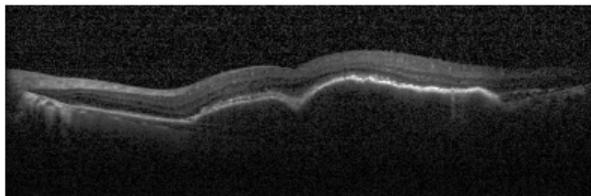
Máquinas
Imágenes
Métricas

Resultados

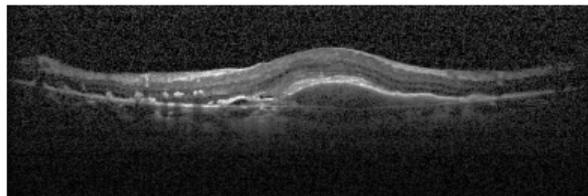
Conclusiones

Líneas futuras

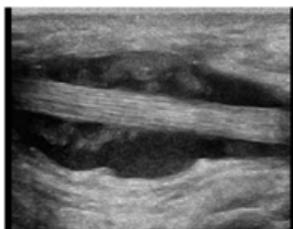
Imágenes utilizadas (II)



(a) OCT 5 (496 x 1536)



(b) OCT 6 (496 x 1536)



(c) US 1 (450 x 600)

Figura: Imágenes utilizadas

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Métricas

El *speedup* se define como:

$$S_p = \frac{T_{sec}}{T_p}$$

Donde T_{sec} es el tiempo de la ejecución secuencial y T_p el tiempo de la ejecución del código paralelo al hacer uso de p unidades de cálculo.

La eficiencia se define como:

$$E = \frac{S_p}{p}$$

Donde S_p es el *speedup* definido anteriormente, y p es el número de unidades de cálculo utilizadas.

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

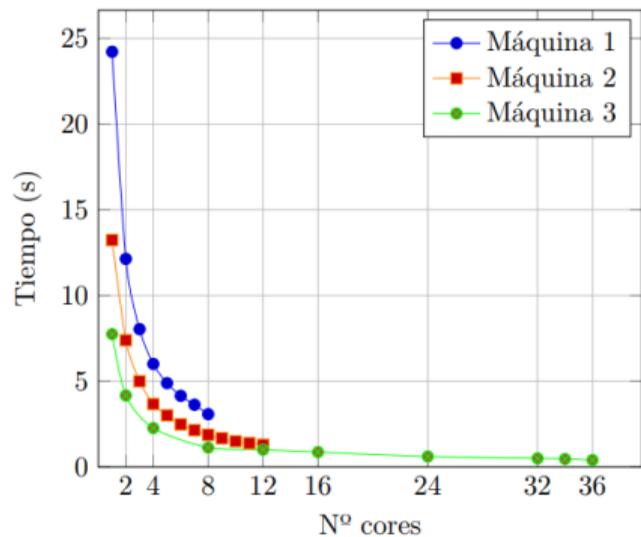
Máquinas
Imágenes
Métricas

Resultados

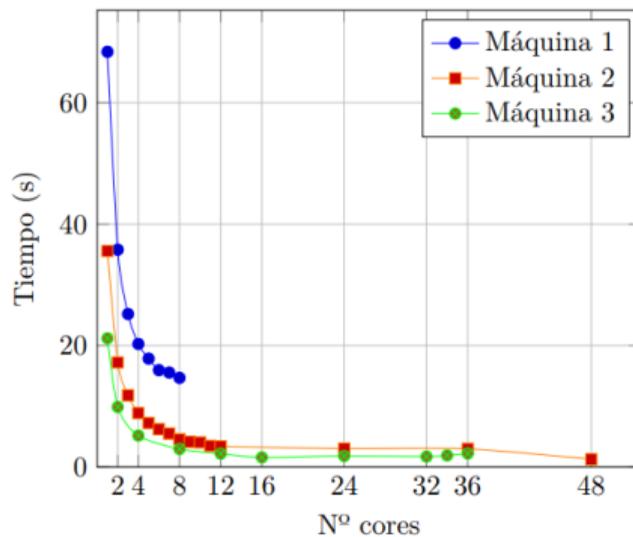
Conclusiones

Líneas futuras

Tiempos obtenidos



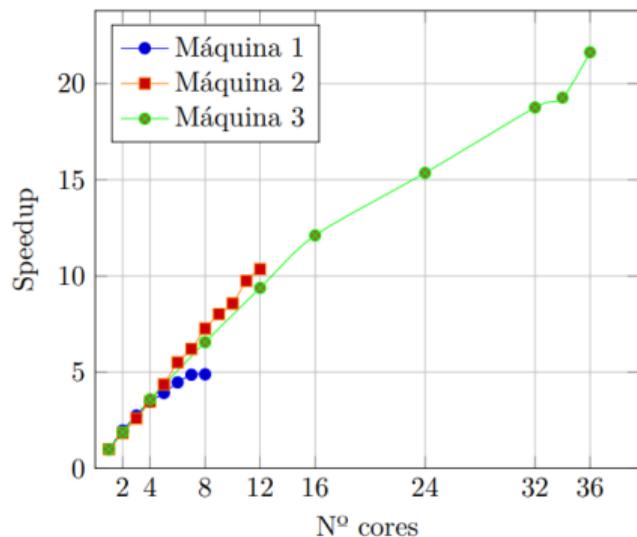
(a) Tiempos obtenidos con OpenMP



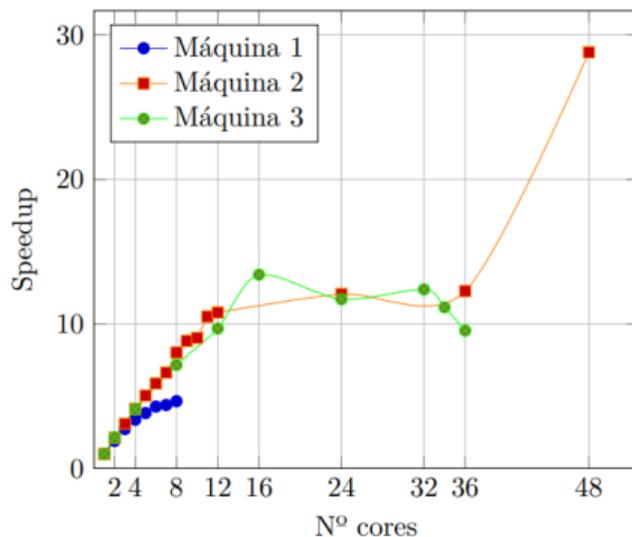
(b) Tiempos obtenidos con MPI

Figura: Tiempos obtenidos en OCT 1

Speedup obtenido



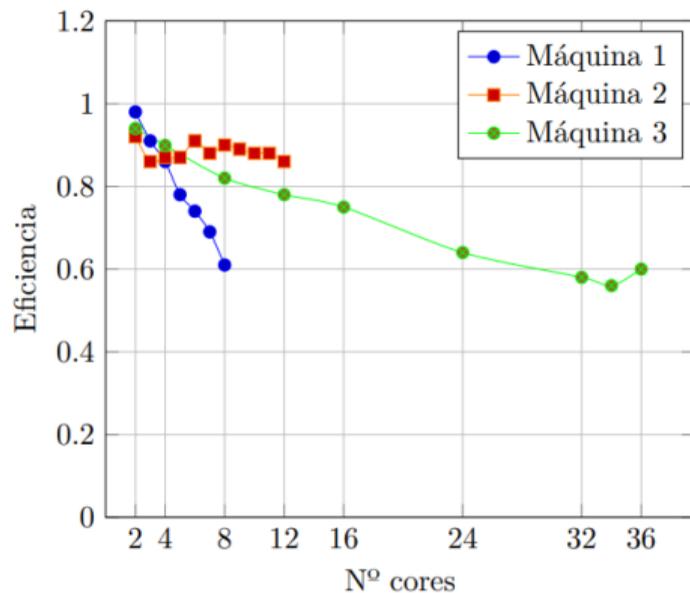
(a) Speedups obtenidos con OpenMP



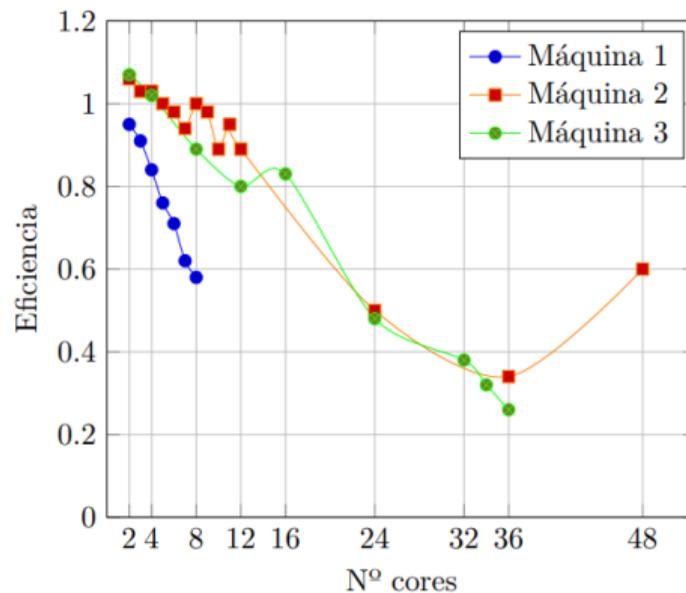
(b) Speedups obtenidos con MPI

Figura: Speedups obtenidos en OCT 1

Eficiencia obtenida



(a) Eficiencias obtenidas con OpenMP



(b) Eficiencias obtenidas con MPI

Figura: Eficiencias obtenidas en OCT 1

Versión híbrida

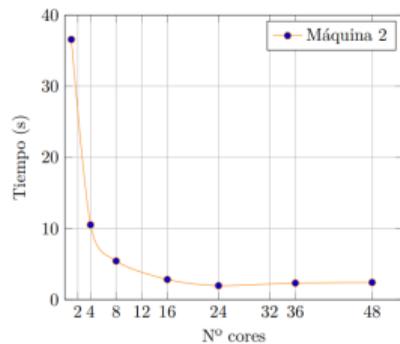


Figura: Tiempos obtenidos en OCT 1 con la versión híbrida

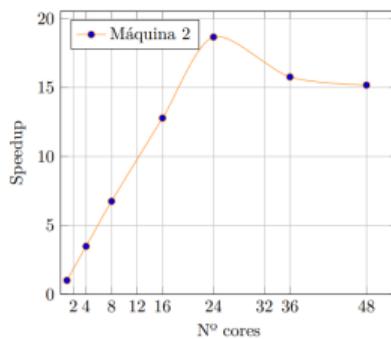


Figura: Speedups obtenidos en OCT 1 con la versión híbrida

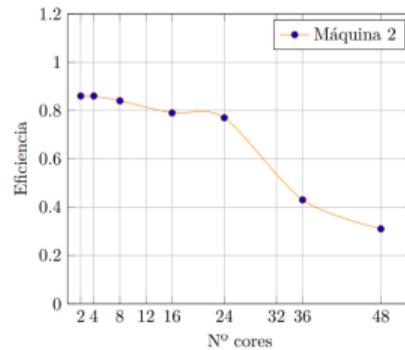


Figura: Eficiencias obtenidas en OCT 1 con la versión híbrida

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Conclusiones

- SRAD presenta una buena calidad de filtrado pero con un alto coste computacional.
- Se ha obtenido una versión secuencial de la que se han implementado las versiones paralelas con librerías OpenMP y MPI.
- Se han desarrollado experimentos con imágenes OCT y ultrasonidos.
- Se han obtenido speedups significativos en tres máquinas paralelas.
- La reducción de tiempo de cómputo de las versiones paralelas facilitan el uso del método SRAD en imágenes médicas.



Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Líneas de trabajo futuras



- Implementación del código paralelo unidades de procesamiento gráfico utilizando tarjetas gráficas de NVIDIA.
- Implementación de un modelo de iteraciones internas locales.
- Desarrollo de un filtro paralelo que combine SRAD con otro filtro, como por ejemplo un filtro guiado.

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Gracias por su atención



Escuela
Politécnica
Superior

Algoritmos paralelos para la reducción de ruido speckle en imágenes médicas.

 Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Adrián Jiménez González
Tutor:
Josep Arnal García

 Universitat d'Alacant
Universidad de Alicante

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

Número de iteraciones seleccionado mediante un estudio del contrast-to-noise ratio (CNR) y las imágenes de salida.

Se calcula mediante:

$$CNR_m = 10 \log_{10} \frac{|\mu_m - \mu_b|}{(0,5(\sigma_m^2 + \sigma_b^2))^{\frac{1}{2}}}$$

donde σ_b y μ_b son la desviación estándar y la media de la sección de fondo, mientras que μ_m y σ_m son la media y la desviación estándar de la sección de interés m .

Iteraciones	OCT 1	OCT 2	OCT 4
100	21.13	24.26	20.39
200	21.91	24.96	22.45
300	23.02	25.33	23.97
400	23.72	25.85	25.12
500	24.1	25.4	25.5
600	24.62	25.65	26.12
700	24.7	26.1	26.54

Cuadro: CNR en distintas imágenes

Paralelización
SRAD

Adrián
Jiménez

Introducción

Contexto
Speckle
Punto inicial

Desarrollo

Algoritmo secuencial
Versión MPI
Versión OpenMP
Versión híbrida

Experimentos

Máquinas
Imágenes
Métricas

Resultados

Conclusiones

Líneas futuras

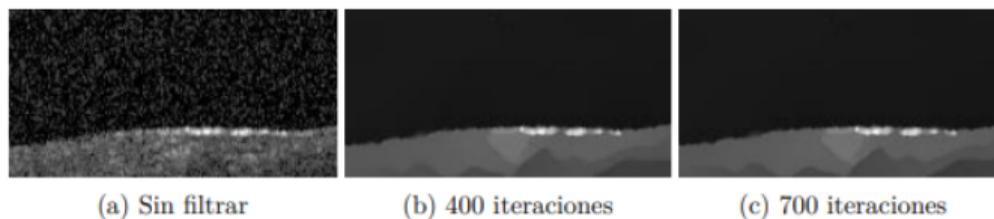


Figura: Filtrado de OCT 1

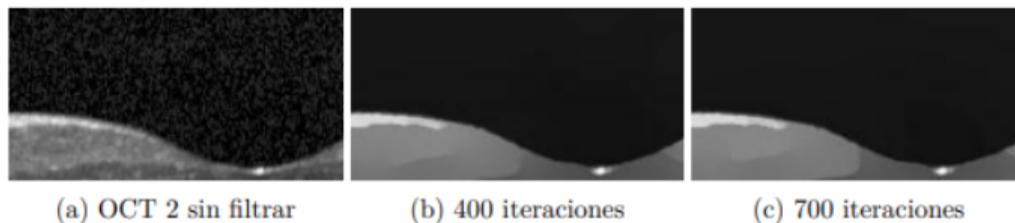


Figura: Filtrado de OCT 2